SYSTEM MCQs COLLECTION

NSCT Prep

Free MCQ Practice for NSCT Test Preparation

Programming (C++, Java, Python)

1440 Multiple Choice Questions

nsctprep.dev

This dataset is created and compiled by Muhammad Abdullah Awais

Easy Questions

480 questions

Q1. What is a computer program?

A. A set of instructions for a computer to execute
B. A system that manages hardware resources
C. A physical component inside the computer case
D. A protocol for transmitting network data

Answer: A

Q2. Which of the following is a low-level programming language?

A. JavaScript
B. Kotlin
C. Assembly
D. Python

Answer: C

Q3. What does a compiler do?

A. Allocates and manages runtime memory
B. Converts entire source code to machine code
C. Identifies and removes bugs from programs
D. Executes source code one line at a time

Answer: B

Q4. Which is NOT a programming paradigm?

A. Procedural Programming
B. Functional Programming
C. Structural Networking
D. Object-Oriented Programming

Answer: C

Q5. What is the purpose of pseudocode?

A. To plan program logic in plain language
B. To write actual machine-executable code
C. To compile a program into bytecode
D. To debug errors in existing code

Answer: A

Q6. What is a syntax error?

A. A failure in the underlying hardware
B. A violation of the language grammar rules
C. A mistake in the program logic flow
D. A crash that occurs during runtime only

Answer: B

Q7. What is the difference between a compiler and an interpreter?

A. An interpreter generates standalone machine code files for later execution
B. They are functionally identical with no meaningful differences at all
C. A compiler translates all code at once; an interpreter translates line by line
D. A compiler always executes faster than an interpreter in every case

Answer: C

Q8. What is an algorithm?

A. A step-by-step problem-solving procedure
B. A high-level programming language
C. A standalone software application
D. A structured data storage format

Answer: A

Q9. Which of the following is a high-level language?

A. Python
B. Assembly
C. FORTRAN
D. Haskell

Answer: A

Q10. What does IDE stand for?

A. Integrated Development Environment
B. Internal Data Engine
C. Indexed Database Engine
D. Internet Development Extension

Answer: A

Q11. Which of the following is a primitive data type in most languages?

A. List
B. Integer
C. Object
D. Array

Answer: B

Q12. What data type is used to store true or false values?

A. string
B. float
C. boolean
D. integer

Answer: B

Q13. What is a variable?

A. A fixed constant value in the program
B. A reusable block of executable code
C. A named storage location in memory
D. A control structure for iteration

Answer: C

Q14. What is the default value of an integer variable in Java?

A. 1
B. -1
C. 0
D. null

Answer: C

Q15. Which data type would you use to store a single letter?

A. String
B. double
C. boolean
D. char

Answer: D

Q16. What is the difference between float and double?

A. They are exactly the same data type
B. double has more precision than float
C. double is used for storing integers
D. float has more precision than double

Answer: B

Q17. What does declaring a variable mean?

A. Specifying its name and type
B. Assigning it an initial value
C. Printing its current value
D. Removing it from the memory

Answer: A

Q18. Which keyword is used to declare a constant in Java?

A. static
B. fixed
C. final
D. const

Answer: C

Q19. What is a string?

A. A boolean value
B. A single character
C. A sequence of characters
D. A numeric data type

Answer: C

Q20. What is type casting?

A. Removing a variable from program memory
B. Declaring a variable with a specific name
C. Converting a value from one type to another
D. Defining a brand new composite data type

Answer: C

Q21. What does the + operator do with two integers?

A. Concatenates them together
B. Compares their values
C. Multiplies them together
D. Adds them arithmetically

Answer: D

Q22. What is the result of 10 % 3?

A. 10
B. 0
C. 3
D. 1

Answer: D

Q23. Which operator is used for assignment in most programming languages?

A. :=
B. !=
C. =
D. ==

Answer: C

Q24. What does the == operator do?

A. Declares a new variable in scope
B. Assigns a value to a variable
C. Performs addition of two values
D. Checks if two values are equal

Answer: D

Q25. What is the result of 5 > 3?

A. true
B. 3
C. false
D. 5

Answer: A

Q26. Which operator performs integer division in Python 3?

A. //
B. %
C. **
D. /

Answer: A

Q27. What does the NOT (!) operator do?

A. Reverses a boolean value
B. Adds two values together
C. Multiplies two operands
D. Assigns a new value

Answer: A

Q28. What is the result of true AND false?

A. false
B. error
C. true
D. null

Answer: A

Q29. What does the ++ operator do?

A. Adds two variables
B. Doubles the value
C. Increments the value by 1
D. Concatenates strings

Answer: C

Q30. Which operator is used for string concatenation in Java?

A. +
B. &
C. .
D. ,

Answer: A

Q31. What does an if statement do?

A. Declares a new variable in the scope
B. Defines a reusable function or method
C. Executes code only if condition is true
D. Repeats a block of code in a loop

Answer: C

Q32. What is a loop?

A. A conditional branching statement
B. A construct that repeats code blocks
C. A syntax for declaring variables
D. A mechanism for calling functions

Answer: B

Q33. What is the purpose of the 'else' clause?

A. To terminate the entire program execution
B. To execute code when if condition is false
C. To define a reusable function or procedure
D. To start a new loop iteration immediately

Answer: B

Q34. Which loop checks the condition before executing the body?

A. both of them
B. do-while loop
C. neither of them
D. while loop

Answer: D

Q35. What does the 'break' statement do in a loop?

A. Restarts the loop from start
B. Terminates the loop entirely
C. Pauses the loop temporarily
D. Skips the current iteration only

Answer: B

Q36. How many times does a for loop 'for(int i=0; i<5; i++)' execute?

A. 4
B. 0
C. 5
D. 6

Answer: C

Q37. What is a switch statement?

A. A syntax for declaring new functions
B. A type of iterative loop construct
C. A multi-way branch on expression value
D. A mechanism for handling exceptions

Answer: C

Q38. What does the 'continue' statement do?

A. Restarts the loop from the very beginning
B. Skips rest of iteration and moves to next
C. Pauses execution until user gives input
D. Exits and terminates the entire loop

Answer: B

Q39. What is a nested loop?

A. A loop that has been terminated
B. A loop without any condition set
C. A loop that runs only one time
D. A loop placed inside another loop

Answer: D

Q40. What is the output of: if(false) print('A'); else print('B');?

A. BA
B. AB
C. A
D. B

Answer: D

Q41. What is a function in programming?

A. A reusable block performing a specific task
B. A loop that iterates over a collection
C. A named variable storing data values
D. A primitive or composite data type name

Answer: A

Q42. What is a function parameter?

A. A global variable accessible from anywhere
B. A variable that receives a passed-in value
C. The return value that a function produces
D. The identifier name given to the function

Answer: B

Q43. What does the return statement do?

A. Declares a variable in the function scope
B. Sends a value back and exits the function
C. Terminates the entire running program
D. Starts a new loop inside the function

Answer: B

Q44. What is the difference between a parameter and an argument?

A. A parameter is the actual value passed during the function call
B. A parameter is in the definition; an argument is the passed value
C. They are exactly the same thing with no differences at all
D. An argument is defined in the function formal declaration only

Answer: B

Q45. What is a void function?

A. A function that takes no input parameters
B. A function with an empty body and no code
C. A function that does not return any value
D. A function that is syntactically invalid

Answer: C

Q46. What is a function call?

A. Writing the definition of a function
B. Invoking a function to execute its code
C. Removing a function from the program
D. Adding a comment above a function

Answer: B

Q47. Can a function call another function?

A. Only void functions can call other ones
B. Only if they are defined in the same class
C. No, functions cannot call other functions
D. Yes, functions can call other functions

Answer: D

Q48. What is a built-in function?

A. A function provided by the language or library
B. A function that you must always write yourself
C. A function that calls itself recursively
D. A function defined inside a loop construct

Answer: A

Q49. What is the main function?

A. Any function found anywhere in the program source
B. The last function that gets called before exiting
C. The entry point of execution in languages like C/Java
D. A mathematical function for numeric calculations

Answer: C

Q50. What happens if a function is defined but never called?

A. It causes an error
B. It runs at the end
C. It runs automatically
D. Its code is never executed

Answer: D

Q51. What does I/O stand for in programming?

A. Integer or Object
B. Index with Order
C. Internal Operations
D. Input and Output

Answer: D

Q52. Which function is used to display output in Python?

A. echo()
B. write()
C. print()
D. printf()

Answer: C

Q53. What is standard input (stdin)?

A. A persistent network data connection
B. The default input source like keyboard
C. A database for storing program data
D. A file stored on the hard disk drive

Answer: B

Q54. What is standard output (stdout)?

A. The default output destination like console
B. A database for persisting program output
C. A connected printer device only
D. A file stored on persistent disk storage

Answer: A

Q55. Which function reads a line of input from the user in Python?

A. input()
B. scanf()
C. getLine()
D. readLine()

Answer: A

Q56. What is a newline character?

A. A whitespace character for horizontal spacing
B. A tab character for indenting text in documents
C. A special character that moves cursor to next line
D. A backspace character that deletes previous input

Answer: C

Q57. What does printf() do in C?

A. Closes an open file descriptor handle
B. Formats and prints output to the console
C. Reads input from standard input stream
D. Allocates dynamic memory on the heap

Answer: B

Q58. What is a format specifier?

A. A variable name in source code
B. A placeholder defining value display format
C. A counter used inside loop bodies
D. A file type extension on disk

Answer: B

Q59. What is console output?

A. Storing structured data in a database table
B. Sending formatted data to a connected printer
C. Displaying information on screen or terminal
D. Writing data to a file on disk storage

Answer: C

Q60. What does Scanner class do in Java?

A. Compiles Java source code into bytecode
B. Reads input from various sources like keyboard
C. Scans the system for viruses and malware
D. Writes formatted output to the console

Answer: B

Q61. How are strings typically stored in memory?

A. As boolean value sequences
B. As integer values in memory
C. As an array of characters
D. As floating-point numbers

Answer: C

Q62. What does the length/len function return for a string?

A. The index of the last character
B. The memory size in bytes
C. The number of characters in the string
D. The ASCII value of the first character

Answer: C

Q63. What is string concatenation?

A. Splitting a string into parts
B. Reversing a string order
C. Joining two or more strings
D. Deleting a string entirely

Answer: C

Q64. What is the index of the first character in a string in most languages?

A. -1
B. 2
C. 0
D. 1

Answer: C

Q65. What does the toUpperCase() method do?

A. Converts all characters to lowercase
B. Converts all characters to uppercase
C. Reverses the character order fully
D. Removes all whitespace characters

Answer: B

Q66. What is a substring?

A. The name of a string variable only
B. A string that contains no characters
C. A contiguous portion of the string
D. A complete copy of the entire string

Answer: C

Q67. What is an empty string?

A. A string with spaces only inside it
B. An uninitialized variable with no type
C. A null value with no reference set
D. A string with zero characters in it

Answer: D

Q68. What does the trim() method do?

A. Removes leading and trailing whitespace
B. Converts all characters to lowercase form
C. Removes every character from the string
D. Splits the string at every space character

Answer: A

Q69. How do you compare strings in Java?

A. Using the == operator directly
B. Using the + concatenation operator
C. Using the > comparison operator
D. Using the equals() method call

Answer: D

Q70. What is a character in programming?

A. A string of exactly two characters
B. A numeric whole number data type
C. A single symbol like a letter or digit
D. A boolean true or false data value

Answer: C

Q71. What is an array?

A. A loop construct that iterates over data elements
B. A collection of same-type elements in contiguous memory
C. A reusable function that performs a specific task
D. A single variable holding one data value

Answer: B

Q72. What is the index of the first element in an array in C/Java?

A. 0
B. -1
C. 2
D. 1

Answer: A

Q73. What is an ArrayList in Java?

A. A doubly linked list of elements
B. A key-value pair mapping structure
C. A fixed-size array that cannot change
D. A dynamic array that grows and shrinks

Answer: D

Q74. What is a list in Python?

A. A dictionary mapping keys to their associated values
B. An immutable sequence that cannot be modified at all
C. A fixed-size array of uniform typed elements
D. An ordered mutable collection of any element types

Answer: D

Q75. What happens when you access an array index out of bounds?

A. It returns zero as the default value for any position
B. It returns null as the default value for the position
C. It throws an exception or causes undefined behavior
D. It automatically resizes the array to fit the index

Answer: C

Q76. What is a dictionary/map?

A. An ordered array with indexed elements
B. A list of words in alphabetical order
C. A string type for storing text values
D. A collection of key-value pair entries

Answer: D

Q77. What is a stack?

A. A LIFO (Last-In First-Out) structure
B. A sorted list of ordered elements
C. A random access data structure
D. A FIFO (First-In First-Out) structure

Answer: A

Q78. What is a queue?

A. A LIFO (Last-In First-Out) structure
B. A sorted array with indexed elements
C. A FIFO (First-In First-Out) structure
D. A hierarchical tree data structure

Answer: C

Q79. What is the difference between an array and a linked list?

A. Arrays can dynamically grow and shrink in size without any overhead
B. Arrays use contiguous memory with O(1) access; linked lists use nodes
C. They are functionally identical data structures with no differences
D. Linked lists are always faster than arrays for every type of operation

Answer: B

Q80. What is a set in programming?

A. A sorted array of ordered elements
B. A specialized type of key-value map
C. A collection with only unique elements
D. An array that allows duplicate elements

Answer: C

Q81. What is a class in OOP?

A. A variable type for storing data
B. A blueprint for creating objects
C. A loop construct for iteration
D. A reusable function for operations

Answer: B

Q82. What is an object in OOP?

A. A standalone function
B. A class definition in code
C. A primitive data type
D. An instance of a class

Answer: D

Q83. What is encapsulation?

A. Inheriting properties from a parent class definition
B. A sorting method that orders elements in a collection
C. A type of iterative loop control structure in code
D. Bundling data and methods while restricting direct access

Answer: D

Q84. What is inheritance in OOP?

A. Acquiring properties and methods from a parent class
B. Creating a new object instance from a class
C. A form of encapsulation that hides internal details
D. A design pattern for structuring application code

Answer: A

Q85. What is a constructor?

A. A static method belonging to the class itself
B. A getter method that returns a field value
C. A destructor that cleans up object state
D. A special method called when creating an object

Answer: D

Q86. What is a method in OOP?

A. A standalone function not tied to any class
B. A function defined inside a class for objects
C. A variable that stores data in the program
D. A data type that defines value constraints

Answer: B

Q87. What does the 'this' keyword refer to?

A. The current object instance
B. A static method reference
C. The class definition itself
D. The parent class reference

Answer: A

Q88. What is an access modifier?

A. A keyword controlling member visibility level
B. A data type for storing numeric values only
C. A type of loop construct in the language
D. A constructor for initializing object state

Answer: A

Q89. What is the 'new' keyword used for?

A. Deleting an existing object from memory
B. Declaring a new variable in the scope
C. Defining a new method in the class
D. Creating a new instance of a class

Answer: D

Q90. What is polymorphism?

A. A class relationship defined by inheritance hierarchy
B. A type of encapsulation hiding internal data details
C. Different objects responding differently to same method
D. Having only one single form of behavior always

Answer: C

Q91. What is RAM?

A. Random Access Memory, volatile memory for programs
B. A programming language for system-level development
C. A type of hard drive for persistent data storage
D. Read-Access Module for hardware communication

Answer: A

Q92. What is the stack in memory?

A. A memory region storing locals and call info in LIFO
B. The hard drive used for persistent data storage
C. A data structure only with no memory role
D. A type of processor cache for fast data access

Answer: A

Q93. What is the heap in memory?

A. Read-only memory that cannot be written to
B. Identical to the stack memory region exactly
C. A sorted data structure for priority access
D. A memory region for dynamic memory allocation

Answer: D

Q94. What is a memory leak?

A. A security vulnerability in network protocols
B. A hardware failure in the memory modules
C. A type of buffer overflow in array operations
D. Allocated memory no longer used but not freed

Answer: D

Q95. What is garbage collection?

A. A sorting algorithm for ordering data values
B. Deleting files from the disk storage system
C. Running disk cleanup utilities on the system
D. Automatic reclamation of unused object memory

Answer: D

Q96. What is a pointer?

A. A composite data type definition
B. A variable storing a memory address
C. A reusable block of executable code
D. A variable holding a computed value

Answer: B

Q97. What is the difference between stack and heap allocation?

A. Stack is automatic and fast with limited size; heap is larger but slower
B. Heap allocation is always faster than stack allocation in all scenarios
C. There is no meaningful difference between the two allocation methods
D. Stack memory is unlimited while heap memory has a very strict size cap

Answer: A

Q98. What is a null pointer?

A. A pointer to the memory address zero
B. A pointer that has been deleted already
C. A pointer to the stack memory region
D. A pointer to no valid memory location

Answer: D

Q99. What does 'free()' do in C?

A. Frees up disk space on the storage drive
B. Deallocates previously allocated heap memory
C. Resets a variable back to its default value
D. Allocates new memory on the heap for use

Answer: B

Q100. What is a variable's lifetime?

A. The duration the variable exists in memory
B. Its declared data type in the program
C. Its identifier name in the source program
D. Its accessibility scope in the source code

Answer: A

Q101. What is an exception?

A. A function call to a subroutine in the program
B. An abnormal event disrupting normal program flow
C. A normal expected program event during execution
D. A variable type for storing error code values

Answer: B

Q102. What is a try-catch block?

A. A function definition with parameters and body
B. A looping construct for repeated iteration
C. A construct attempting code and handling exceptions
D. A conditional statement for branching logic

Answer: C

Q103. What is the purpose of the 'finally' block?

A. To define a new function with a specific name
B. To terminate the program execution immediately
C. To execute code regardless of exception outcome
D. To finalize a variable and prevent changes

Answer: C

Q104. What is a NullPointerException?

A. A compile-time syntax error in the code
B. A valid and expected operation result
C. A warning generated by the code linter
D. An exception from using a null reference

Answer: D

Q105. What does 'throw' do in exception handling?

A. Ignores an exception and continues running
B. Explicitly raises and creates an exception
C. Logs an exception message to the console
D. Catches an exception from the try block

Answer: B

Q106. What is an ArithmeticException?

A. A type mismatch between two variable types
B. A file not found error during I/O access
C. An exception for illegal arithmetic operations
D. A syntax error in the source code text

Answer: C

Q107. What is the difference between compile-time and runtime errors?

A. Compile-time errors cause the running program to crash unexpectedly
B. Runtime errors are detected and caught by the compiler before running
C. Compile-time errors are caught before execution; runtime during it
D. There is no meaningful difference between the two error types

Answer: C

Q108. What is an ArrayIndexOutOfBoundsException?

A. A type error from wrong array element types
B. An exception for accessing an invalid index
C. A memory leak from unreleased array memory
D. A valid and expected array access result

Answer: B

Q109. Can you have multiple catch blocks for one try?

A. Yes, to handle different exception types differently
B. No, only one catch block is allowed per try statement
C. Only for a maximum of exactly two exception types total
D. Only in Python, not in any other programming language

Answer: A

Q110. What is an error message?

A. A variable name used for storing error data
B. A text description about what went wrong
C. A function output returned to calling code
D. A comment written inside the source code

Answer: B

Q111. What is a module in programming?

A. A data structure for storing elements
B. A variable type for storing data values
C. A self-contained unit of related code
D. A type of iterative loop construct

Answer: C

Q112. What is the purpose of the import statement?

A. To include functionality from another module
B. To delete a module from the project files
C. To compile code into executable binaries
D. To export code to other modules or files

Answer: A

Q113. What is a package in Python?

A. A single Python source code file only
B. A Python class with methods and attributes
C. A directory containing modules and __init__.py
D. A standalone Python function definition

Answer: C

Q114. What is a library in programming?

A. A collection of pre-written reusable code modules
B. A place for storing reference books and manuals
C. An integrated development environment application
D. A single function defined in the source code file

Answer: A

Q115. What is the difference between a module and a package?

A. They are exactly the same thing with different names
B. A module is a single file; a package is a collection
C. A package is smaller in size compared to a module
D. A module contains multiple packages within itself

Answer: B

Q116. What is pip in Python?

A. A general-purpose programming language for development
B. A compiler that converts Python to machine code binary
C. A debugger tool for finding errors in Python programs
D. The package installer for Python that downloads packages

Answer: D

Q117. What is npm?

A. A Python-specific development tool only
B. A database for storing application data values
C. A compiler for JavaScript source code files
D. Node Package Manager for JavaScript packages

Answer: D

Q118. What is an API?

A. A general-purpose programming language for development
B. A relational database for storing structured data records
C. A set of definitions and protocols for software integration
D. A compiler that translates source code to machine language

Answer: C

Q119. What is a namespace?

A. A string type for storing text data values in memory
B. A file extension that identifies the programming language
C. A container for identifiers to avoid naming conflicts
D. A variable name in the source code program

Answer: C

Q120. What does 'from math import sqrt' do in Python?

A. Creates a new sqrt function in the current file
B. Imports only the sqrt function from math module
C. Deletes the sqrt function from the math module
D. Imports the entire math module with all functions

Answer: B

Q121. What is a design pattern?

A. A reusable solution template for common software problems
B. A type of algorithm for data processing
C. A user interface visual design layout
D. A programming language for building applications

Answer: A

Q122. What is recursion?

A. A function calling itself to solve problems
B. An error in the program source code
C. A type of iterative loop construct
D. A data structure for storing elements

Answer: A

Q123. What is a data structure?

A. A way of organizing data for efficient access
B. A programming language for writing software
C. A function that performs a specific operation
D. A database management system for storage

Answer: A

Q124. What is an event in event-driven programming?

A. An action or occurrence that a program can respond to
B. A function that performs a specific task when called
C. A syntax error in the source code of the program
D. A variable that stores data values during execution

Answer: A

Q125. What is a generic/template in programming?

A. A design pattern for building complex object hierarchies
B. Code that works with any type specified at compile time
C. A default value assigned to variables automatically
D. A runtime check that validates data types dynamically

Answer: B

Q126. What is the Singleton pattern?

A. Ensuring a class has only one global instance
B. Creating many instances of a class at once
C. An error pattern for handling exceptions
D. A function pattern for recursive operations

Answer: A

Q127. What is abstraction in programming?

A. A primitive data type for storing numeric values only
B. Making code unnecessarily complex and hard to follow
C. Hiding details while exposing essential features only
D. A type of error that occurs during program execution

Answer: C

Q128. What is an annotation/decorator?

A. Metadata or wrapper adding behavior without modifying code
B. A loop construct for iterating over collection elements
C. A comment in the source code for documentation
D. A variable that stores a single data value in memory

Answer: A

Q129. What is refactoring?

A. Adding new features and functionality to the existing codebase
B. Restructuring code to improve structure without changing behavior
C. Deleting unused code and removing deprecated function calls
D. Rewriting the entire codebase from scratch completely

Answer: B

Q130. What is the DRY principle?

A. A testing principle for validating dry run output
B. A moisture management rule for server rooms
C. Avoid duplicating code by abstracting common logic
D. A drying algorithm for removing duplicate data

Answer: C

Q131. What is concurrency in programming?

A. Running only one task at a time sequentially
B. Handling multiple tasks in overlapping time periods
C. Exactly the same concept as true parallelism
D. Strictly sequential execution of all operations

Answer: B

Q132. What is a thread?

A. A reusable function definition block
B. A type of string data structure
C. The smallest unit of execution within a process
D. A data structure for storing elements

Answer: C

Q133. What is a process?

A. A function call that executes and then returns
B. An independent program with its own memory space
C. A variable that stores data during execution
D. The same thing as a thread in all respects

Answer: B

Q134. What is the difference between concurrency and parallelism?

A. Concurrency always requires multiple CPU cores to function properly
B. Parallelism is always slower than concurrency in every scenario
C. They are exactly the same concept with no differences at all
D. Concurrency manages overlapping tasks; parallelism runs simultaneously

Answer: D

Q135. What is multithreading?

A. A type of recursion with multiple recursive calls
B. Using multiple programming languages in one project
C. Using multiple processors in a distributed system
D. Running multiple threads within a single process

Answer: D

Q136. What is a race condition?

A. A debugging tool for analyzing multi-threaded applications
B. A bug where outcome depends on non-deterministic thread timing
C. A performance competition between different threads
D. A type of exception thrown during concurrent data access

Answer: B

Q137. What is synchronization?

A. A file operation for reading and writing data safely
B. Running code statements in sequential order always
C. A network protocol for data transfer between systems
D. Coordinating shared resource access to prevent races

Answer: D

Q138. What is a mutex?

A. An algorithm for sorting elements in a collection
B. A primitive data type for storing values
C. A lock ensuring exclusive resource access for one thread
D. A function for performing calculations on data values

Answer: C

Q139. What is a deadlock?

A. A memory leak from unreleased dynamically allocated data
B. A type of infinite loop caused by incorrect conditions
C. Threads blocked forever waiting for each other resources
D. A program crash caused by a runtime error in code

Answer: C

Q140. What is the main benefit of concurrency?

A. Improved performance and responsiveness of programs
B. Simpler code that is easier to read and understand
C. Smaller program binaries with reduced memory footprint
D. Fewer bugs and errors during program execution overall

Answer: A

Q141. What is debugging?

A. Compiling code into machine language
B. Writing new source code from scratch
C. Finding and fixing errors in a program
D. Running automated tests on the code

Answer: C

Q142. What is a breakpoint?

A. An error in the source code of the program
B. A marker where the debugger pauses for inspection
C. An exit point from a loop during execution flow
D. The end of a function where control is returned

Answer: B

Q143. What is a unit test?

A. A user acceptance test for final validation
B. Testing the entire application end to end
C. A test verifying a single isolated code unit
D. A performance test measuring execution speed

Answer: C

Q144. What is a bug?

A. A type of variable storing debug data
B. A feature that works as intended
C. A function used for error handling
D. An error or flaw causing wrong behavior

Answer: D

Q145. What is print/console debugging?

A. Using print statements to display values and flow
B. An automated process for finding and fixing bugs
C. A graphical user interface tool for visual debugging
D. Professional debugging using specialized IDE tools

Answer: A

Q146. What is a test case?

A. A debug tool for stepping through code line by line
B. A set of inputs and expected results for verification
C. A code review process performed by team members
D. A programming language for writing test scripts

Answer: B

Q147. What does the term 'step over' mean in debugging?

A. Execute current line and move to next without entering calls
B. Skip the entire program execution completely
C. Jump directly to the end of the current function body
D. Restart the debugging session from the very beginning

Answer: A

Q148. What is a syntax error?

A. A logic mistake producing incorrect program results
B. A runtime error that occurs during program execution
C. A design flaw in the overall application architecture
D. A violation of language grammar rules at compile time

Answer: D

Q149. What is the purpose of an assertion?

A. To create a new variable in the current scope of code
B. To import a module from an external library into the file
C. To define a new function with parameters and return type
D. To verify a condition is true and fail with message if not

Answer: D

Q150. What is a logic error?

A. An error where program runs but produces incorrect results
B. A syntax mistake caught during code compilation by compiler
C. A type error from assigning incompatible values to variables
D. A compile error that prevents the program from being built

Answer: A

Q151. What is version control?

A. Tracking and managing code changes over time
B. Controlling which program version users access
C. A testing tool for validating software behavior
D. A deployment tool for releasing code to servers

Answer: A

Q152. What is Git?

A. A distributed version control system for code
B. A general-purpose programming language
C. A database for storing application data
D. A compiler for translating source to binaries

Answer: A

Q153. What is a code review?

A. Deleting unused code from the codebase
B. Having developers examine code before merging
C. Writing documentation for the application
D. Running automated tests on the source code

Answer: B

Q154. What is agile development?

A. A programming language designed for rapid application development tasks
B. An iterative approach emphasizing collaboration and short delivery cycles
C. A testing method for validating software behavior against requirements
D. Developing software as fast as possible without any planning at all

Answer: B

Q155. What is a commit in version control?

A. A deployment to the production environment
B. A branch created for feature development
C. Deleting code from the shared repository
D. A snapshot of changes saved with a message

Answer: D

Q156. What is a branch in Git?

A. An independent line of development from main
B. A variable used to store temporary values
C. A copy of a file in the project directory
D. A function defined in the source code file

Answer: A

Q157. What is a merge in Git?

A. Creating a new commit in the repository
B. Splitting code into separate source files
C. Combining changes from one branch into another
D. Deleting a branch from the repository

Answer: C

Q158. What is documentation in software development?

A. Only code comments written inside source files
B. Written descriptions of software design and usage
C. A testing tool for validating software correctness
D. A deployment process for releasing to production

Answer: B

Q159. What is the purpose of code comments?

A. To define variables inside the source file
B. To make the code execute faster at runtime
C. To handle errors during program execution
D. To explain the intent and reasoning in code

Answer: D

Q160. What is a pull request?

A. A data request sent to a server for information
B. A build request to compile and package the application
C. Pulling code packages from the internet registry
D. A request to merge changes with review and discussion

Answer: D

Q161. What is source code in programming?

A. Human-readable instructions for a program
B. The memory address of a running process
C. A hardware signal sent to the processor
D. The binary output after compilation

Answer: A

Q162. What does a linker do in program building?

A. Checks source code for syntax errors
B. Allocates memory for runtime variables
C. Converts code into an intermediate format
D. Combines object files into an executable

Answer: D

Q163. Which term describes a fixed value written directly in code?

A. Keyword
B. Operator
C. Literal
D. Variable

Answer: C

Q164. What is the main purpose of a flowchart?

A. To encrypt data before transmission
B. To allocate memory during execution
C. To compile source code into bytecode
D. To visually represent program logic flow

Answer: D

Q165. What does the term debugging mean?

A. Designing the user interface layout
B. Compiling code into machine language
C. Writing new features for software
D. Finding and fixing errors in code

Answer: D

Q166. Which of these is an example of a markup language?

A. HTML
B. Python
C. Java
D. Ruby

Answer: A

Q167. What is a runtime environment?

A. A database for storing application data
B. A text editor for writing source code files
C. A network protocol for data transmission
D. Software that provides execution support

Answer: D

Q168. What is machine code composed of?

A. English keywords and symbols
B. Flowchart diagram elements
C. Pseudocode planning statements
D. Binary digits zeros and ones

Answer: D

Q169. What is the purpose of comments in code?

A. To speed up program execution time
B. To declare variables automatically
C. To handle errors during runtime
D. To explain code for human readers

Answer: D

Q170. What does open source mean for software?

A. Software that runs only on the internet
B. Software with publicly available source code
C. Software that requires no installation
D. Software created by a single developer

Answer: B

Q171. What is a Boolean data type used to represent?

A. Text strings
B. Memory addresses
C. Decimal numbers
D. True or false

Answer: D

Q172. What is the default value of an uninitialized integer in Java?

A. Undefined
B. Null
C. Zero
D. One

Answer: C

Q173. Which data type stores a single character?

A. char
B. integer
C. float
D. string

Answer: A

Q174. What does the term type casting mean?

A. Creating a new custom data type
B. Assigning a name to a constant
C. Deleting a variable from memory
D. Converting a value to another type

Answer: D

Q175. Which data type is used for numbers with decimal points?

A. char
B. integer
C. float
D. boolean

Answer: C

Q176. What is the difference between a variable and a constant?

A. A variable is global; a constant is always local
B. A constant uses more memory than a variable
C. A variable is faster than a constant in speed
D. A variable can change; a constant cannot change

Answer: D

Q177. What does null represent in programming?

A. A Boolean false state
B. The absence of a value
C. The number zero value
D. An empty text string

Answer: B

Q178. Which keyword declares a constant in many languages?

A. dim
B. let
C. const
D. var

Answer: C

Q179. What is an unsigned integer?

A. An integer with unlimited storage capacity
B. An integer storing only non-negative values
C. An integer stored as a floating-point value
D. An integer that can hold negative values only

Answer: B

Q180. What is a string in programming?

A. A sequence of characters
B. A type of loop structure
C. A memory address pointer
D. A numeric calculation result

Answer: A

Q181. What does the modulus operator (%) return?

A. The quotient of division
B. The remainder of division
C. The absolute difference
D. The product of two values

Answer: B

Q182. Which operator is used for logical AND in most languages?

A. ||
B. ^^
C. &&
D. !!

Answer: C

Q183. What does the increment operator (++) do?

A. Multiplies a value by two
B. Divides a value by one
C. Decreases a value by two
D. Increases a value by one

Answer: D

Q184. What is the assignment operator in most programming languages?

A. ==
B. =
C. !=
D. <=

Answer: B

Q185. Which operator checks if two values are NOT equal?

A. ==
B. >=
C. <=
D. !=

Answer: D

Q186. What does the logical OR operator (||) return?

A. True if at least one operand is true
B. False only when first operand is false
C. The sum of two Boolean operand values
D. True only if both operands are true

Answer: A

Q187. What type of operator is the greater-than sign (>)?

A. Comparison
B. Logical
C. Assignment
D. Arithmetic

Answer: A

Q188. What does the NOT operator (!) do to a Boolean value?

A. Doubles the value
B. Deletes the value
C. Inverts the value
D. Converts to string

Answer: C

Q189. Which symbol represents the multiplication operator?

A. asterisk
B. forward slash
C. minus sign
D. plus sign

Answer: A

Q190. What is the result of 10 / 3 in integer division?

A. 0
B. 3.33
C. 4
D. 3

Answer: D

Q191. What does a while loop do in programming?

A. Executes code a fixed number of times
B. Skips code blocks based on values
C. Repeats code while condition remains true
D. Runs code exactly once then stops

Answer: C

Q192. What is the purpose of the break statement?

A. It restarts the loop from beginning
B. It pauses execution for specified time
C. It skips to the next function in code
D. It exits the current loop or switch

Answer: D

Q193. Which control structure selects between multiple code paths?

A. for loop
B. while loop
C. if-else statement
D. do-while loop

Answer: C

Q194. What does the continue statement do in a loop?

A. Ends the loop entirely and exits immediately
B. Skips the rest of current iteration only
C. Restarts the loop counter to initial value
D. Pauses the loop until user provides input

Answer: B

Q195. What is a do-while loop?

A. A loop needing no ending condition at all
B. A loop that checks condition after executing
C. A loop that runs exactly twice every time
D. A loop that might never execute its body

Answer: B

Q196. What does a switch statement do?

A. Switches between two different program files
B. Transfers control to another function call
C. Toggles a Boolean variable between states
D. Compares a value against multiple case labels

Answer: D

Q197. What is an infinite loop?

A. A loop whose condition never becomes false
B. A loop that processes infinite data items
C. A loop iterating exactly one million times
D. A loop that runs fast but eventually stops

Answer: A

Q198. What is the purpose of the else clause?

A. It always executes regardless of condition
B. It executes when the if condition is false
C. It executes when the if condition is true
D. It ends the program after the if statement

Answer: B

Q199. What are the three parts of a standard for loop?

A. Try, catch, and finally block sections
B. Input, process, and output statements
C. Initialization, condition, and update
D. Declaration, assignment, and return

Answer: C

Q200. What is nesting in control structures?

A. Converting loops into conditional checks
B. Placing one structure inside another one
C. Removing unused code from the program
D. Connecting two programs externally together

Answer: B

Q201. What is a function in programming?

A. A variable that holds numeric values
B. A data type that stores text values
C. A reusable block of code with a name
D. A control structure for looping code

Answer: C

Q202. What is a parameter in a function?

A. A variable that receives input to function
B. The name given to the function itself
C. The data type of the function output
D. The return value of the function call

Answer: A

Q203. What does the return statement do?

A. It repeats the function from the start
B. It sends a value back to the caller
C. It declares a new variable in the scope
D. It prints output to the screen display

Answer: B

Q204. What is a function call?

A. Defining a new function in the program
B. Renaming a function to a new identifier
C. Executing a function by using its name
D. Deleting a function from the program

Answer: C

Q205. What is the difference between an argument and a parameter?

A. Arguments are types; parameters are values only
B. They are the same thing with no difference
C. Arguments are in definition; parameters in calls
D. Parameters are in definition; arguments in calls

Answer: D

Q206. What is a void function?

A. A function that cannot be called by others
B. A function that takes no parameters at all
C. A function that returns no value to caller
D. A function that runs only one time ever

Answer: C

Q207. What does the term function signature mean?

A. The comments written above function body
B. The digital signature used to secure code
C. The total number of lines in the function
D. The name, parameters, and return type info

Answer: D

Q208. What is a built-in function?

A. A function that cannot be modified at all
B. A function that runs before main function
C. A function provided by the language itself
D. A function written by the programmer only

Answer: C

Q209. What is a recursive function?

A. A function that takes no arguments at all
B. A function that runs in parallel threads
C. A function that calls itself to solve tasks
D. A function that never returns any values

Answer: C

Q210. What is the scope of a local variable inside a function?

A. It is shared across all program functions
B. It is accessible from anywhere in program
C. It persists after the function has ended
D. It exists only within that function body

Answer: D

Q211. What does standard input (stdin) refer to?

A. The file system storage on disk
B. The network connection to internet
C. The default input source like keyboard
D. The screen display output device

Answer: C

Q212. What does standard output (stdout) refer to?

A. The printer attached to the computer
B. The default output destination like screen
C. The database server for data storage
D. The keyboard input device

Answer: B

Q213. What is a file in the context of I/O?

A. A temporary variable in program memory
B. A named collection of data stored on disk
C. A network connection between two machines
D. A function that handles user inputs

Answer: B

Q214. What does reading from a file mean?

A. Renaming the file to a new file name
B. Retrieving data from a file into program
C. Deleting the file from the file system
D. Writing new data to the file on disk

Answer: B

Q215. What is the purpose of the print function?

A. To create a new variable in the program
B. To display output to the screen or console
C. To save data permanently to a file on disk
D. To read input from the keyboard device

Answer: B

Q216. What is a file path?

A. The content stored inside a file itself
B. The location address of a file on system
C. The date when a file was last modified
D. The size of a file measured in bytes

Answer: B

Q217. What does writing to a file mean?

A. Storing data from program into a file
B. Deleting all contents of the file only
C. Moving a file to a different directory
D. Reading data from the file into memory

Answer: A

Q218. What is a console in programming?

A. A database management system for data
B. A text-based interface for program I/O
C. A gaming device for entertainment use
D. A graphical design tool for layouts

Answer: B

Q219. What does the input function do in Python?

A. It writes data to a file on the disk
B. It creates a new variable in the code
C. It displays formatted output on screen
D. It reads a line of text from the user

Answer: D

Q220. What is the difference between text and binary file modes?

A. Text mode is faster; binary mode is slower always
B. Text mode is for images; binary mode is for documents
C. Text mode is read only; binary mode is write only
D. Text mode handles characters; binary handles raw bytes

Answer: D

Q221. What is string concatenation?

A. Converting string to number
B. Joining two strings together
C. Splitting a string into parts
D. Reversing a string backward

Answer: B

Q222. What does the length property of a string return?

A. The number of words inside
B. The index of last character
C. The memory size in bytes
D. The number of characters

Answer: D

Q223. What is a substring?

A. A string used as a variable
B. A string with no characters
C. A portion of another string
D. A string stored in memory

Answer: C

Q224. What is an empty string?

A. A string that equals null
B. A string with zero characters
C. A string containing spaces only
D. A string holding the digit zero

Answer: B

Q225. What does converting a string to uppercase mean?

A. Removing all spaces from the string text
B. Adding exclamation marks after each word
C. Changing all letters to capital letters
D. Increasing the font size of the string

Answer: C

Q226. What is a character index in a string?

A. The total count of characters in string
B. The numeric position of a character
C. The ASCII value of the character
D. The memory address of the character

Answer: B

Q227. What does the trim function do to a string?

A. Converts the string to lowercase only
B. Removes leading and trailing whitespace
C. Shortens the string to ten characters
D. Removes all characters from the string

Answer: B

Q228. What is an escape character in a string?

A. A character that exits the program when found
B. A character that deletes the previous character
C. A character that converts text into numeric data
D. A character starting a special sequence in text

Answer: D

Q229. What does string comparison do?

A. Counts how many strings exist in program
B. Joins two strings into one combined string
C. Converts strings into integer numeric values
D. Checks if two strings are equal or ordered

Answer: D

Q230. What is the newline character used for?

A. To add a space between two words
B. To underline the current text line
C. To delete the current line of text
D. To move text to the next line down

Answer: D

Q231. What is an array in programming?

A. A collection of elements stored in order
B. A function that processes numeric data
C. A single variable that holds one value
D. A loop that iterates over user input

Answer: A

Q232. What is an array index?

A. The data type of array elements
B. The position number of an element
C. The memory size of the full array
D. The total number of elements in array

Answer: B

Q233. What does it mean to traverse an array?

A. To delete all elements from the array
B. To visit each element of the array once
C. To sort the array in ascending order
D. To copy the array to another variable

Answer: B

Q234. What is a two-dimensional array?

A. An array organized in rows and columns
B. An array that stores only two elements
C. An array with two different index types
D. An array that can hold two data types

Answer: A

Q235. What is a list in Python?

A. A read-only sequence of characters
B. A function that returns multiple values
C. A fixed-size array of integers only
D. An ordered mutable collection of items

Answer: D

Q236. What does the push operation do on a stack?

A. Searches for an element in the stack
B. Adds an element to the top of stack
C. Sorts all elements within the stack
D. Removes the top element from stack

Answer: B

Q237. What is the difference between a stack and a queue?

A. Both use the same ordering of elements
B. Stack is LIFO; queue is FIFO ordering
C. Stack stores numbers; queue stores text
D. Stack is FIFO; queue is LIFO ordering

Answer: B

Q238. What is the purpose of a dictionary or map?

A. To store data as key-value pair entries
B. To count the number of elements in list
C. To sort data in alphabetical order always
D. To convert arrays into string format

Answer: A

Q239. What happens when you access an array out of bounds?

A. The array automatically grows in size
B. An error or undefined behavior occurs
C. The array resets to its initial state
D. The first element is returned instead

Answer: B

Q240. What is a set in programming?

A. A collection of unique elements only
B. A sorted sequence of text characters
C. A fixed-size array of numeric values
D. An ordered list allowing duplicate values

Answer: A

Q241. What is a class in object-oriented programming?

A. A function that processes data values
B. A loop structure for iteration
C. A variable storing numeric data
D. A blueprint for creating objects

Answer: D

Q242. What is an object in programming?

A. A type of loop control structure
B. A keyword reserved by the language
C. An instance of a class in memory
D. A comment in the source code text

Answer: C

Q243. What is inheritance in OOP?

A. Copying code from one file to another
B. A class acquiring properties from parent
C. Renaming a class to a new identifier
D. Deleting unused classes from program

Answer: B

Q244. What is encapsulation?

A. Running multiple objects at the same time
B. Converting objects into primitive data type values
C. Splitting a class into multiple smaller classes
D. Bundling data and methods while restricting access

Answer: D

Q245. What is a constructor in a class?

A. A function that converts objects to strings
B. A variable that stores the class name text
C. A special method called when object is created
D. A method that deletes the object from memory

Answer: C

Q246. What is a method in OOP?

A. A comment describing the class purpose
B. A variable declared inside a class body
C. A function defined within a class body
D. A data type used only in class files

Answer: C

Q247. What does the keyword 'this' refer to in a class?

A. The first parameter of any method
B. The current object instance itself
C. The main method of the application
D. The parent class of the current class

Answer: B

Q248. What is polymorphism in OOP?

A. Creating multiple constructors in a class
B. Deleting objects after they are used once
C. Storing different data types in one array
D. Objects taking many forms via shared interface

Answer: D

Q249. What is an attribute or property of a class?

A. A variable that holds data within a class
B. A method that performs calculations on data
C. A class that inherits from another class
D. A keyword that defines the class access

Answer: A

Q250. What is an interface in OOP?

A. A tool for debugging object-oriented programs
B. A type of variable used in class constructors
C. A contract specifying methods a class must implement
D. A graphical user interface for the application

Answer: C

Q251. What is RAM in a computer?

A. Volatile memory for temporary data storage
B. A processor that executes program instructions
C. A permanent storage device like hard drive
D. A network component for internet connection

Answer: A

Q252. What is the stack in memory management?

A. A region for storing program source code
B. A region for network data buffering use
C. A region for dynamically allocated objects
D. A region for function calls and local vars

Answer: D

Q253. What is the heap in memory management?

A. Memory for dynamic allocation at runtime
B. Memory for function call stack frames
C. Memory for operating system kernel data
D. Memory for storing constant values only

Answer: A

Q254. What is garbage collection?

A. Deleting old files from the file system
B. Defragmenting the hard drive storage
C. Automatic reclamation of unused memory
D. Manually freeing unused memory in code

Answer: C

Q255. What is a memory leak?

A. When a program uses too much CPU power
B. When a program writes to read-only memory
C. When data is lost during a power outage
D. When allocated memory is never freed up

Answer: D

Q256. What does memory allocation mean?

A. Reserving a block of memory for program use
B. Compressing data to use less memory space
C. Transferring memory between two computers
D. Deleting data from computer memory entirely

Answer: A

Q257. What is a pointer in programming?

A. A variable storing a memory address value
B. A function that points to another function
C. A graphical cursor on the screen display
D. A keyword used to create new variables

Answer: A

Q258. What is the difference between static and dynamic memory allocation?

A. Static is for objects; dynamic is for integers
B. Static is at compile time; dynamic is at runtime
C. Static is slower; dynamic is faster always
D. Static uses the heap; dynamic uses the stack

Answer: B

Q259. What happens when a program runs out of memory?

A. The program automatically frees old memory up
B. The program switches to using the CPU cache only
C. The operating system adds more physical RAM chips
D. The program crashes or throws an out-of-memory error

Answer: D

Q260. What is a null pointer?

A. A pointer that stores a numeric zero value
B. A pointer to the largest memory address
C. A pointer that does not point to anything
D. A pointer to the first element of array

Answer: C

Q261. What is an exception in programming?

A. A normal expected program execution result
B. An event disrupting normal program flow
C. A type of variable for storing text
D. A function for mathematical calculation

Answer: B

Q262. What does the try block do?

A. It tests the program for performance speed
B. It retries failed operations multiple times
C. It wraps code that might throw exceptions
D. It attempts to fix errors automatically

Answer: C

Q263. What does the catch block do?

A. It prevents all exceptions from occurring
B. It handles a specific type of exception
C. It catches user input from the keyboard
D. It stops the program from running further

Answer: B

Q264. What does the finally block do?

A. It runs only when no exception was thrown
B. It finalizes variable values permanently
C. It ends the program after error handling
D. It executes code regardless of exceptions

Answer: D

Q265. What does throwing an exception mean?

A. Preventing the program from starting at all
B. Deleting an error from the program memory
C. Signaling that an error condition has occurred
D. Converting an error into a warning message

Answer: C

Q266. What is a NullPointerException?

A. An error from dividing a number by zero
B. An error from accessing a null reference
C. An error from invalid string formatting
D. An error from array index being negative

Answer: B

Q267. What is an ArrayIndexOutOfBoundsException?

A. Deleting an element from an empty array
B. Creating an array with too many elements
C. Accessing an array with invalid index value
D. Accessing an array with valid index number

Answer: C

Q268. What is error handling?

A. The process of creating custom error messages
B. The process of hiding errors from the user
C. The process of writing error-free code always
D. The process of managing and responding to errors

Answer: D

Q269. What is the difference between an error and an exception?

A. Errors are recoverable; exceptions are not recoverable
B. Errors are serious system issues; exceptions are recoverable
C. Both errors and exceptions are exactly the same thing
D. Exceptions are more serious than errors in all cases

Answer: B

Q270. What is a stack trace?

A. A list of variables currently stored on stack
B. A report showing the sequence of method calls
C. A diagram of the program memory layout
D. A log of all user inputs during execution

Answer: B

Q271. What is a module in programming?

A. A type of variable for storing numbers
B. A comment explaining program functionality
C. A self-contained unit of reusable code
D. A single line of executable source code

Answer: C

Q272. What is a package in programming?

A. A collection of related modules grouped together
B. A compressed file containing program source code
C. A user interface component for displaying data
D. A single function within a larger program

Answer: A

Q273. What does the import statement do?

A. It exports code to another program file
B. It deletes unused variables from memory
C. It compiles the program into machine code
D. It brings external code into current scope

Answer: D

Q274. What is a library in programming?

A. A network protocol for data communication
B. A tool for designing graphical user interfaces
C. A database for storing user information data
D. A collection of pre-written reusable code

Answer: D

Q275. What is a namespace?

A. A file that lists all variables in the program
B. A container preventing naming conflicts in code
C. A variable that stores the program name string
D. A function that generates random names for vars

Answer: B

Q276. What is a dependency in software?

A. A function that runs only in debug build mode
B. A bug that depends on specific input to occur
C. A variable that depends on user input value
D. An external library that a program requires

Answer: D

Q277. What is the purpose of a package manager?

A. To test programs for bugs and performance
B. To compile source code into executable files
C. To install and manage software dependencies
D. To write new code automatically for developers

Answer: C

Q278. What is an API in the context of libraries?

A. The documentation website for the library only
B. The internal source code of the library files
C. The set of functions and rules for using library
D. The version number assigned to the library code

Answer: C

Q279. What does exporting a function from a module mean?

A. Copying the function into a separate backup file
B. Deleting the function from the current module
C. Making the function available to other modules
D. Converting the function into a different language

Answer: C

Q280. What is a framework in programming?

A. A testing tool for verifying program correctness
B. A type of database for storing application data
C. A single function for processing data values
D. A structure providing foundation for building apps

Answer: D

Q281. What is recursion in programming?

A. A function that calls itself to solve tasks
B. A variable that stores multiple data values
C. A loop that runs a fixed number of times
D. A technique for sorting array elements fast

Answer: A

Q282. What is a data structure?

A. A way to organize and store data efficiently
B. A network protocol for transmitting data
C. A programming language for data analysis
D. A tool for designing user interface layouts

Answer: A

Q283. What is a generic type in programming?

A. A type that can only store integer values
B. A type that cannot be used in functions
C. A type specific to one programming language
D. A type parameterized to work with any type

Answer: D

Q284. What is a design pattern?

A. A reusable solution to a common coding problem
B. A database schema for storing application data
C. A testing method for finding software bugs
D. A visual design for user interface screens

Answer: A

Q285. What is an iterator?

A. A counter variable used inside for loops
B. A class that stores a collection of objects
C. A function that sorts elements in an array
D. An object that traverses elements one by one

Answer: D

Q286. What is a linked list?

A. A sorted array with binary search capability
B. An array stored in non-contiguous memory blocks
C. A list of all links on a web page document
D. A sequence of nodes connected by pointers

Answer: D

Q287. What is a binary tree?

A. A tree with exactly two levels of depth total
B. A tree where each node has at most two children
C. A tree that stores only binary data zero and one
D. A tree that can only be traversed in one way

Answer: B

Q288. What is Big O notation used for?

A. Counting the number of lines in source code
B. Describing algorithm efficiency and scalability
C. Determining the memory size of data types used
D. Measuring the exact runtime in milliseconds

Answer: B

Q289. What is a graph data structure?

A. A set of nodes connected by edges or links
B. A chart displaying numerical data visually
C. A database table with rows and column fields
D. A type of array with two dimensions in it

Answer: A

Q290. What is an abstract class?

A. A class that cannot be instantiated directly
B. A class with no methods or properties defined
C. A class that exists only in pseudocode form
D. A class that is automatically garbage collected

Answer: A

Q291. What is concurrency in programming?

A. Using multiple monitors for one computer
B. Running one task at a time in sequence
C. Managing multiple tasks overlapping in time
D. Connecting multiple computers on a network

Answer: C

Q292. What is a thread in programming?

A. A connection between client and server
B. A lightweight unit of execution in process
C. A complete independent running program
D. A sequence of characters in a string

Answer: B

Q293. What is parallelism?

A. Writing code in two programming languages
B. Executing multiple tasks simultaneously on CPUs
C. Connecting computers in a parallel network
D. Running tasks one after another in order

Answer: B

Q294. What is a process in operating systems?

A. A single instruction executed by the CPU
B. An independent program instance in execution
C. A file stored on the hard drive disk
D. A connection between two network devices

Answer: B

Q295. What is a race condition?

A. A competition between programmers writing code
B. A performance test comparing two algorithms speed
C. An error when threads access shared data unsafely
D. A condition that causes programs to run too fast

Answer: C

Q296. What is a deadlock?

A. When a program produces incorrect output data
B. When the CPU overheats from excessive usage
C. When threads wait for each other indefinitely
D. When a program runs too slowly for the user

Answer: C

Q297. What is multithreading?

A. Running multiple threads within one process
B. Using multiple monitors for programming work
C. Writing code in multiple languages at once
D. Storing data in multiple databases together

Answer: A

Q298. What does synchronization mean in concurrent programming?

A. Coordinating thread access to shared resources
B. Converting synchronous code to asynchronous
C. Copying data between two different computers
D. Keeping system clocks at the same exact time

Answer: A

Q299. What is a mutex?

A. A lock ensuring only one thread accesses resource
B. A function for mathematical calculations only
C. A type of data structure for sorting elements
D. A variable that stores multiple values at once

Answer: A

Q300. What is the difference between concurrency and parallelism?

A. Concurrency is faster than parallelism in all execution scenarios
B. Concurrency runs tasks simultaneously; parallelism overlaps them
C. Both terms mean exactly the same thing in all programming contexts
D. Concurrency manages overlapping tasks; parallelism runs them at once

Answer: D

Q301. What is a unit test?

A. A test of the entire application system
B. A test of network connectivity speed
C. A test of a single small code component
D. A test of the user interface design layout

Answer: C

Q302. What is a breakpoint in debugging?

A. A marker indicating the end of a code block
B. A point where the code has a syntax error
C. A point where the program terminates normally
D. A marker where execution pauses for inspection

Answer: D

Q303. What is a bug in software?

A. An error causing unexpected program behavior
B. A security update for the application code
C. A comment left in the source code text
D. A feature that works exactly as intended

Answer: A

Q304. What does a debugger tool do?

A. It compiles source code into machine language
B. It designs the user interface for applications
C. It helps find and fix errors in program code
D. It automatically writes code for the developer

Answer: C

Q305. What is a test case?

A. The outer casing of a testing hardware device
B. A folder containing all source code files used
C. A specific scenario to verify expected behavior
D. A function that generates random test data sets

Answer: C

Q306. What is a syntax error?

A. A violation of programming language grammar rules
B. An error caused by network connection failure
C. A mistake in the user interface visual design
D. An error in the logic of the program algorithm

Answer: A

Q307. What does stepping through code mean?

A. Executing code one line at a time to inspect flow
B. Deleting unused lines of code from the program
C. Running the entire program as fast as possible
D. Skipping over sections of code during execution

Answer: A

Q308. What is a test suite?

A. A software tool for writing source code files
B. A single test case for one specific function
C. A room where testing equipment is kept stored
D. A collection of related test cases grouped together

Answer: D

Q309. What is a runtime error?

A. An error that occurs while the program is running
B. An error that prevents the IDE from opening up
C. An error in the project configuration file text
D. An error detected during code compilation phase

Answer: A

Q310. What is the purpose of print debugging?

A. To print the source code on paper for review
B. To display the final output to the end user
C. To print test results in a formatted report doc
D. To add print statements to trace program values

Answer: D

Q311. What is version control?

A. Limiting the number of code files in project
B. Controlling access permissions to the server
C. Controlling which version of language is used
D. Tracking and managing changes to source code

Answer: D

Q312. What is a code review?

A. Examining code by peers for quality and issues
B. An automated test that runs on every commit
C. A process of deleting unused code from files
D. A tool that formats source code automatically

Answer: A

Q313. What does DRY stand for in programming?

A. Deploy Run Yield in application lifecycle
B. Do Repeat Yourself for code clarity
C. Debug Review Yell in team collaboration
D. Don't Repeat Yourself in code writing

Answer: D

Q314. What is a branch in version control?

A. A copy of the main code for independent work
B. A tool for merging two programs into one
C. A comment added to track code change history
D. A permanent deletion of code from repository

Answer: A

Q315. What is refactoring?

A. Adding new features to existing program code
B. Restructuring code without changing its behavior
C. Removing all comments from the source code
D. Converting code to a different language entirely

Answer: B

Q316. What is a commit in version control?

A. Running the program to check for any errors
B. Deleting a file from the code repository
C. A request to merge code into main branch
D. A saved snapshot of changes to the codebase

Answer: D

Q317. What is agile software development?

A. A single long development phase with no feedback
B. An iterative approach with frequent delivery cycles
C. A technique for debugging code more efficiently
D. A method of writing code as fast as possible

Answer: B

Q318. What is documentation in software development?

A. The user interface of the application system
B. Written explanations of code and system design
C. The compiled binary output of the program
D. The source code of the program itself only

Answer: B

Q319. What is a pull request?

A. A request to review and merge code into a branch
B. A request to compile the code into executable
C. A request to delete code from the repository
D. A request to pull data from a database server

Answer: A

Q320. What is the purpose of coding standards?

A. To limit the number of lines per source file
B. To make all code run faster on every machine
C. To prevent any bugs from occurring in programs
D. To ensure consistent readable code across team

Answer: D

Q321. What is a variable in programming?

A. A named storage location in memory
B. A hardware component
C. A type of loop
D. A function that returns nothing

Answer: A

Q322. What is the purpose of indentation in Python?

A. It improves readability only
B. It defines code blocks and scope
C. It is used for commenting
D. It speeds up execution

Answer: B

Q323. Which of the following is an example of an interpreted language?

A. Assembly
B. Rust
C. C
D. Python

Answer: D

Q324. What is the output of a program?

A. The compiler error messages
B. The result produced by executing the program
C. The input given by the user
D. The source code written by the programmer

Answer: B

Q325. What does the term 'syntax' refer to in programming?

A. The rules governing the structure of code
B. The speed of program execution
C. The process of finding bugs
D. The meaning behind the code logic

Answer: A

Q326. What is a binary number system?

A. A system that uses hexadecimal digits
B. A system based on the number 8
C. A system that uses digits 0 through 9
D. A system that uses only 0 and 1

Answer: D

Q327. What is a logic error in a program?

A. An error that crashes the program immediately
B. An error caught by the compiler
C. An error where the program runs but produces wrong results
D. An error caused by missing semicolons

Answer: C

Q328. What is the role of an operating system for running programs?

A. It designs the user interface of programs
B. It writes the source code automatically
C. It compiles all programs into machine code
D. It manages hardware resources and provides services to programs

Answer: D

Q329. What is a keyword in a programming language?

A. Any word used as a variable name
B. A reserved word with a special meaning in the language
C. A comment in the source code
D. A user-defined function name

Answer: B

Q330. What does the term 'execute' mean in programming?

A. To delete a program from memory
B. To save a file to disk
C. To run a program and carry out its instructions
D. To write source code in an editor

Answer: C

Q331. What is an integer data type used for?

A. Storing true or false values
B. Storing whole numbers without decimal points
C. Storing text values
D. Storing images and multimedia

Answer: B

Q332. What is a floating-point number?

A. A number used only for counting
B. A number stored as text
C. A number that is always positive
D. A number that includes a decimal point

Answer: D

Q333. What is the byte data type typically used for?

A. Storing long text strings
B. Storing very large decimal numbers
C. Storing date and time values
D. Storing small integer values in the range 0 to 255

Answer: D

Q334. What does initializing a variable mean?

A. Deleting the variable from memory
B. Assigning a value to a variable for the first time
C. Changing the data type of a variable
D. Declaring the variable without a value

Answer: B

Q335. What is a long data type?

A. A type used exclusively for loop counters
B. An integer type with a larger range than int
C. A type that stores only positive numbers
D. A data type for storing very short strings

Answer: B

Q336. What is the difference between a literal and a variable?

A. A variable cannot change its value while a literal can
B. Variables are only used in loops
C. A literal is a fixed value written directly in code while a variable is a named storage location
D. Literals are only used for strings

Answer: C

Q337. What is an array data type?

A. A collection of elements of the same type stored in contiguous memory
B. A single value stored in memory
C. A type that stores only Boolean values
D. A special type of function

Answer: A

Q338. What does the short data type represent?

A. A text string with few characters
B. A fractional number with limited precision
C. An integer type with a smaller range than int
D. A Boolean value that is always false

Answer: C

Q339. What is a double data type?

A. An integer type with twice the range of long
B. A type that doubles the value stored automatically
C. A type that stores two variables at once
D. A floating-point type with double the precision of float

Answer: D

Q340. What is a reference type in programming?

A. A type that can only hold numeric values
B. A type used only in functional programming
C. A type that references external files
D. A type that stores the memory address of data rather than the data itself

Answer: D

Q341. What is an arithmetic operator?

A. An operator that combines Boolean values
B. An operator that assigns values to variables
C. An operator that performs mathematical calculations like addition and subtraction
D. An operator that compares two values

Answer: C

Q342. What is the result of 15 / 4 in integer arithmetic?

A. 3
B. 3.75
C. 4
D. 15

Answer: A

Q343. What does the less-than-or-equal operator (<=) check?

A. Whether two values are exactly equal
B. Whether the left value is less than or equal to the right value
C. Whether the left value is greater than the right value
D. Whether the left value is strictly less than the right value

Answer: B

Q344. What does the subtraction operator (-) do?

A. It adds two numbers together
B. It multiplies two numbers
C. It divides one number by another
D. It calculates the difference between two numbers

Answer: D

Q345. What is the decrement operator (--) used for?

A. Increasing a variable's value by one
B. Decreasing a variable's value by one
C. Setting a variable to zero
D. Doubling a variable's value

Answer: B

Q346. What is a comparison operator?

A. An operator that performs string concatenation
B. An operator that allocates memory
C. An operator that compares two values and returns a Boolean result
D. An operator that modifies the value of a variable

Answer: C

Q347. What does the division operator (/) do?

A. It compares two numbers
B. It finds the remainder of division
C. It divides the left operand by the right operand
D. It multiplies two numbers together

Answer: C

Q348. What is the compound assignment operator += used for?

A. Comparing whether two values are equal
B. Adding a value to a variable and storing the result back
C. Creating a new variable with an initial value
D. Concatenating two strings permanently

Answer: B

Q349. What does the equality operator (==) return?

A. A new variable with the combined value
B. True if two values are equal, false otherwise
C. The sum of two values
D. The larger of two values

Answer: B

Q350. What is the logical AND operator (&&) used for?

A. Returning true only when both operands are true
B. Negating a Boolean value
C. Adding two numbers together
D. Returning true when at least one operand is true

Answer: A

Q351. What is a conditional statement?

A. A statement that repeats code multiple times
B. A statement that terminates the program
C. A statement that executes code based on whether a condition is true or false
D. A statement that always executes regardless of conditions

Answer: C

Q352. What is the purpose of the 'elif' or 'else if' clause?

A. To define a new function
B. To declare a new variable
C. To check an additional condition when the previous if condition was false
D. To end a loop immediately

Answer: C

Q353. How many times does a for loop 'for(int i=1; i<=10; i++)' execute its body?

A. 11 times
B. 10 times
C. 9 times
D. 1 time

Answer: B

Q354. What is the default behavior when no break is used in a switch case?

A. The program terminates immediately
B. Execution falls through to the next case
C. Only the matched case executes
D. An error is thrown

Answer: B

Q355. What is a counter variable in a loop?

A. A variable that stores the exit code
B. A variable that holds the loop condition
C. A variable that stores the loop result
D. A variable that keeps track of how many iterations have occurred

Answer: D

Q356. What happens when a while loop condition is false from the start?

A. The loop body never executes
B. The loop body executes once before stopping
C. The loop runs infinitely
D. The program crashes with an error

Answer: A

Q357. What is the difference between a for loop and a while loop?

A. A for loop is used when the number of iterations is known while a while loop is used when it depends on a condition
B. A for loop cannot use counter variables
C. A for loop can only iterate 10 times
D. A while loop always runs at least once

Answer: A

Q358. What is a Boolean condition in an if statement?

A. A string that names the condition
B. A numeric value used for calculation
C. An expression that evaluates to true or false
D. A reference to another function

Answer: C

Q359. What does the 'return' statement do inside a loop?

A. It exits only the current loop
B. It skips to the next iteration of the loop
C. It restarts the loop from the beginning
D. It exits the entire function, ending the loop as well

Answer: D

Q360. What is the purpose of curly braces {} in control structures in languages like Java?

A. They are used for arithmetic grouping
B. They declare new variables
C. They define the block of code that belongs to the control structure
D. They indicate comments in the code

Answer: C

Q361. What is the purpose of a return value in a function?

A. To end the program immediately
B. To send a result back to the code that called the function
C. To declare a new variable
D. To print output to the console

Answer: B

Q362. What is a function prototype or declaration?

A. A comment describing what the function does
B. A statement that specifies the function name, return type, and parameters without providing the body
C. The complete implementation of a function
D. A test case for the function

Answer: B

Q363. What is the difference between a local variable and a global variable?

A. A local variable is accessible only within its function while a global variable is accessible throughout the program
B. Global variables can only store numbers
C. Local variables are always faster than global variables
D. Local variables persist longer than global variables

Answer: A

Q364. Can a function have multiple parameters?

A. Functions cannot have any parameters
B. No, functions can only have one parameter
C. Only built-in functions can have multiple parameters
D. Yes, functions can have multiple parameters separated by commas

Answer: D

Q365. What happens when a function calls itself?

A. It creates a new copy of the function in memory permanently
B. It creates an infinite loop that always runs forever
C. It performs recursion, which requires a base case to stop
D. The program crashes immediately

Answer: C

Q366. What is a method in the context of object-oriented programming?

A. A variable that stores a function reference
B. A module that contains multiple functions
C. A function that is defined inside a class and operates on its data
D. A standalone function not associated with any class

Answer: C

Q367. What is the difference between calling and defining a function?

A. Calling creates the function while defining runs it
B. Defining executes the function while calling creates it
C. There is no difference between them
D. Defining specifies what the function does while calling executes it

Answer: D

Q368. What is a nested function?

A. A function that never returns a value
B. A function defined inside another function
C. A function that only calls built-in functions
D. A function that takes no parameters

Answer: B

Q369. What is the purpose of the 'def' keyword in Python?

A. It deletes a variable from memory
B. It defines a new function
C. It defers execution to another thread
D. It declares a constant value

Answer: B

Q370. What is function reusability?

A. Copying a function from one language to another
B. Writing the same code in multiple places
C. Restarting the program to run functions again
D. Writing a function once and calling it from multiple places in the program

Answer: D

Q371. What is a file extension used for?

A. To encrypt the file contents
B. To increase the file size
C. To indicate the type or format of a file
D. To compress the file automatically

Answer: C

Q372. What does opening a file in write mode do?

A. It creates a new file or overwrites the existing file with new data
B. It locks the file from other users
C. It reads the contents of the file
D. It appends data to the end of the file

Answer: A

Q373. What does opening a file in append mode do?

A. It adds new data to the end of the file without removing existing content
B. It opens the file in read-only mode
C. It deletes all existing content before writing
D. It creates a backup of the file first

Answer: A

Q374. What is the purpose of closing a file after reading or writing?

A. To delete the file from disk
B. To convert the file to a different format
C. To encrypt the file contents
D. To release system resources and ensure all data is written

Answer: D

Q375. What is the difference between reading and writing a file?

A. Writing can only handle text files
B. Reading retrieves data from a file while writing sends data to a file
C. Reading modifies the file while writing preserves it
D. Reading is faster than writing in all cases

Answer: B

Q376. What is the System.out.println() method used for in Java?

A. Printing a line of text to the console with a newline at the end
B. Reading input from the keyboard
C. Creating a new string variable
D. Opening a file for writing

Answer: A

Q377. What is user input in programming?

A. The source code written by the programmer
B. The output displayed on the screen
C. Data provided by the user to the program during execution
D. Data that the program generates internally

Answer: C

Q378. What is the difference between stdout and stderr?

A. stdout handles errors while stderr handles normal output
B. stdout is for normal program output while stderr is for error messages
C. stderr is faster than stdout
D. They are the same output stream with different names

Answer: B

Q379. What does the readline() method do?

A. It deletes a line from a file
B. It counts the number of lines in a file
C. It reads a single line of text from input
D. It reads all lines from a file at once

Answer: C

Q380. What is a file handle or file descriptor?

A. A reference that the program uses to access an open file
B. The name of the file on disk
C. The size of the file in bytes
D. A physical part of the hard drive

Answer: A

Q381. What is a string literal?

A. A sequence of characters enclosed in quotation marks written directly in code
B. A variable that holds text
C. A number converted to text
D. A function that returns text

Answer: A

Q382. What does the replace() method do on a string?

A. It deletes the entire string
B. It converts the string to a number
C. It substitutes occurrences of a specified substring with another substring
D. It reverses the order of characters

Answer: C

Q383. What is the null terminator in C strings?

A. The character '\0' that marks the end of a string in C
B. A special character that marks the beginning of a string
C. A method that deletes a string from memory
D. A character used to separate words in a string

Answer: A

Q384. What does the indexOf() or find() method do on a string?

A. It removes a character at a specified position
B. It counts the total characters in the string
C. It inserts a character at a specified position
D. It returns the position of the first occurrence of a substring

Answer: D

Q385. What is the difference between single quotes and double quotes for strings?

A. Single quotes create mutable strings and double quotes create immutable strings
B. There is no difference in any programming language
C. Single quotes are for characters and double quotes for strings in languages like Java and C
D. Single quotes are faster to process than double quotes

Answer: C

Q386. What does the toLowerCase() method do?

A. It converts all characters in a string to lowercase letters
B. It counts the number of lowercase letters
C. It removes all lowercase letters from the string
D. It moves lowercase letters to the end of the string

Answer: A

Q387. What does string slicing allow you to do?

A. Extract a portion of a string by specifying start and end indices
B. Convert a string to an array
C. Delete parts of a string permanently
D. Join two strings together

Answer: A

Q388. What is the startsWith() method used for?

A. It starts the execution of a string function
B. It moves the first character to the end of the string
C. It creates a new string starting with a given character
D. It checks whether a string begins with a specified prefix

Answer: D

Q389. What does the contains() method check on a string?

A. Whether the string is empty
B. Whether the string is stored in memory
C. Whether the string contains a specified substring anywhere within it
D. Whether the string contains only numbers

Answer: C

Q390. What is string reversal?

A. Moving the first character to the end
B. Creating a new string with characters in reverse order
C. Deleting the last character of a string
D. Converting a string to uppercase

Answer: B

Q391. What is the length or size of an array?

A. The number of elements it contains
B. The amount of memory it uses in bytes
C. The maximum value stored in the array
D. The index of the last element

Answer: A

Q392. What is a dynamic array?

A. An array that sorts itself automatically
B. An array that changes its data type automatically
C. An array that only holds dynamic data
D. An array that can grow or shrink in size as elements are added or removed

Answer: D

Q393. What does the pop operation do on a stack?

A. It removes and returns the element from the top of the stack
B. It adds an element to the top of the stack
C. It clears all elements from the stack
D. It returns the bottom element without removing it

Answer: A

Q394. What is the enqueue operation in a queue?

A. Checking if the queue is empty
B. Adding an element to the back of the queue
C. Removing an element from the front
D. Sorting the elements in the queue

Answer: B

Q395. What is the dequeue operation in a queue?

A. Removing and returning the element from the front of the queue
B. Doubling the size of the queue
C. Adding an element to the front
D. Reversing the order of elements

Answer: A

Q396. What is a key-value pair?

A. A data structure that associates a unique key with a corresponding value
B. Two arrays linked together
C. Two functions that call each other
D. A pair of variables with the same data type

Answer: A

Q397. What is the difference between a sorted array and an unsorted array?

A. Sorted arrays use more memory than unsorted arrays
B. Sorted arrays can only store numbers
C. Unsorted arrays cannot be searched
D. A sorted array has elements arranged in order while an unsorted array has elements in arbitrary order

Answer: D

Q398. What is an empty collection?

A. A collection that only holds null values
B. A collection that has not been initialized
C. A collection that contains no elements
D. A collection that has been deleted from memory

Answer: C

Q399. What is the difference between a list and a set?

A. A list cannot be sorted
B. A set maintains insertion order while a list does not
C. A set can hold more elements than a list
D. A list allows duplicate elements and maintains order while a set contains only unique elements

Answer: D

Q400. What is the peek operation on a stack or queue?

A. It counts the number of elements
B. It adds a new element to the collection
C. It returns the top or front element without removing it
D. It removes the top or front element

Answer: C

Q401. What is a destructor in OOP?

A. A method that copies one object to another
B. A method that modifies all object properties
C. A special method called when an object is being destroyed to clean up resources
D. A method that creates new objects

Answer: C

Q402. What is an instance variable?

A. A variable that belongs to a specific object instance of a class
B. A variable shared by all objects of a class
C. A variable that cannot change its value
D. A variable declared outside any class

Answer: A

Q403. What is method invocation in OOP?

A. Calling a method on an object to execute its code
B. Deleting a method from a class
C. Defining a new method inside a class
D. Copying a method from one class to another

Answer: A

Q404. What is a subclass or child class?

A. A class that cannot be instantiated
B. A class that contains only static methods
C. A class that inherits properties and methods from another class
D. A class that has no methods

Answer: C

Q405. What is the 'private' access modifier?

A. It restricts access to only within the same class
B. It makes the member accessible from the same package
C. It allows access from any class in the program
D. It allows access from subclasses only

Answer: A

Q406. What is an abstract method?

A. A method declared without implementation that must be overridden by subclasses
B. A method that cannot be called
C. A method that returns abstract data types
D. A method with a complete implementation

Answer: A

Q407. What is object instantiation?

A. Defining a class in source code
B. Creating a new object from a class using a constructor
C. Deleting an object from memory
D. Copying all methods from one class to another

Answer: B

Q408. What is a getter method?

A. A method that retrieves the value of a private field
B. A method that sets the value of a private field
C. A method that creates a new field in the class
D. A method that deletes a field from the object

Answer: A

Q409. What is a setter method?

A. A method that sets or updates the value of a private field with optional validation
B. A method that removes a field from the object
C. A method that creates a new class
D. A method that returns the value of a field

Answer: A

Q410. What is the 'public' access modifier?

A. It allows access only from subclasses
B. It restricts access to the same class only
C. It makes the member read-only
D. It allows access from any class in the program

Answer: D

Q411. What is memory deallocation?

A. Allocating additional memory for a program
B. Compressing memory to save space
C. Moving data from RAM to the hard drive
D. Releasing memory that is no longer needed back to the system

Answer: D

Q412. What is the stack used for in program execution?

A. Storing function call information, local variables, and return addresses
B. Storing all program files on disk
C. Running the operating system
D. Permanently storing user data

Answer: A

Q413. Why is the heap used for dynamic memory allocation?

A. Because it automatically cleans up all allocations
B. Because it is the only available memory region
C. Because it allows allocating memory of any size at runtime that persists until explicitly freed
D. Because it is faster than the stack

Answer: C

Q414. What is a stack overflow?

A. When the heap runs out of space
B. When the stack exceeds its maximum size, typically due to deep or infinite recursion
C. When a variable stores a value that is too large
D. When the CPU overheats

Answer: B

Q415. What does the 'new' keyword do in languages like Java and C++?

A. It allocates memory on the heap and creates a new object
B. It deletes an existing object from memory
C. It declares a new variable on the stack
D. It opens a new file for writing

Answer: A

Q416. What is automatic memory management?

A. Memory that never needs to be allocated
B. A system where the runtime environment automatically handles memory allocation and deallocation
C. Memory that is managed by an external program
D. A tool for manually tracking memory usage

Answer: B

Q417. What is a reference in memory management?

A. A copy of an object stored in a different location
B. A comment in the source code
C. A value that points to or identifies the location of an object in memory
D. A backup of the original memory

Answer: C

Q418. What happens to local variables when a function returns?

A. They are moved to the heap
B. Their memory on the stack is automatically reclaimed
C. They become global variables
D. They are saved to disk permanently

Answer: B

Q419. What is the difference between manual and automatic memory management?

A. Manual is always faster than automatic
B. Automatic management cannot handle large programs
C. Manual requires the programmer to allocate and free memory while automatic uses garbage collection
D. Manual management is only used in interpreted languages

Answer: C

Q420. What is the purpose of the delete keyword in C++?

A. It deletes an entry from a database
B. It removes a variable declaration from the code
C. It frees memory that was allocated with new and calls the destructor
D. It deletes a source code file

Answer: C

Q421. What is the purpose of error handling in programming?

A. To speed up program execution
B. To detect and respond to errors gracefully without crashing the program
C. To prevent users from running the program
D. To add more features to the program

Answer: B

Q422. What is a FileNotFoundException?

A. An exception thrown when a program tries to access a file that does not exist
B. An error when the file has the wrong extension
C. An exception when the file system is full
D. An error when a file is too large to open

Answer: A

Q423. What is an IllegalArgumentException?

A. An exception when the compiler finds syntax errors
B. An error when the program has too many arguments
C. An error when the program runs out of memory
D. An exception thrown when a method receives an argument that is not valid

Answer: D

Q424. What happens if an exception is not caught?

A. The exception is saved to a file for later review
B. The exception is automatically fixed
C. The program terminates abnormally and typically displays an error message
D. The exception is ignored and the program continues

Answer: C

Q425. What is a ClassCastException?

A. An error when a class has too many methods
B. An exception when a class has duplicate names
C. An error when a class cannot be loaded
D. An exception thrown when trying to cast an object to a type it is not an instance of

Answer: D

Q426. Can a try block have no catch block?

A. Yes, a try block can exist alone without catch or finally
B. No, it must have both catch and finally
C. Yes, if it has a finally block instead
D. No, every try must have at least one catch

Answer: C

Q427. What is a NumberFormatException?

A. An exception when arithmetic operations fail
B. An exception thrown when trying to convert a non-numeric string to a number
C. An error when numbers exceed the integer range
D. An error when formatting numbers for display

Answer: B

Q428. What is the purpose of the 'catch' keyword?

A. It throws a new exception
B. It defines a block that handles a specific type of exception
C. It stops the program from running
D. It creates a new error object

Answer: B

Q429. What is an IndexOutOfBoundsException?

A. An exception when too many indices are created
B. An exception thrown when accessing a collection with an invalid index
C. An error when the file index is corrupted
D. An error when a database index is missing

Answer: B

Q430. What is the purpose of the 'raise' keyword in Python?

A. It raises the value of a variable
B. It increases the stack size
C. It throws or raises an exception manually
D. It increases the priority of a process

Answer: C

Q431. What is the purpose of importing a module?

A. To compile the module into machine code
B. To delete the module from the system
C. To make the functions and classes defined in the module available in the current file
D. To create a backup copy of the module

Answer: C

Q432. What is the __init__.py file used for in Python packages?

A. It creates a backup of the package
B. It stores initialization data for the program
C. It marks a directory as a Python package and can contain initialization code
D. It initializes all variables to zero

Answer: C

Q433. What is a third-party library?

A. A library created by someone other than the language developers or the project team
B. A library that requires three imports
C. A library stored in a third location
D. A library built into the programming language

Answer: A

Q434. What is the purpose of the 'export' keyword in JavaScript modules?

A. It sends data to an external server
B. It removes a variable from the module
C. It converts the module to a different format
D. It makes functions, classes, or variables available for other modules to import

Answer: D

Q435. What is a standard library?

A. A collection of modules and packages included with the programming language by default
B. A library that can only store standard data types
C. A physical library where programmers study
D. A library that follows a specific coding standard

Answer: A

Q436. What is a Maven repository in Java development?

A. A version control system for Java code
B. A storage location for Java libraries and dependencies that Maven can download automatically
C. A testing framework for Java modules
D. A database for storing Java application data

Answer: B

Q437. What does the 'require' function do in Node.js?

A. It imports and loads a module, returning its exported functionality
B. It checks if a condition is true
C. It validates that a file exists
D. It requires user authentication

Answer: A

Q438. What is the purpose of a .gitignore file in relation to packages?

A. It ignores all errors in the package
B. It specifies files and directories (like node_modules) that should not be tracked by version control
C. It hides packages from other developers
D. It imports Git-related packages

Answer: B

Q439. What is a framework in software development?

A. A single function imported from a library
B. A debugging tool for tracing program execution
C. A comprehensive package that provides a structure and set of tools for building applications
D. A physical frame used to hold a computer

Answer: C

Q440. What is the difference between a default export and a named export in JavaScript?

A. Named exports can only export functions
B. Default exports are faster than named exports
C. A default export allows importing without curly braces while named exports require them
D. Default exports must be constants

Answer: C

Q441. What is the Observer pattern used for?

A. Notifying multiple objects when the state of another object changes
B. Watching for syntax errors in code
C. Monitoring memory usage in real time
D. Observing CPU performance metrics

Answer: A

Q442. What is a hash function?

A. A function that validates user passwords
B. A function that maps input data to a fixed-size output value used for efficient data lookup
C. A function that encrypts all data permanently
D. A function that removes duplicate data

Answer: B

Q443. What is an event listener?

A. A hardware device that detects sound
B. A tool that monitors network events
C. A function that waits for and responds to a specific event in event-driven programming
D. A log file that records all system events

Answer: C

Q444. What is the MVC pattern?

A. A method of compressing video files
B. A security protocol for web applications
C. A version control system for managing code
D. A pattern that separates an application into Model (data), View (presentation), and Controller (logic)

Answer: D

Q445. What is time complexity?

A. The time needed to compile the program
B. A measure of how the running time of an algorithm grows relative to input size
C. The time a program takes to load from disk
D. The time taken to write the code

Answer: B

Q446. What is the main advantage of concurrent programming?

A. It reduces the amount of memory needed
B. It makes source code shorter
C. It allows programs to perform multiple tasks simultaneously, improving responsiveness and throughput
D. It eliminates all bugs from the program

Answer: C

Q447. What is a thread-safe data structure?

A. A data structure that can be safely accessed by multiple threads concurrently without data corruption
B. A data structure stored in thread-local memory
C. A data structure that can only be used in single-threaded programs
D. A data structure that is locked permanently

Answer: A

Q448. What does the 'join' method do on a thread?

A. It makes the calling thread wait until the target thread finishes execution
B. It joins the thread to a thread pool
C. It combines two threads into one
D. It creates a connection between two threads

Answer: A

Q449. What is a shared resource in concurrent programming?

A. A library used by multiple projects
B. A file that is shared on a network drive
C. A CPU core shared between programs
D. Data or resource that can be accessed by multiple threads or processes simultaneously

Answer: D

Q450. What is the start() method used for with threads?

A. It restarts a completed thread
B. It initializes thread-local variables
C. It starts the main program execution
D. It begins the execution of a new thread by calling its run method in a separate thread of execution

Answer: D

Q451. What is starvation in concurrent programming?

A. When a thread is perpetually denied access to a resource because other threads keep acquiring it first
B. When a program runs out of memory
C. When a CPU has no tasks to execute
D. When a thread pool runs out of threads

Answer: A

Q452. What is the run() method in a thread?

A. It contains the code that the thread will execute when started
B. It measures the thread's execution speed
C. It runs the entire program from the beginning
D. It runs diagnostics on the thread

Answer: A

Q453. What is a critical section in concurrent programming?

A. The most important part of the source code
B. A section of code that accesses shared resources and must not be executed by more than one thread at a time
C. A section of code that runs with highest priority
D. A code section that handles critical errors

Answer: B

Q454. What is the difference between a daemon thread and a user thread?

A. User threads cannot access shared resources
B. Daemon threads have higher priority than user threads
C. Daemon threads run faster than user threads
D. Daemon threads are background threads that do not prevent the program from exiting, while user threads keep the program alive

Answer: D

Q455. What is the sleep() method used for with threads?

A. It releases all thread resources
B. It pauses the thread for a specified duration before it resumes execution
C. It permanently stops the thread
D. It puts the entire program to sleep

Answer: B

Q456. What is the purpose of a test framework?

A. A physical frame for holding test devices
B. A tool that provides structure and utilities for writing and running automated tests
C. A framework for testing hardware components
D. A document that describes test procedures

Answer: B

Q457. What is the difference between a bug and an error?

A. Errors are more severe than bugs
B. A bug is a flaw in the code that causes incorrect behavior while an error is a mistake made by the programmer
C. They are exactly the same thing
D. Bugs only occur in production while errors occur in development

Answer: B

Q458. What does the 'step into' command do in a debugger?

A. It executes the current line and enters the called function to debug it line by line
B. It skips all remaining code in the function
C. It restarts the debugging session
D. It steps over the current line

Answer: A

Q459. What is a test assertion?

A. A legal assertion about software ownership
B. A claim that the software is bug-free
C. A statement that checks whether an expected condition is true, failing the test if it is not
D. A developer's confidence in their code

Answer: C

Q460. What is a watch expression in a debugger?

A. A log message that watches for errors
B. A timer that measures code execution speed
C. A scheduled check that runs periodically
D. An expression whose value is continuously monitored and displayed as you step through code

Answer: D

Q461. What is a test fixture?

A. A fixture that holds test equipment
B. A fixed baseline state or set of conditions set up before running tests
C. A hardware device used for testing
D. A test that runs automatically every day

Answer: B

Q462. What is the call stack in a debugger?

A. A list of all breakpoints set in the program
B. A display showing the chain of function calls that led to the current point of execution
C. A list of all variables in the program
D. A stack of error messages

Answer: B

Q463. What does a test pass or fail indicate?

A. A pass means the code compiled successfully, a fail means it did not
B. A pass means the code behaves as expected, a fail means the actual result differs from expected
C. A pass means the test ran quickly, a fail means it was too slow
D. A pass means no exceptions occurred regardless of correctness

Answer: B

Q464. What is the purpose of logging in debugging?

A. Recording program events, state, and data flow to help diagnose issues without using a debugger
B. Creating log files for regulatory compliance
C. Recording user login attempts
D. Logging the time spent on each function

Answer: A

Q465. What is a test runner?

A. A thread dedicated to running tests in the background
B. A tool that automatically discovers, executes, and reports the results of test cases
C. A person who manually runs all tests
D. A script that runs the program in test mode

Answer: B

Q466. What is a merge in version control?

A. Reverting all changes to the original state
B. Creating a new repository from scratch
C. Deleting a branch permanently
D. Combining changes from one branch into another

Answer: D

Q467. What is the purpose of code commenting?

A. To make the code run faster
B. To increase the file size for storage
C. To prevent code from being compiled
D. To add explanations that help developers understand the code's purpose and logic

Answer: D

Q468. What is a repository in version control?

A. A physical location where servers are stored
B. A collection of programming tutorials
C. A database for user information
D. A storage location containing the project files and their complete change history

Answer: D

Q469. What is the purpose of a README file?

A. To list all errors found in the code
B. To provide project documentation including description, setup instructions, and usage information
C. To read data from a file at runtime
D. To mark files as read-only

Answer: B

Q470. What is a coding convention?

A. A conference about programming
B. A legal contract for software development
C. An agreed-upon set of guidelines for writing code consistently within a team or project
D. A mathematical formula used in algorithms

Answer: C

Q471. What does 'git clone' do?

A. It creates a new empty repository
B. It copies a remote repository to the local machine including all history
C. It merges two repositories together
D. It deletes a remote repository

Answer: B

Q472. What is the purpose of an issue tracker?

A. Recording and managing bugs, feature requests, and tasks for a software project
B. Tracking performance issues in the CPU
C. Monitoring server uptime issues
D. Tracking internet connectivity issues

Answer: A

Q473. What is a deployment in software development?

A. The process of making a software application available for use in a specific environment
B. The process of hiring new developers
C. The process of designing the user interface
D. The process of writing new code

Answer: A

Q474. What is a staging environment?

A. An environment that mirrors production, used for final testing before releasing to users
B. A temporary development environment for writing code
C. An environment used exclusively for database operations
D. A backup environment for disaster recovery

Answer: A

Q475. What is the purpose of a .env file?

A. Configuring the text editor environment
B. Storing user session data
C. Encrypting the entire project
D. Storing environment-specific configuration variables like API keys and database URLs separately from code

Answer: D

Q476. What is the difference between a stack and a heap data structure?

A. A heap follows First-In-First-Out order
B. A heap is always faster than a stack
C. A stack can hold more elements than a heap
D. A stack follows Last-In-First-Out order while a heap is a tree-based structure that maintains a priority order

Answer: D

Q477. What is a queue data structure used for?

A. Randomly accessing elements by index
B. Processing elements in First-In-First-Out order where the first added element is removed first
C. Storing elements in sorted order
D. Storing elements in reverse order

Answer: B

Q478. What is the difference between linear search and binary search?

A. Binary search works on unsorted data while linear search requires sorted data
B. Linear search is always faster than binary search
C. Linear search checks each element sequentially while binary search repeatedly divides a sorted collection in half
D. Binary search checks every element one by one

Answer: C

Q479. What is a sorting algorithm?

A. An algorithm that arranges elements in a specific order such as ascending or descending
B. An algorithm that searches for a specific element
C. An algorithm that deletes duplicate elements
D. An algorithm that compresses data for storage

Answer: A

Q480. What is a callback function in event-driven programming?

A. A function passed as an argument to be executed when an event or operation completes
B. A function that always returns to the caller immediately
C. A function that calls the main function repeatedly
D. A function that calls back the operating system

Answer: A

Medium Questions

480 questions

Q481. What is the difference between compilation and interpretation in terms of error detection?

A. Compilation detects syntax errors before execution; interpretation at runtime
B. Interpretation detects every error before execution even begins
C. There is no practical difference between the two in error detection
D. Compilation detects all errors only at runtime during execution

Answer: A

Q482. Which generation of programming languages includes languages like C and Java?

A. First generation
B. Second generation
C. Fifth generation
D. Third generation

Answer: D

Q483. What is Just-In-Time (JIT) compilation?

A. Compiling code during execution for performance
B. Handling runtime errors through catch blocks
C. Compiling all code before shipping to users
D. Writing source code at the very last minute

Answer: A

Q484. What is the role of a linker in program compilation?

A. It combines object files into an executable
B. It interprets bytecode during execution
C. It validates syntax in the source code
D. It manages runtime memory allocation

Answer: A

Q485. What is the difference between strongly typed and weakly typed languages?

A. There is no real difference between the two type systems
B. Strongly typed languages allow implicit type conversions freely
C. Strongly typed enforces strict rules and restricts conversions
D. Weakly typed languages enforce very strict type conversion rules

Answer: C

Q486. What is bytecode?

A. Source code that is written using bytes
B. Encrypted form of the original source
C. Intermediate code run on a virtual machine
D. Binary code identical to machine code

Answer: C

Q487. What is the purpose of a preprocessor in C/C++?

A. To optimize generated machine code output
B. To process directives like #include and #define
C. To execute the compiled program directly
D. To link external libraries to the binary

Answer: B

Q488. Which of the following best describes a runtime error?

A. An error detected during the compilation phase
B. An error that occurs while the program executes
C. A syntax mistake caught before program runs
D. A warning generated by the development IDE

Answer: B

Q489. What is type inference?

A. A runtime error caused by wrong type assignment
B. The compiler automatically determining variable type
C. A debugging technique to inspect type values
D. Manually specifying every variable data type

Answer: B

Q490. What is the call stack used for in program execution?

A. Tracking function calls and local variables
B. Handling incoming network data requests
C. Storing all global variables and constants
D. Managing all file input/output operations

Answer: A

Q491. What is the difference between implicit and explicit type casting?

A. Implicit casting always causes errors while explicit casting never does
B. Implicit is done automatically by the compiler; explicit is done manually
C. Explicit casting executes much faster than implicit casting at runtime
D. They are exactly the same type conversion process with no differences

Answer: B

Q492. What is the size of an int in Java?

A. 32 bits
B. 16 bits
C. 8 bits
D. 64 bits

Answer: A

Q493. What is the difference between value types and reference types?

A. Value types store data directly; reference types store a memory address
B. They are completely identical in behavior and memory representation
C. Value types are immutable and cannot be modified after creation
D. Reference types always execute faster than value types at all times

Answer: A

Q494. What happens when you assign a float value to an int variable without casting?

A. It works fine with no issues at all
B. Compilation error in strongly typed languages
C. The decimal part is automatically rounded up
D. The program crashes with a runtime exception

Answer: B

Q495. What is an enumeration (enum)?

A. A looping construct for iteration
B. A dynamically resizable array type
C. A function for string manipulation
D. A set of named constant values

Answer: D

Q496. What is the difference between signed and unsigned integers?

A. There is no meaningful difference between signed and unsigned types
B. Signed holds only positive values; unsigned holds negative values
C. They have different byte sizes in memory but the same value range
D. Signed holds negative and positive; unsigned holds non-negative only

Answer: D

Q497. What is a nullable type?

A. A type that is always null
B. A type with no default value
C. A type that cannot be null
D. A type that can hold a value or null

Answer: D

Q498. What is the result of integer overflow?

A. The value automatically becomes null
B. An exception is always thrown by runtime
C. The program always crashes immediately
D. The value wraps around in most languages

Answer: D

Q499. What is a union type in TypeScript?

A. A built-in type for numeric calculations
B. A type that can be one of several types
C. A type that combines properties of two types
D. A class that extends multiple parent classes

Answer: B

Q500. What is type aliasing?

A. Hiding a type from external modules
B. Renaming a declared variable in scope
C. Converting between different data types
D. Creating a new name for existing type

Answer: D

Q501. What is the difference between ++i and i++?

A. ++i always executes faster than i++ does overall
B. ++i increments before use; i++ increments after
C. There is no difference between them at all
D. i++ increments the value by two instead of one

Answer: B

Q502. What is short-circuit evaluation?

A. Stopping evaluation once result is known
B. A hardware feature of modern processors
C. Evaluating all conditions in every case
D. A compiler optimization for loop bodies

Answer: A

Q503. What is the ternary operator?

A. An operator that takes a single operand only
B. A conditional operator with three operands
C. A bitwise operator for binary manipulation
D. An operator that takes exactly two operands

Answer: B

Q504. What does the << operator do in C/Java?

A. Reads input from the console
B. Performs a left bitwise shift
C. Compares two values for equality
D. Performs string stream insertion

Answer: B

Q505. What is operator precedence?

A. The order in which operators are defined in code
B. The rules determining operator evaluation order
C. The execution speed of a particular operator
D. The data type returned by a given operator

Answer: B

Q506. What is the result of 5 & 3 (bitwise AND)?

A. 5
B. 15
C. 8
D. 1

Answer: D

Q507. What is the difference between == and === in JavaScript?

A. There is no difference between them
B. == checks value; === checks value and type
C. === is used for variable assignment
D. == checks type without value comparison

Answer: B

Q508. What does the XOR (^) operator return?

A. True when the operands are different
B. Always false regardless of the operands
C. True when both operands are the same
D. Always true regardless of the operands

Answer: A

Q509. What is the null coalescing operator (??) in JavaScript?

A. It checks the code for syntax errors at compile time
B. It returns the right operand if the left is null
C. It always returns a null value to the caller
D. It performs integer division on two operands

Answer: B

Q510. What is operator associativity?

A. How operators are stored internally within the memory
B. The direction operators of same precedence are evaluated
C. The underlying data type of an operator in the language
D. The execution speed of evaluating a given expression

Answer: B

Q511. What is the difference between while and do-while loops?

A. do-while is faster in execution speed compared to while loop
B. while always runs the loop body at least one time guaranteed
C. while checks condition first; do-while runs body at least once
D. There is no meaningful difference between them at all

Answer: C

Q512. What is an infinite loop?

A. A syntax error detected at compile time
B. A loop that has an empty body
C. A loop whose condition never becomes false
D. A loop that executes very fast

Answer: C

Q513. What is the purpose of the 'default' case in a switch statement?

A. It executes when no other case matches
B. It is required in every switch
C. It must always be first
D. It resets the switch value

Answer: A

Q514. What is fall-through in a switch statement?

A. Execution continues to next case without break
B. A compilation error in the switch block
C. The switch statement fails and throws an error
D. The default case gets skipped during execution

Answer: A

Q515. What is a sentinel value in loop control?

A. A special value signaling loop termination
B. A constant defined in the loop body
C. A randomly generated input value
D. A variable used as a loop counter

Answer: A

Q516. What is the time complexity of a nested loop where both run n times?

A. O(n)
B. O(2^n)
C. O(log n)
D. O(n^2)

Answer: D

Q517. What is a for-each loop?

A. A loop that counts its own iterations explicitly
B. A loop that runs indefinitely until interrupted
C. A loop that skips every other element in turn
D. A loop iterating over each element without index

Answer: D

Q518. What happens if you forget the break in a switch case in Java?

A. A compilation error is raised by the compiler immediately
B. Nothing happens and the switch works as normally expected
C. The program crashes with an unrecoverable runtime error
D. Fall-through: subsequent cases execute until a break is hit

Answer: D

Q519. What is a labeled break statement?

A. A break that exits a specific labeled outer loop
B. A break statement followed by a comment
C. A syntax error that the compiler will reject
D. A break statement that logs a debug message

Answer: A

Q520. What is loop unrolling?

A. Making loops intentionally execute more slowly for debugging
B. Reducing loop overhead by executing multiple iterations per pass
C. Adding additional iterations to increase the loop precision
D. Removing a loop from the code entirely

Answer: B

Q521. What is function overloading?

A. Calling a function too many times in a row
B. Same name functions with different parameter lists
C. A runtime error from excessive function calls
D. Calling a function from within itself recursively

Answer: B

Q522. What is the difference between pass by value and pass by reference?

A. Pass by value copies data; pass by reference passes address
B. Pass by value uses pointers to access the original variable
C. Pass by reference is always slower than pass by value mode
D. They are exactly the same thing with different names only

Answer: A

Q523. What is recursion?

A. A function calling itself
B. A loop construct
C. An error in programming
D. A type of variable

Answer: A

Q524. What is a base case in recursion?

A. The condition that stops recursion
B. The recursive call itself in code
C. An optional function parameter
D. The very first function call made

Answer: A

Q525. What is a lambda function?

A. An anonymous short inline function expression
B. A function that returns no value at all
C. A function that calls itself recursively
D. A function with an explicitly declared name

Answer: A

Q526. What is a callback function?

A. A function that calls itself in a recursive way
B. A function passed as an argument for later call
C. A constructor that initializes object properties
D. A function that contacts the user directly

Answer: B

Q527. What is variable scope in the context of functions?

A. The memory size of a variable in bytes
B. The declared data type of the variable
C. The current assigned value of a variable
D. The region where a variable is accessible

Answer: D

Q528. What are default parameters?

A. Parameters that are always initialized to zero
B. Parameters whose values cannot ever be changed
C. Parameters with values used when none is passed
D. Parameters that the caller must always provide

Answer: C

Q529. What is a higher-order function?

A. A function with high computational time complexity
B. A function defined in a higher-level parent class
C. A function that accepts many input parameters
D. A function that takes or returns other functions

Answer: D

Q530. What is method signature?

A. Only the access modifier of the declared method
B. The complete body of the method implementation
C. Only the return type of the declared method
D. The method name combined with its parameter list

Answer: D

Q531. What is buffered I/O?

A. I/O that applies encryption to all data before transferring
B. I/O that exclusively works with pre-allocated buffer arrays
C. I/O that collects data in a buffer to reduce system calls
D. I/O operations performed without any processing delays

Answer: C

Q532. What is the difference between print() and println() in Java?

A. There is no difference between them at all
B. print() executes faster than println() does
C. println() only prints numeric values to output
D. println() adds a newline; print() does not

Answer: D

Q533. What is file I/O?

A. Reading from and writing to disk files
B. Reading data from the keyboard input
C. Allocating memory for program variables
D. Communicating over the internet network

Answer: A

Q534. What is serialization?

A. Encrypting data for secure network transmission
B. Sorting data in ascending or descending order
C. Converting an object into a byte stream for storage
D. Compressing files to reduce their storage size

Answer: C

Q535. What does the flush() method do in I/O?

A. Deletes the entire output buffer contents
B. Forces buffered output to be written immediately
C. Clears the entire console screen display
D. Reads all contents from the input buffer

Answer: B

Q536. What is the difference between text mode and binary mode in file I/O?

A. Text mode processes files faster than binary mode does
B. There is no functional difference between the two modes
C. Text mode translates newlines; binary reads raw bytes
D. Binary mode is exclusively used for image file formats

Answer: C

Q537. What is standard error (stderr)?

A. A specialized tool used for debugging programs
B. A separate output stream for error messages only
C. An error that occurs in the standard input stream
D. It is the same stream as standard output stdout

Answer: B

Q538. What is the purpose of the 'with' statement for file I/O in Python?

A. It loops through all files in a given directory
B. It reads binary files into memory as byte arrays
C. It writes data to multiple files simultaneously
D. It ensures the file is properly closed after use

Answer: D

Q539. What is CSV and how is it used in I/O?

A. Comma-Separated Values, a text format for tabular data
B. A relational database management system for queries
C. A compiled programming language for systems work
D. A binary file format for storing structured records

Answer: A

Q540. What is piping in command-line I/O?

A. A specialized type of network I/O communication
B. Writing output data to a physical pipe file
C. Redirecting one program output as another input
D. Processing multiple data streams in parallel mode

Answer: C

Q541. What is the difference between mutable and immutable strings?

A. Mutable strings always execute faster than immutable string types
B. Immutable strings can dynamically grow in size without any limit
C. There is no meaningful difference between the two string types
D. Mutable strings can be modified in place; immutable ones cannot

Answer: D

Q542. What is a regular expression?

A. A standard normal string literal value
B. An algorithm for sorting string collections
C. A pattern for matching and manipulating text
D. A mathematical expression using operators

Answer: C

Q543. What is string interpolation?

A. Converting strings to their numeric equivalents
B. Splitting strings by a delimiter character
C. Comparing two strings for equality or ordering
D. Embedding variables directly within a string

Answer: D

Q544. What is the String Pool in Java?

A. A thread pool dedicated to string processing
B. A mutable buffer for building string values
C. A collection of string utility methods
D. A heap area that caches string literals for reuse

Answer: D

Q545. What is the difference between String, StringBuilder, and StringBuffer in Java?

A. StringBuilder is immutable while String and StringBuffer are both mutable
B. They are all identical in function and performance with no differences
C. String is immutable; StringBuilder is mutable; StringBuffer is thread-safe
D. StringBuffer is always the fastest option among all three implementations

Answer: C

Q546. What does the split() method do?

A. Divides a string into an array of parts
B. Joins multiple strings into one
C. Converts string to a character array
D. Removes specified characters from string

Answer: A

Q547. What is Unicode?

A. A type of data encryption algorithm
B. An extension of the ASCII character set
C. A universal character encoding standard
D. A specific font type used in displays

Answer: C

Q548. What is the difference between ASCII and Unicode?

A. ASCII supports many more characters than Unicode ever can
B. Unicode is the older standard that was developed before ASCII
C. They are exactly the same encoding standard with no differences
D. ASCII uses 7 bits for 128 chars; Unicode supports over 143,000

Answer: D

Q549. What is string encoding (like UTF-8)?

A. Compressing a string to reduce storage size
B. Encrypting a string for secure transmission
C. Converting a string value into an integer
D. Representing Unicode characters as byte sequences

Answer: D

Q550. What is a string builder pattern and when should you use it?

A. A design pattern for constructing user interface strings
B. A utility tool for building complex regular expressions
C. A parser that transforms strings into structured data types
D. Using a mutable class for efficient repeated concatenation

Answer: D

Q551. What is the time complexity of accessing an element by index in an array?

A. O(n^2)
B. O(1)
C. O(n)
D. O(log n)

Answer: B

Q552. What is a hash table?

A. A balanced binary tree for hierarchical storage
B. A sorted array of ordered elements in memory
C. A structure mapping keys to values via hashing
D. A stack variant that allows random element access

Answer: C

Q553. What is a collision in a hash table?

A. A runtime error in the hash function code
B. When a requested key is not found in map
C. When the hash table is completely full up
D. When two keys hash to the same index slot

Answer: D

Q554. What is the difference between a singly linked list and a doubly linked list?

A. Singly linked lists always use more memory than doubly linked
B. There is no meaningful difference between the two list types
C. Singly linked has next pointer; doubly linked has next and prev
D. Doubly linked lists are always slower than singly linked lists

Answer: C

Q555. What is a priority queue?

A. A sorted array with indexed element access only
B. A queue where higher-priority elements dequeue first
C. A stack with last-in first-out element ordering
D. A regular first-in first-out queue structure

Answer: B

Q556. What is the difference between HashMap and TreeMap in Java?

A. They are exactly the same data structure implementation
B. HashMap maintains the insertion order of all its entries
C. TreeMap is always faster than HashMap for all operations
D. HashMap is unordered O(1); TreeMap is sorted O(log n)

Answer: D

Q557. What is a tuple?

A. A mutable list that can change its elements
B. A key-value map for looking up data by key
C. An immutable ordered collection of mixed types
D. A sorted set that maintains element ordering

Answer: C

Q558. What is an iterator?

A. An array index used for element access
B. An object for traversing a collection sequentially
C. A sorting algorithm for ordering collections
D. A type of loop construct for repeating code

Answer: B

Q559. What is the difference between a shallow copy and a deep copy of a collection?

A. They are the same operation with identical results in every case
B. Shallow copy shares element references; deep copy duplicates all
C. Shallow copy is always slower than deep copy for all collections
D. Deep copy shares references while shallow copy duplicates objects

Answer: B

Q560. What is a deque (double-ended queue)?

A. A regular first-in first-out queue
B. A queue allowing insertion at both ends
C. A sorted list with binary search access
D. A stack variant for ordered elements

Answer: B

Q561. What is the difference between method overloading and method overriding?

A. They are the same concept with different names in different languages
B. Overloading requires an inheritance hierarchy between the classes involved
C. Overriding changes the method parameter list in the parent class definition
D. Overloading: same name different params; overriding: same signature in subclass

Answer: D

Q562. What is an abstract class?

A. A fully implemented class with all methods defined
B. A static class that only contains class-level members
C. A final class that cannot be extended by other classes
D. A class that cannot be instantiated with abstract methods

Answer: D

Q563. What is an interface?

A. A class with full method implementations provided
B. A contract defining method signatures without bodies
C. An abstract class with some methods implemented
D. A type of variable for storing object references

Answer: B

Q564. What is the difference between composition and inheritance?

A. Composition is always slower than inheritance in performance
B. Inheritance is an is-a relationship; composition is has-a
C. They are exactly the same concept with no difference at all
D. Inheritance is more flexible and loosely coupled overall

Answer: B

Q565. What is a static method?

A. A method belonging to the class not any specific instance
B. A constructor method for initializing new object state
C. A method that can never be overridden by subclasses
D. A private method accessible only within the same class

Answer: A

Q566. What are getter and setter methods?

A. Static methods belonging to the class directly
B. Methods for controlled field read and write access
C. Loop constructs for iterating over collections
D. Constructor types for initializing objects

Answer: B

Q567. What is the super keyword?

A. A global variable accessible everywhere in code
B. A reference to parent class for calling its members
C. The main class that serves as program entry point
D. A type of access modifier restricting member access

Answer: B

Q568. What is multiple inheritance and which languages support it?

A. Creating multiple instances of the same class simultaneously
B. Inheriting from a single parent class in a linear hierarchy
C. Inheriting from multiple parents; supported by C++ and Python
D. All programming languages support this feature universally

Answer: C

Q569. What is the diamond problem?

A. An error related to geometric shapes in rendering code
B. A memory leak issue caused by circular object references only
C. An ambiguity when inheriting from classes with common ancestor
D. A compilation warning about unused variables in the program

Answer: C

Q570. What is the Liskov Substitution Principle (LSP)?

A. A design pattern for constructing complex object hierarchies
B. A principle for sorting elements in a collection efficiently
C. A rule for managing memory allocation in large applications
D. Subclass objects should be substitutable for superclass objects

Answer: D

Q571. What is a dangling pointer?

A. A pointer referencing already freed memory
B. A pointer to a valid live object in memory
C. A pointer with a null value assigned to it
D. A smart pointer that manages object lifetime

Answer: A

Q572. What is reference counting as a garbage collection strategy?

A. Counting variables in the current scope of code
B. Tracking reference count and freeing at zero count
C. A manual memory management technique using counters
D. Counting the total number of memory blocks allocated

Answer: B

Q573. What is a buffer overflow?

A. Slow memory access due to fragmentation issues
B. Writing data beyond the allocated buffer boundary
C. Running out of available system memory completely
D. A type of memory leak from unreleased resources

Answer: B

Q574. What is the difference between malloc() and calloc() in C?

A. They are exactly the same function with no differences
B. malloc initializes all allocated memory bytes to zero
C. malloc allocates uninitialized; calloc initializes to zero
D. calloc is always faster than malloc in all scenarios

Answer: C

Q575. What is virtual memory?

A. Shared memory accessible by multiple processes at once
B. Cloud storage accessible over the internet network
C. Using disk space to extend memory beyond physical RAM
D. Encrypted memory for storing sensitive data securely

Answer: C

Q576. What is a memory pool?

A. A pre-allocated block for smaller allocations reducing fragmentation
B. A collection of physical memory hardware modules
C. A memory leak caused by unreleased dynamically allocated resources
D. A type of processor cache that stores frequently accessed data values

Answer: A

Q577. What is the difference between deep copy and shallow copy regarding memory?

A. Deep copy allocates new memory for nested objects; shallow copies refs
B. Shallow copy always uses more memory than deep copy for same structure
C. There is no difference in memory usage between the two copy methods
D. Deep copy uses less memory than shallow copy by sharing object refs

Answer: A

Q578. What are smart pointers in C++?

A. Regular raw pointers with no extra features at all
B. Objects that wrap pointers and manage memory automatically
C. Pointers that point to function addresses in the program
D. Pointers that find data faster through optimized algorithms

Answer: B

Q579. What is memory fragmentation?

A. Memory that has been encrypted for data security
B. Memory that has been compressed to save space
C. Corrupted memory caused by hardware failures in RAM
D. Free memory broken into small non-contiguous blocks

Answer: D

Q580. What is the RAII pattern?

A. A design pattern for building user interfaces
B. A type of garbage collection used in managed languages
C. Tying resource lifetime to object lifetime via destructors
D. A memory allocation algorithm for dynamic memory pools

Answer: C

Q581. What is the difference between checked and unchecked exceptions in Java?

A. Unchecked exceptions are always more severe than checked exceptions
B. Checked exceptions are always faster to handle than unchecked ones
C. Checked must be declared or caught; unchecked RuntimeExceptions do not
D. There is no meaningful difference between the two exception types

Answer: C

Q582. What is exception propagation?

A. Unhandled exception moving up the call stack to find a handler
B. Fixing an exception by catching and logging it
C. Logging all exceptions to a file for later review and analysis
D. Throwing multiple exceptions at the same time from one method

Answer: A

Q583. What is a custom exception?

A. A user-defined exception class for domain-specific errors
B. A syntax error detected during the compilation process
C. A warning generated by the IDE during code development
D. A built-in exception provided by the language runtime

Answer: A

Q584. What is the 'throws' keyword in Java?

A. A keyword for suppressing exceptions so they are ignored
B. A mechanism for catching and handling thrown exceptions
C. It is exactly the same as the throw keyword in Java
D. A declaration of which checked exceptions a method may throw

Answer: D

Q585. What is a stack trace?

A. A log file containing all program output and error data
B. The call stack data structure itself in memory
C. A memory dump of the entire program state at crash
D. A report showing the method call chain to an exception

Answer: D

Q586. What is the purpose of try-with-resources in Java?

A. To automatically close resources when the try block finishes
B. To try executing multiple methods in parallel simultaneously
C. To retry the try block harder if the first attempt fails
D. To allocate additional resources for the try block execution

Answer: A

Q587. What is the difference between Error and Exception in Java?

A. Errors are serious JVM problems; Exceptions are recoverable issues
B. Exceptions cannot be caught or handled by any catch block
C. They are the same thing with no meaningful differences at all
D. Errors are less serious and easier to handle than Exceptions

Answer: A

Q588. What is a try-except-else block in Python?

A. An advanced loop construct for iterating with error handling
B. try runs risky code; except handles; else runs if no exception
C. Invalid syntax that causes a compilation error
D. A triple try block for handling three exceptions at once

Answer: B

Q589. What is exception chaining?

A. Throwing many exceptions at the same time from one method call
B. Catching exceptions in a specific order from most to least severe
C. Wrapping one exception inside another to preserve original cause
D. A linked list of error objects stored in chronological order

Answer: C

Q590. What is the principle of 'fail fast'?

A. Testing the speed of program execution under various load levels
B. Detecting and reporting errors as soon as they occur in the program
C. Making programs crash as quickly as possible for testing purposes
D. Running programs as quickly as possible to improve the performance

Answer: B

Q591. What is dependency management?

A. Managing memory allocation for program variables only
B. Tracking and maintaining external libraries and versions
C. Sorting functions in alphabetical order within modules
D. Managing source code files in a project directory

Answer: B

Q592. What is a package.json file?

A. A database schema definition for SQL table structures
B. A log file recording application events and errors
C. A Node.js config file listing metadata and dependencies
D. A generic JSON data file for storing values

Answer: C

Q593. What is semantic versioning (SemVer)?

A. Alphabetical versioning using letter-based identifiers
B. Date-based versioning using release date as version
C. MAJOR.MINOR.PATCH scheme indicating type of changes
D. Random version numbers assigned without any rules

Answer: C

Q594. What is a virtual environment in Python?

A. A Docker container with a bundled application inside
B. An isolated Python environment with its own packages
C. A cloud-hosted computing environment for deployment
D. A full virtual machine running a complete OS image

Answer: B

Q595. What is the difference between static and dynamic linking of libraries?

A. Static includes library at compile time; dynamic loads at runtime
B. Dynamic linking is always slower than static linking in every case
C. Static linking uses less memory than dynamic linking at all times
D. There is no meaningful difference between the two linking methods

Answer: A

Q596. What is a circular dependency?

A. When two or more modules depend on each other
B. A circular buffer data structure for streaming
C. A loop in the code that repeats indefinitely
D. A recursive function calling itself repeatedly

Answer: A

Q597. What is tree shaking?

A. A data structure operation on binary tree nodes only
B. A sorting technique that organizes elements by weight
C. Removing trees from the project code repository
D. Removing unused dead code from the final build bundle

Answer: D

Q598. What is the purpose of a lock file (e.g., package-lock.json)?

A. To lock the project and prevent any code changes
B. To encrypt installed packages for secure data storage use
C. To record exact dependency versions for reproducible builds
D. To prevent anyone from making changes to the source code

Answer: C

Q599. What is the difference between devDependencies and dependencies in npm?

A. devDependencies are completely optional and never actually installed
B. dependencies are operating system packages not application libraries
C. There is no meaningful difference between the two dependency types
D. dependencies are for production; devDependencies for development only

Answer: D

Q600. What is a monorepo?

A. A repository containing only a single source file
B. A read-only repository that cannot accept any changes
C. A private repository with restricted access permissions
D. A single repository containing multiple related projects

Answer: D

Q601. What is the Observer pattern?

A. A logging pattern for recording application events to a file
B. A security pattern for authenticating users before granting access
C. A subject notifies dependent observers of state changes automatically
D. Watching code execute step by step in the debugger tool

Answer: C

Q602. What is the Factory pattern?

A. A singleton variant that creates exactly one object instance
B. A manufacturing system for producing physical goods
C. An iterator pattern for traversing collections sequentially
D. A pattern providing an interface for creating objects flexibly

Answer: D

Q603. What is the Strategy pattern?

A. A sorting method that orders elements in a specific sequence
B. A game strategy for winning competitions against opponents
C. A pattern defining interchangeable algorithms varying independently
D. A caching technique for storing frequently accessed data values

Answer: C

Q604. What is the difference between composition and inheritance as design strategies?

A. Composition offers flexibility and loose coupling; inheritance is tighter
B. They are identical design strategies with no practical differences
C. Inheritance is always better than composition in every design scenario
D. Composition is always slower in execution than inheritance-based designs

Answer: A

Q605. What is the Model-View-Controller (MVC) pattern?

A. A database pattern for organizing tables and their relationships
B. A testing framework for writing and running automated test suites
C. Separating application into Model (data), View (UI), Controller (logic)
D. A data structure for organizing elements in memory efficiently

Answer: C

Q606. What is lazy loading?

A. Loading resources in reverse chronological order
B. Deferring initialization until actually needed
C. A slow loading technique for large file downloads
D. Loading all data and resources at startup time

Answer: B

Q607. What is the difference between aggregation and composition in UML?

A. Composition implies weaker ownership than aggregation in all designs
B. Aggregation implies stronger ownership than composition in all cases
C. Aggregation implies weak ownership; composition implies strong ownership
D. They are identical concepts with no practical differences at all

Answer: C

Q608. What is the Iterator pattern?

A. Accessing collection elements sequentially without exposing internals
B. A search technique for finding elements in a sorted collection
C. A sorting algorithm for ordering elements in ascending sequence
D. A loop construct for iterating over elements in arrays

Answer: A

Q609. What is the Decorator pattern?

A. A user interface pattern for visual element styling
B. A logging tool for recording method execution traces
C. Adding annotations to class methods for documentation
D. Dynamically adding responsibilities to objects through wrapping

Answer: D

Q610. What is the Adapter pattern?

A. A data converter for transforming file format encodings
B. A sorting adapter for converting between sort algorithms
C. A pattern wrapping one interface to match another interface
D. A physical power adapter for electrical devices

Answer: C

Q611. What is a semaphore?

A. A counter-based primitive controlling shared resource access
B. A thread type for running background tasks in parallel
C. A simple flag variable for boolean state tracking
D. A type of lock for exclusive thread synchronization

Answer: A

Q612. What is thread safety?

A. A testing concept for validating thread performance
B. Code functioning correctly under multi-thread access
C. A debugging feature for analyzing thread behavior
D. Threads that are guaranteed to never crash or fail

Answer: B

Q613. What is a thread pool?

A. Pre-created worker threads reused for tasks
B. A memory pool for thread-local storage data
C. A collection of all threads in the system
D. A connection pool for database access threads

Answer: A

Q614. What is the producer-consumer problem?

A. A database issue with concurrent read and write transaction conflicts
B. A business problem related to supply chain management
C. A network problem caused by packet loss during data transmission
D. A synchronization problem with shared buffer between producers and consumers

Answer: D

Q615. What is an atomic operation?

A. An indivisible operation completing without interruption
B. A nuclear physics operation on atomic particles
C. A file operation that reads and writes data to disk
D. A database transaction that groups multiple queries

Answer: A

Q616. What is the volatile keyword in Java?

A. A final variable that cannot be modified after assignment
B. A keyword ensuring reads always come from main memory
C. A variable that changes randomly without any cause
D. A type of exception thrown during concurrent operations

Answer: B

Q617. What is a Future/Promise?

A. An object representing a value available in the future
B. A type of thread that executes tasks asynchronously
C. A variable that is declared for use at a later time
D. A timer that triggers a callback after a set delay

Answer: A

Q618. What is context switching?

A. Saving one thread state and loading another for CPU switching
B. Changing the scope of a variable from local to global
C. Changing between different programming languages in a project
D. Switching between different source code files in an editor

Answer: A

Q619. What is the dining philosophers problem?

A. A sorting problem for ordering elements in a collection
B. A classic problem illustrating resource allocation and deadlock
C. A database problem with concurrent transaction conflicts
D. A food-related algorithm for meal planning optimization

Answer: B

Q620. What is async/await?

A. A type of loop for iterating over collection elements
B. A testing pattern for validating asynchronous behavior
C. A syntax for writing async code that looks synchronous
D. A threading model for managing parallel execution of tasks

Answer: C

Q621. What is Test-Driven Development (TDD)?

A. A deployment strategy for releasing code to production
B. Testing code only after all development is complete
C. Writing tests before code to drive the design process
D. A debugging technique for finding runtime exceptions

Answer: C

Q622. What is code coverage?

A. How much source code has been written in total
B. The percentage of code lines executed during testing
C. A code review tool for examining code quality metrics
D. A security measure for protecting source code files

Answer: B

Q623. What is the difference between black-box and white-box testing?

A. There is no meaningful difference between the two testing approaches
B. Black-box tests functionality without code; white-box tests internals
C. White-box testing is fully automated while black-box is manual
D. Black-box testing is always better than white-box in every case

Answer: B

Q624. What is a mock object in testing?

A. A simulated object mimicking real behavior for tests
B. A real database connection used in production
C. A randomly generated object with unpredictable state
D. A deleted object that has been garbage collected

Answer: A

Q625. What is integration testing?

A. Testing interaction between multiple components together
B. Testing the user interface for visual correctness
C. Testing individual functions in complete isolation
D. Testing performance under heavy load conditions

Answer: A

Q626. What is regression testing?

A. Testing for performance regression and slower execution
B. Re-running tests after changes to ensure nothing broke
C. Testing only brand new features and functionality
D. Testing backward compatibility with previous versions

Answer: B

Q627. What is profiling in software optimization?

A. Writing documentation for the application source code
B. Analyzing execution to identify performance bottlenecks
C. Formatting code according to style guide conventions
D. Creating user profiles for application personalization

Answer: B

Q628. What is the difference between stubs and mocks?

A. Mocks are simpler and provide less functionality than stubs do
B. They are identical testing concepts with no differences at all
C. Stubs are more complex to implement than mocks in every case
D. Stubs provide predefined responses; mocks also verify interactions

Answer: D

Q629. What is a memory profiler?

A. A physical RAM module installed in the computer
B. A tool monitoring memory usage and detecting leaks
C. A compiler for translating source code to binaries
D. A debugger for stepping through code line by line

Answer: B

Q630. What is the difference between manual and automated testing?

A. There is no practical difference between the two testing methods
B. Manual testing is done by humans; automated uses scripts and tools
C. Automated testing is less reliable than manual in every case
D. Manual testing is faster than automated testing for all suites

Answer: B

Q631. What is Continuous Integration (CI)?

A. A deployment method for releasing code directly to production servers
B. Integrating code changes only once at the very end of development
C. A branching strategy for organizing feature development work in Git
D. Frequently merging code with automated builds and tests for feedback

Answer: D

Q632. What is Continuous Deployment/Delivery (CD)?

A. Automatically deploying or making release-ready every passing change
B. A testing strategy for validating software behavior before release
C. A branching model for organizing development work into branches
D. Manual deployment performed by operations team members only

Answer: A

Q633. What is the KISS principle?

A. A security principle for protecting sensitive user data
B. A greeting or expression of affection between people
C. A testing method for validating software behavior fully
D. Keep It Simple: avoid unnecessary complexity in design

Answer: D

Q634. What is technical debt?

A. Financial debt from purchasing technology equipment
B. Server costs for hosting applications in the cloud
C. The cost of future rework from choosing quick solutions
D. A type of bug caused by incorrect logic in the code

Answer: C

Q635. What is the YAGNI principle?

A. A testing approach for validating code correctness
B. A design pattern for creating reusable components
C. A programming language for building applications
D. Do not implement functionality until actually needed

Answer: D

Q636. What is pair programming?

A. Programming alone without any collaboration at all
B. Writing duplicate code for redundancy and safety nets
C. Two developers at one workstation: driver and navigator
D. Programming with matched pairs of source code files

Answer: C

Q637. What is a sprint in Scrum?

A. A fixed-length iteration for completing prioritized tasks
B. A deployment window for releasing features to production
C. A programming technique for optimizing code performance
D. Running fast to meet project deadlines under pressure

Answer: A

Q638. What is a code smell?

A. Malicious code injected by an attacker into the system
B. A runtime bug that crashes the application unexpectedly
C. A surface indication suggesting a deeper design problem
D. A syntax error detected during the compilation process

Answer: C

Q639. What is the difference between waterfall and agile methodologies?

A. Agile has no planning or structure and relies entirely on improvisation
B. There is no meaningful difference between the two methodologies at all
C. Waterfall is always faster than agile for delivering software to production
D. Waterfall is sequential with fixed phases; agile is iterative and flexible

Answer: D

Q640. What is a merge conflict?

A. A deployment error when releasing code to production servers
B. A runtime error that occurs during program execution in prod
C. A syntax error caused by incorrect code in the source file
D. When Git cannot auto-merge because branches changed same code

Answer: D

Q641. What distinguishes a strongly typed language from a weakly typed one?

A. Strongly typed languages do not support any string operations
B. Weakly typed languages cannot handle any numeric data types
C. Strongly typed languages enforce strict rules with fewer casts
D. Strongly typed languages run faster than weakly typed ones always

Answer: C

Q642. What is the difference between a logical error and a runtime error?

A. Logical errors crash the program; runtime errors give wrong results
B. Both types always occur at compile time before program even starts
C. Logical errors give wrong results; runtime errors halt execution
D. Runtime errors give wrong output; logical errors are always harmless

Answer: C

Q643. What is the role of a preprocessor in languages like C?

A. It processes directives like includes and macros before compiling
B. It manages memory allocation and garbage collection automatically
C. It links all object files together into one single final binary
D. It executes the program line by line during the runtime phase

Answer: A

Q644. What is the key advantage of using an interpreted language?

A. Interpreted languages cannot be used for web applications
B. Interpreted languages always produce faster executables overall
C. Interpreted languages enable platform-independent execution
D. Interpreted languages require a separate linking step always

Answer: C

Q645. What does bytecode represent in languages like Java?

A. The final machine code running directly on hardware
B. An intermediate code executed by a virtual machine
C. The original source code written by the programmer
D. A binary format used only for storing databases

Answer: B

Q646. How does just-in-time (JIT) compilation improve performance?

A. It compiles the entire program before any execution begins
B. It translates bytecode to native code at runtime for speed
C. It replaces the interpreter with a static linker entirely
D. It removes all comments and whitespace from source code

Answer: B

Q647. What is the purpose of a symbol table during compilation?

A. It maps identifiers to their types, scopes, and locations
B. It stores the graphical icons used in the program interface
C. It records user input received during program execution
D. It contains the complete list of syntax errors in the code

Answer: A

Q648. What is the difference between static and dynamic typing?

A. Static typing checks types at compile time; dynamic at runtime
B. Static typing is only for scripting; dynamic for systems coding
C. Static typing has no types; dynamic typing has strict types only
D. Static typing is slower; dynamic typing is always faster overall

Answer: A

Q649. What does lexical analysis do during compilation?

A. It links external libraries into the final binary
B. It checks the logic correctness of the algorithm
C. It optimizes the generated machine code for speed
D. It breaks source code into tokens for the parser

Answer: D

Q650. What is a cross-compiler used for?

A. It compiles code for a different target platform
B. It converts machine code back into source code
C. It interprets and compiles code at the same time
D. It compiles two languages into one single program

Answer: A

Q651. What is the risk of implicit type conversion in weakly typed languages?

A. It may produce unexpected results by silently converting
B. It prevents the programmer from using numeric data types
C. It always causes the program to crash immediately on start
D. It makes the program significantly faster than explicit casts

Answer: A

Q652. Why might a developer choose a double over a float?

A. A double provides greater precision for decimal values
B. A double is faster to process than a float on all CPUs
C. A double uses less memory than a float in all cases
D. A double can store text while a float cannot do that

Answer: A

Q653. What is an enumeration (enum) data type used for?

A. Creating dynamic arrays that resize automatically
B. Defining a set of named constant integer values
C. Encrypting sensitive string data in the program
D. Storing floating-point numbers with high precision

Answer: B

Q654. What happens when integer overflow occurs?

A. The program automatically switches to a larger type
B. The OS allocates additional memory for the variable
C. The compiler prevents the code from being compiled
D. The value wraps around to the minimum of its range

Answer: D

Q655. What is the difference between value types and reference types?

A. Value types require garbage collection; references do not
B. Value types are slower to access than reference types
C. Value types store data directly; references store addresses
D. Reference types cannot be used inside any functions

Answer: C

Q656. Why are floating-point numbers sometimes imprecise?

A. Computers cannot perform mathematical operations
B. Binary representation cannot express all decimal values
C. Languages intentionally introduce errors for speed
D. Floating-point numbers are always rounded to integers

Answer: B

Q657. What is a union data type in languages like C?

A. A type combining two arrays into one structure
B. A type that converts between different types
C. A type where all members share the same location
D. A type that stores multiple values simultaneously

Answer: C

Q658. What is type narrowing in TypeScript?

A. Restricting variables to only hold integer values
B. Refining a variable type within a conditional
C. Converting a large data type to a smaller one
D. Removing type information from variables at runtime

Answer: B

Q659. What is the purpose of the void type in programming?

A. To indicate a function returns no value at all
B. To represent null pointer references in memory
C. To declare variables with unspecified data type
D. To store empty string values in memory locations

Answer: A

Q660. How does a BigInteger type differ from a standard integer?

A. BigInteger uses less memory than a standard integer
B. BigInteger stores decimal fractions unlike standard int
C. BigInteger can represent arbitrarily large whole numbers
D. BigInteger is faster but less accurate than standard

Answer: C

Q661. What is the difference between prefix and postfix increment?

A. Prefix decrements and postfix increments the variable
B. Prefix returns value after increment; postfix returns before
C. Prefix works on integers only; postfix works on floats
D. Prefix is faster and postfix is slower in all languages

Answer: B

Q662. What does the ternary operator (?:) do?

A. It performs three separate arithmetic operations
B. It provides a shorthand for if-else expression
C. It creates three variables from one declaration
D. It loops through a collection exactly three times

Answer: B

Q663. What does the bitwise left shift operator (<<) do?

A. Moves bits left, effectively multiplying by powers of two
B. Rotates bits in circular pattern within byte boundaries
C. Moves bits right, effectively dividing by powers of two
D. Compares two values bit by bit and returns differences

Answer: A

Q664. What is short-circuit evaluation in logical expressions?

A. It skips the second operand when the result is determined
B. It converts logical expressions into arithmetic for speed
C. It evaluates all operands regardless of intermediate results
D. It caches the results of all Boolean operations for reuse

Answer: A

Q665. What is operator overloading in object-oriented programming?

A. Increasing operator precedence beyond default values
B. Combining multiple operators into one new operator
C. Defining custom behavior for operators with objects
D. Using more operators than necessary in expressions

Answer: C

Q666. What does the null coalescing operator (??) do?

A. It checks if a value is undefined and throws error
B. It converts null values to empty strings always
C. It prevents null values from being assigned anywhere
D. It returns the right operand when left is null

Answer: D

Q667. How does bitwise XOR (^) differ from bitwise OR (|)?

A. XOR returns one only when bits differ; OR either is one
B. XOR returns one when both bits are one; OR returns zero
C. XOR works on signed integers only; OR on unsigned ones
D. XOR does logical operations; OR does arithmetic only

Answer: A

Q668. What is the purpose of the instanceof operator in Java?

A. It creates a new instance of a class from constructor
B. It deletes an instance and frees allocated memory
C. It counts how many instances of a class exist now
D. It checks if an object is instance of a specific class

Answer: D

Q669. What does the spaceship operator (<=>) return?

A. Negative, zero, or positive based on comparison
B. The average of two numeric values as float result
C. A new object merging properties from two objects
D. A Boolean true or false based on strict equality

Answer: A

Q670. What is the difference between == and === in JavaScript?

A. Both perform the same comparison with no difference
B. === checks value only; == checks value and type
C. == checks value only; === checks value and type
D. == is for numbers only; === for strings and objects

Answer: C

Q671. What is the advantage of a for-each loop over a traditional for loop?

A. For-each loops can modify collection size in iteration
B. For-each loops work only with arrays not other types
C. For-each loops simplify iteration by abstracting index
D. For-each loops execute faster than for loops always

Answer: C

Q672. What is a sentinel value in loop control?

A. A special value that signals the loop should stop
B. A flag tracking whether the loop has been started
C. A variable counting the number of loop iterations
D. A constant defining maximum loop iteration count

Answer: A

Q673. What is fall-through behavior in a switch statement?

A. Execution continues into next case when break missing
B. The switch exits immediately after matching first case
C. The switch skips all cases and goes to default
D. The default case always executes before other cases

Answer: A

Q674. What is tail recursion and why is it important?

A. A technique to reverse order of recursive calls only
B. Recursion at the end of an array for faster access
C. A recursive call as last action enabling optimization
D. Recursion that always uses the tail element of list

Answer: C

Q675. When should a developer use a do-while loop instead of while?

A. When the loop condition depends on hardware signals
B. When the loop needs to run a predetermined count
C. When the loop should never execute if condition false
D. When the loop body must execute at least one time

Answer: D

Q676. What is the purpose of labeled break statements in Java?

A. They exit a specific outer loop from nested loops
B. They add descriptive labels for code documentation
C. They create named variables within loop block scope
D. They mark loops for automatic parallel JVM execution

Answer: A

Q677. What is a guard clause in programming?

A. A security check validating user authentication
B. A try-catch block placed around every function call
C. A mutex lock preventing concurrent shared data access
D. An early return handling edge cases at function start

Answer: D

Q678. How does pattern matching differ from a traditional switch?

A. Pattern matching is slower and less efficient than switch
B. Pattern matching can destructure and match complex shapes
C. Pattern matching cannot have a default case for unmatched
D. Pattern matching only works with integer and string values

Answer: B

Q679. What is the risk of deeply nested control structures?

A. They cause undefined behavior in all languages used
B. They always cause the program to run out of memory
C. They reduce readability and increase complexity greatly
D. They prevent the compiler from generating optimized code

Answer: C

Q680. What is a loop invariant and why is it useful?

A. A special loop type that cannot be modified in execution
B. An optimization removing unnecessary variables from loop
C. A condition true before and after each loop iteration
D. A constant declared outside the loop that never changes

Answer: C

Q681. What is the difference between pass-by-value and pass-by-reference?

A. Pass-by-value is faster than pass-by-reference in all cases
B. Pass-by-value sends a copy; reference sends original location
C. Both methods behave identically with no differences at all
D. Pass-by-reference creates a copy of the argument each time

Answer: B

Q682. What is a lambda function?

A. A small anonymous function defined inline in the code
B. A function that runs automatically at program startup
C. A function that cannot accept any parameters at all
D. A function that can only return Boolean values always

Answer: A

Q683. What is function overloading?

A. Defining multiple functions with same name different params
B. Replacing an existing function with a new one entirely
C. Calling a function more times than it can handle
D. Loading a function into memory before it is needed

Answer: A

Q684. What is a callback function?

A. A function that undoes the actions of another function
B. A function that calls back the previous function always
C. A function passed as an argument to another function
D. A function that only executes during error handling

Answer: C

Q685. What is a closure in programming?

A. A function that cannot be called more than once
B. A function that retains access to its enclosing scope
C. A function that closes the program when called
D. A function that blocks all other functions from running

Answer: B

Q686. What is a default parameter in a function?

A. A parameter with a preset value used if no arg given
B. A parameter that overrides all other parameter values
C. A parameter that must always be provided by caller
D. A parameter that is automatically deleted after use

Answer: A

Q687. What is a higher-order function?

A. A function that requires administrator access to run
B. A function that takes or returns other functions
C. A function with more than ten parameters defined
D. A function that runs with higher priority than others

Answer: B

Q688. What is the purpose of the main function in C or Java?

A. It handles all error messages in the entire program
B. It defines all variables used throughout the program
C. It serves as the entry point for program execution
D. It manages memory allocation for all data structures

Answer: C

Q689. What is a variadic function?

A. A function that can only be called once in program
B. A function that accepts a variable number of args
C. A function that changes its name during runtime
D. A function that returns different types each time

Answer: B

Q690. What is a pure function in functional programming?

A. A function written in a pure functional language only
B. A function that only works with primitive data types
C. A function with no side effects and deterministic output
D. A function that has been optimized for maximum speed

Answer: C

Q691. What is buffered I/O and why is it used?

A. It converts binary data to text format before file output
B. It compresses files automatically to save disk space used
C. It stores data in a buffer to reduce system call overhead
D. It encrypts all data before writing to prevent data loss

Answer: C

Q692. What is the purpose of flushing a buffer?

A. To convert buffer contents from binary to text encoding
B. To clear the buffer and discard all data within it
C. To force buffered data to be written to its destination
D. To increase the size of the buffer for more data storage

Answer: C

Q693. What is serialization in the context of I/O?

A. Converting objects into a storable or transmittable format
B. Executing I/O operations in a strict serial sequence
C. Assigning serial numbers to files for organization
D. Sorting data in sequential order before processing

Answer: A

Q694. What is the difference between synchronous and asynchronous I/O?

A. Synchronous blocks until complete; asynchronous does not block
B. Synchronous only works with network; asynchronous with files
C. Synchronous is always faster than asynchronous operations
D. Asynchronous cannot read files only write to them on disk

Answer: A

Q695. What is a stream in I/O programming?

A. A sequence of data elements available over time to read
B. A database connection for querying stored information
C. A graphical interface for displaying program output data
D. A fixed-size block of memory for storing data values

Answer: A

Q696. What is the purpose of the with statement for file I/O in Python?

A. It prevents other programs from accessing the same file
B. It converts file contents to uppercase text characters
C. It creates a new file with specific permissions on disk
D. It ensures the file is properly closed after operations

Answer: D

Q697. What is CSV format used for?

A. Compressing large files to reduce their disk size
B. Storing executable program code in text files
C. Encrypting sensitive data for secure transmission
D. Storing tabular data with comma-separated values

Answer: D

Q698. What is the difference between absolute and relative file paths?

A. Absolute paths can only reference files not directories
B. Absolute starts from root; relative from current directory
C. Absolute is shorter than relative paths in all cases
D. Relative paths work across different operating systems only

Answer: B

Q699. What is JSON and why is it popular for data interchange?

A. A binary format designed for storing image data on disk
B. A protocol for encrypting data during network transfer
C. A programming language for building web applications
D. A lightweight text format that is easy to read and parse

Answer: D

Q700. What does the file seek operation do?

A. It creates a backup copy of the file on another disk
B. It moves the read/write position to a specific location
C. It searches for a specific string within file content
D. It locks the file to prevent concurrent access by others

Answer: B

Q701. What is the difference between mutable and immutable strings?

A. Mutable strings use less memory; immutable use more always
B. Mutable strings are faster; immutable are always slower
C. Mutable strings can be changed; immutable cannot after creation
D. Mutable strings have a fixed length; immutable can grow

Answer: C

Q702. What is string interpolation?

A. Converting a string to a different character encoding
B. Splitting a string by a specific delimiter character
C. Embedding expressions directly inside a string literal
D. Comparing two strings character by character position

Answer: C

Q703. What is a regular expression used for?

A. Encrypting strings to protect sensitive data values
B. Converting strings between different data type formats
C. Defining patterns for searching and matching text
D. Storing mathematical formulas as text strings

Answer: C

Q704. Why is string concatenation in a loop often inefficient?

A. Because each concatenation creates a new string object
B. Because the loop counter interferes with string length
C. Because loops cannot process string data type values
D. Because concatenation requires network access each time

Answer: A

Q705. What is Unicode and why is it important for strings?

A. A standard encoding supporting characters from all languages
B. A programming language designed for text processing
C. A compression algorithm that reduces string memory usage
D. A sorting algorithm designed specifically for text strings

Answer: A

Q706. What does the split function do to a string?

A. It converts the string from uppercase to lowercase
B. It duplicates the string into two identical copies
C. It removes every other character from the string text
D. It divides a string into an array using a delimiter

Answer: D

Q707. What is the difference between StringBuilder and String in Java?

A. StringBuilder is slower than String for all text operations
B. StringBuilder is mutable for efficient string modifications
C. StringBuilder is immutable; String is mutable and changeable
D. StringBuilder cannot store text only numeric data values

Answer: B

Q708. What is a string pool in Java?

A. A sorting algorithm specifically designed for string arrays
B. A cache of string literals reused to save memory space
C. A collection of string utility functions in a library
D. A data structure for storing strings in sorted order

Answer: B

Q709. What is the purpose of the replace function on strings?

A. It substitutes occurrences of a pattern with new text
B. It moves the string to a different memory location
C. It renames the variable holding the string value
D. It converts the string from text into binary format

Answer: A

Q710. What is URL encoding for strings?

A. Replacing special characters with percent-encoded values
B. Compressing URL strings to reduce their total length
C. Encrypting URL strings to prevent unauthorized access
D. Converting URLs into string variables in the program

Answer: A

Q711. What is the time complexity of accessing an element by index in an array?

A. O(1) because arrays provide direct index-based access
B. O(log n) because arrays use binary search internally
C. O(n squared) because index calculation is quadratic
D. O(n) because it must search through all elements

Answer: A

Q712. What is a linked list and how does it differ from an array?

A. A linked list cannot store more than one hundred elements
B. A linked list is faster for index access than an array
C. A linked list uses nodes with pointers allowing dynamic size
D. A linked list stores elements contiguously like an array does

Answer: C

Q713. What is a hash map collision and how is it typically resolved?

A. When two values are identical in different hash map entries
B. When the hash map runs out of memory during insertion
C. When the hash function produces negative index values
D. When two keys map to the same bucket needing chain or probe

Answer: D

Q714. What is the difference between an ArrayList and a LinkedList in Java?

A. ArrayList stores only strings; LinkedList stores only numbers
B. ArrayList uses nodes with pointers; LinkedList uses an array
C. ArrayList provides fast index access; LinkedList fast insertion
D. ArrayList is immutable; LinkedList is mutable and changeable

Answer: C

Q715. What is a priority queue?

A. A queue that can only store integer data type elements
B. A queue where elements are processed in insertion order always
C. A queue that automatically removes duplicate element entries
D. A queue where elements are processed based on priority value

Answer: D

Q716. What is the purpose of the Collections.sort() method in Java?

A. It sorts the elements of a list in their natural ordering
B. It removes all duplicate elements from the given collection
C. It converts a collection into an array of primitive values
D. It reverses the order of elements in a collection object

Answer: A

Q717. What is a deque (double-ended queue)?

A. A queue that only allows deletion not insertion operations
B. A queue allowing insertion and removal at both front and back
C. A queue that stores exactly two elements at any given time
D. A queue that automatically sorts elements upon insertion

Answer: B

Q718. What is the advantage of a balanced binary search tree over a sorted array?

A. Balanced BST uses less memory than a sorted array always
B. Balanced BST has faster element access by index position
C. Balanced BST provides efficient insertion and deletion both
D. Balanced BST does not require any comparison operations

Answer: C

Q719. What is a sparse array?

A. An array where most elements are default or zero values
B. An array that stores elements in non-contiguous memory
C. An array that only contains string type data elements
D. An array with a fixed small number of total elements

Answer: A

Q720. What is the difference between a shallow copy and a deep copy of a collection?

A. Deep copies share nested references; shallow copies clone all
B. Shallow copies are slower to create than deep copies always
C. Shallow copies share nested references; deep copies clone all
D. Both shallow and deep copies produce identical independent results

Answer: C

Q721. What is the difference between an abstract class and an interface?

A. Abstract classes can have implementation; interfaces cannot always
B. Abstract classes cannot have fields; interfaces can have fields
C. Interfaces support multiple inheritance; abstract classes do not
D. Abstract classes and interfaces are functionally identical always

Answer: A

Q722. What is method overriding in OOP?

A. Creating a method with a different number of parameters
B. Deleting a parent method from the child class entirely
C. Defining a method with the same name in a child class
D. Calling a method from an unrelated class in the program

Answer: C

Q723. What is composition in OOP and how does it differ from inheritance?

A. Composition is slower than inheritance in all execution scenarios
B. Composition and inheritance produce identical code structure always
C. Composition cannot reuse code from other classes unlike inheritance
D. Composition uses has-a relationship; inheritance uses is-a relation

Answer: D

Q724. What is the purpose of access modifiers like public, private, protected?

A. They determine the execution speed of methods in the class
B. They control the visibility and accessibility of class members
C. They define the order in which methods are called in class
D. They specify the data type that a class member can store

Answer: B

Q725. What is the Liskov Substitution Principle?

A. Objects should have only one reason to change at any time
B. Subtypes must be substitutable for their base types safely
C. A class should be open for extension closed for modification
D. Classes should depend on abstractions not concrete classes

Answer: B

Q726. What is a static method in a class?

A. A method that cannot accept any parameters at all
B. A method that cannot be called from outside the class
C. A method belonging to the class itself not instances
D. A method that runs automatically when program starts

Answer: C

Q727. What is the diamond problem in multiple inheritance?

A. A class has too many methods causing diamond-shaped UML
B. A class cannot be instantiated due to circular references
C. A class inherits from itself creating an infinite loop
D. A class inherits from two classes sharing a common ancestor

Answer: D

Q728. What is dependency injection and why is it useful?

A. Providing dependencies externally rather than creating inside
B. Injecting code into a running program for hot patching
C. Adding extra methods to a class during program execution
D. Inserting breakpoints into code for debugging at runtime

Answer: A

Q729. What is the difference between aggregation and composition?

A. Aggregation has strong ownership; composition has weak ownership
B. Both aggregation and composition represent identical relationships
C. Aggregation has weak ownership; composition has strong ownership
D. Aggregation uses inheritance; composition uses interfaces only

Answer: C

Q730. What is the purpose of the final keyword in Java?

A. It prevents classes from being inherited or methods overridden
B. It specifies that a variable can only hold final data type
C. It marks a method as the first one to execute in class
D. It indicates the method will return the final calculated value

Answer: A

Q731. What is a dangling pointer and why is it dangerous?

A. A pointer that takes too long to dereference in code
B. A pointer that points to the beginning of an allocated block
C. A pointer that is declared but never used in the program
D. A pointer referencing freed memory causing undefined behavior

Answer: D

Q732. What is reference counting as a garbage collection strategy?

A. Tracking how many references point to each object in memory
B. Tracking how many times each function is called at runtime
C. Counting the total amount of memory allocated by program
D. Counting how many variables exist in the program scope

Answer: A

Q733. What is the difference between stack overflow and heap overflow?

A. Both stack and heap overflow are caused by the same exact problem
B. Stack overflow uses too much heap; heap overflow uses too much stack
C. Stack overflow is harmless; heap overflow always crashes program
D. Stack overflow is from deep recursion; heap from excess allocation

Answer: D

Q734. What is virtual memory and why do operating systems use it?

A. A type of memory used only by virtual machines and emulators
B. A memory encryption technique to protect sensitive program data
C. A technique using disk to extend available memory beyond RAM
D. Physical RAM that is faster than regular memory modules

Answer: C

Q735. What is a memory pool allocation strategy?

A. Pooling memory from multiple computers on a network
B. Storing all variables in a single continuous memory block
C. Allocating memory from a single global heap always
D. Pre-allocating fixed-size blocks for fast repeated allocation

Answer: D

Q736. What is the purpose of smart pointers in C++?

A. They convert pointers into integers for easier comparison
B. They automatically manage memory ownership and deallocation
C. They make pointer arithmetic faster on modern CPUs
D. They prevent pointers from being passed to any functions

Answer: B

Q737. What is memory fragmentation?

A. When free memory exists but is split into small unusable chunks
B. When memory is completely full with no free space left
C. When memory hardware develops physical defects over time
D. When a program accesses memory belonging to another process

Answer: A

Q738. What is a buffer overflow?

A. Writing more data than a buffer can hold overwriting adjacent memory
B. Reading data from a buffer that has already been fully emptied
C. Allocating a buffer that is larger than available memory space
D. Writing less data than a buffer can hold in its capacity

Answer: A

Q739. What is the RAII pattern in C++?

A. A pattern for parallel processing of arrays using multiple threads
B. A technique for reducing memory usage of string operations
C. A pattern for implementing recursive algorithms efficiently
D. Resource acquisition tied to object lifetime via constructor and destructor

Answer: D

Q740. What is the difference between malloc and new in C++?

A. malloc is faster than new in all performance benchmarks
B. malloc is for C++ only; new is for C language only
C. new cannot allocate memory for arrays unlike malloc can
D. malloc allocates raw memory; new also calls the constructor

Answer: D

Q741. What is the difference between checked and unchecked exceptions in Java?

A. Checked exceptions are less severe than unchecked exceptions
B. Checked exceptions occur at runtime; unchecked at compile time
C. Both checked and unchecked exceptions behave identically always
D. Checked must be declared or caught; unchecked are not required

Answer: D

Q742. What is exception chaining and why is it useful?

A. Catching multiple exceptions in a single catch block always
B. Throwing the same exception multiple times in succession
C. Creating a chain of try-catch blocks for different error types
D. Wrapping an exception in another to preserve the original cause

Answer: D

Q743. What is the purpose of custom exception classes?

A. They make exceptions run faster than built-in exception types
B. They automatically fix errors without any developer intervention
C. They replace all built-in exceptions in the language entirely
D. They provide domain-specific error types with relevant context

Answer: D

Q744. Why should you avoid using exceptions for normal control flow?

A. Because exceptions prevent the program from being compiled
B. Because exceptions cannot carry any data or message with them
C. Because exceptions are expensive and obscure program logic flow
D. Because exceptions only work inside the main function of program

Answer: C

Q745. What is the difference between throw and throws in Java?

A. throw is for checked exceptions only; throws for unchecked only
B. throw creates a new exception class; throws catches exceptions
C. throw raises an exception; throws declares exceptions a method may throw
D. Both throw and throws perform the exact same operation always

Answer: C

Q746. What is a try-with-resources statement in Java?

A. A try block that limits CPU resources used during execution
B. A try block that automatically imports required library classes
C. A try block that automatically closes resources when done
D. A try block that retries the operation with different data

Answer: C

Q747. What is the purpose of logging exceptions?

A. To convert exceptions into warnings that can be safely ignored
B. To automatically send exceptions to users via email alerts
C. To prevent exceptions from occurring in future executions
D. To record error details for debugging and monitoring purposes

Answer: D

Q748. What happens if an exception is thrown but not caught?

A. The exception is automatically converted to a return value
B. The program terminates and typically displays a stack trace
C. The exception is silently ignored and execution continues
D. The program pauses and waits for user input to continue

Answer: B

Q749. What is a multi-catch block in Java?

A. Nesting multiple catch blocks inside one another deeply
B. A catch block that runs multiple times for one exception
C. A single catch block handling multiple exception types
D. Catching exceptions from multiple try blocks at one time

Answer: C

Q750. What is defensive programming in relation to error handling?

A. Anticipating potential errors and handling them proactively
B. Only handling errors after they cause the program to crash
C. Ignoring all possible errors to keep the code base simple
D. Delegating all error handling to the operating system only

Answer: A

Q751. What is semantic versioning (SemVer)?

A. Dating versions with the release date as the number
B. Numbering versions sequentially with single integers
C. Naming versions with random words for easy recall
D. A system using MAJOR.MINOR.PATCH format for versions

Answer: D

Q752. What is dependency injection in the context of modules?

A. Providing module dependencies externally rather than hardcoding
B. Injecting malicious code into a module for exploitation
C. Removing unused dependencies from the project configuration
D. Automatically downloading dependencies from the internet

Answer: A

Q753. What is circular dependency and why is it problematic?

A. When a module depends on too many other external modules
B. When two modules have functions with the same name defined
C. When a dependency is updated too frequently breaking code
D. When module A depends on B and B depends on A causing issues

Answer: D

Q754. What is the difference between a static and dynamic library?

A. Static is linked at compile time; dynamic is linked at runtime
B. Static is for C only; dynamic is for all other languages only
C. Static is faster but dynamic uses less total memory always
D. Static cannot be updated; dynamic updates automatically always

Answer: A

Q755. What is a lockfile in dependency management?

A. A file that limits the number of dependencies allowed
B. A file that encrypts the dependencies for secure storage
C. A file that locks the source code from being modified
D. A file recording exact versions of all installed dependencies

Answer: D

Q756. What is tree shaking in module bundling?

A. Converting modules from one format to another at build
B. Reorganizing module code into a tree data structure
C. Sorting modules alphabetically for better organization
D. Removing unused code from the final bundle to reduce size

Answer: D

Q757. What is the difference between CommonJS and ES modules?

A. CommonJS is newer than ES modules in JavaScript evolution
B. CommonJS uses require/exports; ES modules use import/export
C. CommonJS uses import/export; ES modules use require/exports
D. Both systems use identical syntax and behave the same way

Answer: B

Q758. What is a monorepo?

A. A single repository containing multiple related projects
B. A repository containing only one small module of code
C. A repository that stores only documentation not code
D. A repository that cannot have more than one contributor

Answer: A

Q759. What is the purpose of a module bundler like Webpack?

A. It compresses images used in web application projects
B. It automatically writes unit tests for all module code
C. It combines multiple modules into optimized output bundles
D. It deploys applications to production servers directly

Answer: C

Q760. What is a transitive dependency?

A. A dependency of a dependency that is indirectly required
B. A dependency that transitions between different versions
C. A dependency that is optional and not always installed
D. A dependency that is only used during testing phases

Answer: A

Q761. What is the time complexity of binary search on a sorted array?

A. O(n squared) because it uses nested loops for searching
B. O(n) because it checks every element in the array
C. O(1) because it directly accesses the target element
D. O(log n) because it halves the search space each step

Answer: D

Q762. What is memoization and how does it improve performance?

A. It memorizes variable names for faster variable lookup
B. It logs all function calls to memory for later review
C. It caches function results to avoid redundant computation
D. It allocates extra memory to make functions run faster

Answer: C

Q763. What is the observer pattern in software design?

A. A pattern where one object observes system memory usage
B. A pattern where objects subscribe to and receive event updates
C. A pattern for creating exactly one instance of a class only
D. A pattern for sorting collections using comparison callbacks

Answer: B

Q764. What is dynamic programming?

A. Breaking problems into overlapping subproblems and caching
B. Creating programs that adapt to user preferences at runtime
C. Writing code that changes its own source during execution
D. Programming in a dynamically typed language like Python

Answer: A

Q765. What is a hash function and what makes a good one?

A. A function that encrypts data using a secret key for security
B. A function that compresses files to reduce their storage size
C. A function mapping data to fixed-size values with few collisions
D. A function that sorts data into alphabetical order always

Answer: C

Q766. What is the singleton pattern?

A. A pattern ensuring a class has only one instance globally
B. A pattern for connecting a single client to a server only
C. A pattern that creates multiple instances of different classes
D. A pattern that limits functions to one parameter at most

Answer: A

Q767. What is the factory pattern?

A. A pattern for converting objects between different types
B. A pattern for destroying objects when no longer needed
C. A pattern delegating object creation to a factory method
D. A pattern for mass producing identical objects in batches

Answer: C

Q768. What is the difference between BFS and DFS graph traversal?

A. BFS is only for trees; DFS is only for general graph structures
B. BFS explores depth first; DFS explores breadth first always
C. BFS uses a stack; DFS uses a queue for storing nodes to visit
D. BFS explores level by level; DFS explores as deep as possible

Answer: D

Q769. What is a heap data structure?

A. A complete binary tree satisfying the heap ordering property
B. A hash table that automatically resizes when it gets full
C. A linked list sorted in ascending order of element values
D. A disorganized pile of unsorted data in memory blocks

Answer: A

Q770. What is the strategy pattern?

A. A pattern for choosing between inheritance and composition
B. A pattern for planning the development timeline of projects
C. A pattern encapsulating interchangeable algorithms in objects
D. A pattern that strategically allocates memory for performance

Answer: C

Q771. What is a semaphore and how does it differ from a mutex?

A. A semaphore allows only one thread; a mutex allows multiple
B. A semaphore is for processes only; a mutex is for threads only
C. A semaphore allows multiple threads up to a count; mutex allows one
D. Both semaphores and mutexes behave identically in all situations

Answer: C

Q772. What is a thread pool and why is it used?

A. A pool of memory shared between all threads in a program
B. A data structure for storing thread execution results only
C. A group of pre-created threads reused for executing tasks
D. A debugging tool that monitors thread activity in programs

Answer: C

Q773. What are the four necessary conditions for a deadlock to occur?

A. Speed, memory, disk, and network resource limitations
B. Compilation, linking, loading, and execution phase errors
C. Input, processing, output, and storage operation failures
D. Mutual exclusion, hold-and-wait, no preemption, circular wait

Answer: D

Q774. What is the producer-consumer problem?

A. A problem of converting data between different encoding formats
B. A problem of coordinating threads that produce and consume data
C. A problem of allocating memory between different program modules
D. A problem about optimizing compilation speed of programs

Answer: B

Q775. What is an atomic operation?

A. An operation that completes indivisibly without interruption
B. An operation that uses atomic energy for faster processing
C. An operation that can be divided into smaller sub-steps
D. An operation that only works on integer data type values

Answer: A

Q776. What is the difference between preemptive and cooperative multitasking?

A. Both use identical scheduling mechanisms in all operating systems
B. Preemptive only works on single-core; cooperative on multi-core
C. Preemptive OS controls scheduling; cooperative tasks yield control
D. Preemptive is slower; cooperative is faster in all situations

Answer: C

Q777. What is a condition variable used for?

A. Storing the condition of an if-else statement in memory
B. Checking if a variable meets certain validation requirements
C. Allowing threads to wait for a specific condition to become true
D. Defining constant values that cannot be changed during runtime

Answer: C

Q778. What is the dining philosophers problem about?

A. Managing memory allocation in object-oriented program design
B. Optimizing database queries for better application performance
C. Designing efficient algorithms for sorting large data arrays
D. Illustrating deadlock and resource contention among processes

Answer: D

Q779. What is thread safety?

A. Code that functions correctly when accessed by multiple threads
B. A technique for making threads run faster on modern hardware
C. A security feature that prevents threads from being terminated
D. Writing code that runs only on a single thread at one time

Answer: A

Q780. What is the difference between blocking and non-blocking algorithms?

A. Blocking can cause threads to wait; non-blocking guarantee progress
B. Both types have identical performance in all concurrent scenarios
C. Blocking algorithms are faster; non-blocking are always slower
D. Non-blocking algorithms cannot be used with multiple threads

Answer: A

Q781. What is test-driven development (TDD)?

A. Writing tests after all code is complete and deployed
B. Developing tests and code simultaneously without any plan
C. Writing tests before code then implementing to pass tests
D. Testing only the most critical features of the application

Answer: C

Q782. What is code coverage in testing?

A. The total lines of code in the entire project codebase
B. The percentage of code executed during test suite runs
C. The number of developers who have reviewed the code
D. The amount of documentation written for the codebase

Answer: B

Q783. What is an integration test?

A. A test verifying a single function works in isolation
B. A test ensuring code compiles without any syntax errors
C. A test measuring the speed of algorithm execution time
D. A test verifying that multiple components work together

Answer: D

Q784. What is mocking in unit testing?

A. Replacing real dependencies with controlled fake objects
B. Ridiculing code that is poorly written by developers
C. Copying production data to use in test environments
D. Running tests multiple times to check for consistency

Answer: A

Q785. What is regression testing?

A. Testing the application on different operating systems platforms
B. Re-running tests to ensure changes have not broken existing code
C. Testing new features before they are implemented in code
D. Testing the program under extreme load and stress conditions

Answer: B

Q786. What is a profiler used for in optimization?

A. Measuring performance bottlenecks in program execution
B. Formatting source code to follow style guidelines set
C. Writing automated test cases for the program codebase
D. Translating code from one programming language to another

Answer: A

Q787. What is the difference between white-box and black-box testing?

A. Both approaches test code in exactly the same way always
B. White-box tests the UI; black-box tests the backend code
C. White-box is manual testing; black-box is automated testing
D. White-box tests with code knowledge; black-box without it

Answer: D

Q788. What is assertion in testing?

A. A statement that imports external libraries into the code
B. A statement that declares a variable with a specific type
C. A statement that verifies a condition is true during testing
D. A statement that prints debug information to the console

Answer: C

Q789. What is continuous integration (CI)?

A. Integrating third-party libraries into the project once
B. Continuously writing code without any breaks or pauses
C. Automatically building and testing code on every commit
D. Integrating all code at the end of the development cycle

Answer: C

Q790. What is a flaky test and why is it problematic?

A. A test that always fails consistently every time it runs
B. A test that only works on the original developer's machine
C. A test that passes or fails inconsistently without code changes
D. A test that is too slow to run during the CI pipeline build

Answer: C

Q791. What is continuous deployment (CD)?

A. Deploying all code changes at the end of the project
B. Continuously writing deployment scripts for the team
C. Deploying code only during scheduled maintenance windows
D. Automatically deploying every passing build to production

Answer: D

Q792. What is the SOLID principle in software design?

A. A single principle about writing solid unbreakable programs
B. A principle about making code run on solid-state hardware
C. Five design principles for maintainable object-oriented code
D. A principle requiring all code to be stored in solid files

Answer: C

Q793. What is technical debt?

A. The monetary cost of purchasing development tools and licenses
B. The amount of time spent learning new programming technologies
C. The implied cost of future rework from choosing quick solutions
D. The storage cost of maintaining large code repositories online

Answer: C

Q794. What is a microservice architecture?

A. An architecture using very small functions for all operations
B. A development practice of writing very concise source code
C. An architecture breaking an app into small independent services
D. A very small application with minimal functionality only

Answer: C

Q795. What is pair programming?

A. Two developers working together on the same code actively
B. Two programs running in parallel on the same computer
C. Two developers working on separate tasks simultaneously
D. Two teams competing to write the same feature faster

Answer: A

Q796. What is the purpose of a linter?

A. To encrypt source code to protect intellectual property
B. To convert code from one programming language to another
C. To analyze code for style issues and potential errors found
D. To compile source code into optimized machine code output

Answer: C

Q797. What is the difference between monolithic and microservice architectures?

A. Monolithic is one deployable unit; microservices are many separate
B. Both architectures deploy and scale in exactly the same way
C. Microservices use one language; monolithic uses many languages
D. Monolithic is faster; microservices are slower in all cases

Answer: A

Q798. What is a sprint in Scrum methodology?

A. A fixed-length iteration for completing a set of planned work
B. A quick fix applied to production code in emergency situations
C. A code review session held at the end of every work day
D. A performance test measuring application response time speed

Answer: A

Q799. What is the purpose of a retrospective in agile development?

A. To assign tasks to team members for the next sprint period
B. To review the final product before releasing to customers
C. To reflect on the process and identify areas for improvement
D. To estimate the effort required for upcoming development work

Answer: C

Q800. What is infrastructure as code (IaC)?

A. Managing infrastructure through machine-readable definition files
B. Converting infrastructure diagrams into programming source code
C. Writing code that runs on physical infrastructure hardware
D. Coding directly on production servers without version control

Answer: A

Q801. What is the difference between a compiler error and a linker error?

A. Linker errors happen before compiler errors in the build process
B. Compiler errors are warnings while linker errors crash the system
C. There is no difference between them
D. Compiler errors occur in syntax checking while linker errors occur when combining object files

Answer: D

Q802. What is the purpose of an object file in compilation?

A. It is the final executable program
B. It contains machine code that has not yet been linked
C. It stores the original source code
D. It holds only debug information

Answer: B

Q803. What is the difference between procedural and object-oriented programming?

A. Procedural languages cannot handle large programs
B. OOP cannot use functions at all
C. Procedural is faster than OOP in all cases
D. Procedural uses functions to organize code while OOP uses classes and objects

Answer: D

Q804. What is a semantic error in programming?

A. An error where the code has incorrect punctuation
B. An error where the code is syntactically valid but logically meaningless
C. An error caused by using the wrong file extension
D. An error that prevents compilation due to missing brackets

Answer: B

Q805. What is the role of a loader in program execution?

A. It loads the executable into memory and prepares it for execution
B. It links object files together into an executable
C. It compiles the source code into machine code
D. It checks the program for syntax errors

Answer: A

Q806. What is the difference between compiled and managed languages?

A. Managed languages must be interpreted line by line
B. Compiled languages produce native code while managed languages run on a virtual machine
C. Compiled languages cannot use garbage collection
D. Managed languages are always slower than compiled languages

Answer: B

Q807. What is an abstract syntax tree (AST) used for?

A. Representing the hierarchical structure of source code during compilation
B. Drawing flowcharts of the program
C. Storing the final executable code
D. Managing memory allocation at runtime

Answer: A

Q808. What is the purpose of a garbage-collected runtime environment?

A. To optimize CPU instruction scheduling
B. To prevent all types of programming errors
C. To compile code faster than traditional compilers
D. To automatically reclaim memory that is no longer in use

Answer: D

Q809. What is the difference between a scripting language and a systems language?

A. Scripting languages cannot define functions
B. Systems languages are always object-oriented
C. Scripting languages prioritize ease of use while systems languages prioritize performance and control
D. Scripting languages can only run on web browsers

Answer: C

Q810. What is the purpose of intermediate representation (IR) in compilers?

A. To display the program output to the user
B. To serve as a platform-independent form between source code and machine code
C. To manage the file system during compilation
D. To encrypt the source code for security

Answer: B

Q811. What is autoboxing in Java?

A. Automatically compiling code without user intervention
B. Automatically converting primitive types to their wrapper class objects
C. Automatically resizing arrays when they are full
D. Automatically casting all types to strings

Answer: B

Q812. What is the difference between a weakly typed and a strongly typed language?

A. Strongly typed languages cannot perform arithmetic operations
B. Weakly typed languages are always interpreted
C. Weakly typed languages do not have data types at all
D. Weakly typed languages allow implicit type conversions while strongly typed languages restrict them

Answer: D

Q813. What is a tuple data type?

A. A type that stores exactly two values
B. An immutable ordered collection of elements that can have different types
C. A mutable list of elements of the same type
D. A type used only for database records

Answer: B

Q814. What is type coercion?

A. Explicitly casting a variable to a new type using syntax
B. Restricting a variable to a single type permanently
C. Automatic conversion of a value from one type to another by the language
D. Removing type information at compile time

Answer: C

Q815. What is the difference between mutable and immutable data types?

A. Mutable types can be changed after creation while immutable types cannot
B. Mutable types can only store numbers
C. Mutable types use more memory than immutable types always
D. Immutable types are only available in functional languages

Answer: A

Q816. What is a generic type parameter?

A. A placeholder for a type that is specified when the class or method is used
B. A type that is automatically inferred at runtime
C. A default type assigned when no type is specified
D. A type that can only hold generic string values

Answer: A

Q817. What is the purpose of the void type?

A. To represent a variable that holds zero
B. To declare a variable with no type
C. To create an empty array
D. To indicate that a function returns no value

Answer: D

Q818. What is a tagged union or discriminated union?

A. A type that stores one of several possible types along with a tag indicating which type is active
B. A type used only in database programming
C. A type that merges two arrays into one
D. A type that combines multiple unrelated types randomly

Answer: A

Q819. Why does 0.1 + 0.2 not equal 0.3 in most programming languages?

A. Because floating-point representation cannot exactly represent some decimal fractions
B. Because addition is not supported for decimal numbers
C. Because decimal numbers are stored as strings internally
D. Because the compiler rounds all numbers to integers

Answer: A

Q820. What is the difference between a struct and a class in C#?

A. Structs support inheritance but classes do not
B. Classes cannot be instantiated while structs can
C. Structs can have methods but classes cannot
D. Structs are value types allocated on the stack while classes are reference types allocated on the heap

Answer: D

Q821. What is the difference between logical OR (||) and bitwise OR (|)?

A. Logical OR works on numbers while bitwise OR works on Booleans
B. They produce identical results for all input types
C. Bitwise OR short-circuits but logical OR does not
D. Logical OR works on Boolean values and short-circuits while bitwise OR operates on individual bits

Answer: D

Q822. What is the purpose of the modulus operator in determining even or odd numbers?

A. If n % 2 equals 1 the number is even, if it equals 0 the number is odd
B. If n % 3 equals 0 the number is even
C. If n % 2 equals 0 the number is even, if it equals 1 the number is odd
D. The modulus operator cannot be used for this purpose

Answer: C

Q823. What does the bitwise NOT (~) operator do to an integer?

A. It sets all bits to zero
B. It converts the integer to its absolute value
C. It flips all the bits, changing 0s to 1s and 1s to 0s
D. It doubles the value of the integer

Answer: C

Q824. What is the optional chaining operator (?.) used for?

A. Creating optional function parameters
B. Safely accessing nested object properties without throwing if a reference is null
C. Defining nullable return types
D. Forcing a null value to throw an exception

Answer: B

Q825. What is the difference between the assignment operator (=) and the equality operator (==)?

A. Neither operator works with numeric values
B. The assignment operator stores a value in a variable while the equality operator compares two values
C. They perform the same function in all contexts
D. The equality operator stores values while the assignment operator compares them

Answer: B

Q826. How does the right shift operator (>>) work on signed integers?

A. It shifts bits to the right and fills the leftmost bits with the sign bit
B. It divides the number by 10
C. It shifts bits to the right and always fills with zeros
D. It reverses the order of all bits

Answer: A

Q827. What is the spread operator (...) used for in JavaScript?

A. Removing duplicate values from an array
B. Multiplying all elements in an array
C. Expanding an iterable into individual elements
D. Converting an object to a string

Answer: C

Q828. What is the typeof operator used for in JavaScript?

A. Returning a string indicating the type of the operand
B. Comparing two types for equality
C. Creating a new type at runtime
D. Casting a value to a specified type

Answer: A

Q829. What is the result of using the bitwise AND operator (5 & 6)?

A. 6
B. 4
C. 11
D. 5

Answer: B

Q830. What is the exponentiation operator (**) in Python and JavaScript?

A. It raises the left operand to the power of the right operand
B. It performs multiplication of two numbers
C. It performs integer division
D. It calculates the square root of a number

Answer: A

Q831. What is the advantage of using a switch statement over multiple if-else chains?

A. Switch statements automatically handle all possible cases
B. Switch statements can evaluate complex Boolean expressions better
C. Switch statements are clearer and more efficient when comparing a single variable against many constant values
D. Switch statements work with all data types in all languages

Answer: C

Q832. What is the difference between a pre-test loop and a post-test loop?

A. A pre-test loop checks the condition before each iteration while a post-test loop checks after
B. A pre-test loop runs faster than a post-test loop
C. A pre-test loop always runs at least twice
D. A post-test loop cannot use counter variables

Answer: A

Q833. What is a state machine and how does it relate to control flow?

A. A type of loop that runs indefinitely
B. A machine that stores all program variables
C. A hardware device that controls program execution
D. A model where behavior is determined by the current state and transitions triggered by events

Answer: D

Q834. Why should deeply nested control structures be avoided?

A. They reduce code readability and increase cognitive complexity for developers
B. They are not supported by modern compilers
C. They always cause runtime errors in production
D. They consume significantly more memory than flat structures

Answer: A

Q835. What is short-circuit evaluation in the context of control flow?

A. Evaluating only part of a Boolean expression when the result can be determined early
B. Skipping the entire if block when the condition is complex
C. Terminating the program when an error is detected
D. Running loops with fewer iterations than specified

Answer: A

Q836. What is the purpose of the 'yield' keyword in generators?

A. It pauses the function execution and returns a value, resuming from that point on the next call
B. It yields control to the operating system
C. It creates a new thread for the function
D. It terminates the function permanently

Answer: A

Q837. What is the difference between a counted loop and a conditional loop?

A. A conditional loop always runs at least once
B. A counted loop can only count up while a conditional loop counts down
C. A counted loop runs a predetermined number of times while a conditional loop runs until a condition changes
D. A counted loop does not use any variables

Answer: C

Q838. What is the ternary operator and how does it simplify control flow?

A. It creates three separate code branches
B. It allows three conditions to be checked simultaneously
C. It replaces a simple if-else with a concise single-line expression that returns one of two values
D. It replaces for loops with a shorter syntax

Answer: C

Q839. What is the purpose of the 'pass' statement in Python?

A. It skips to the next iteration of a loop
B. It passes a value to a function
C. It terminates the program gracefully
D. It serves as a placeholder where a statement is syntactically required but no action is needed

Answer: D

Q840. What is the difference between break and return inside a nested loop?

A. They behave identically in all situations
B. Break exits only the innermost loop while return exits the entire function
C. Break exits all nested loops while return exits only one
D. Break exits the function while return exits only the loop

Answer: B

Q841. What is the difference between named and anonymous functions?

A. Named functions are faster than anonymous functions
B. Anonymous functions cannot accept parameters
C. Named functions cannot be passed as arguments
D. Named functions have an identifier while anonymous functions are defined without a name

Answer: D

Q842. What is the purpose of the *args parameter in Python?

A. It creates a pointer to another function
B. It converts all arguments to strings
C. It restricts the function to accept only one argument
D. It allows a function to accept a variable number of positional arguments as a tuple

Answer: D

Q843. What is function composition?

A. Writing multiple functions in the same file
B. Inheriting functions from a parent class
C. Creating a function with many parameters
D. Combining two or more functions so the output of one becomes the input of another

Answer: D

Q844. What is the difference between synchronous and asynchronous function calls?

A. Synchronous calls are always faster than asynchronous calls
B. Synchronous calls block until completion while asynchronous calls return immediately and complete later
C. Asynchronous calls cannot return values
D. Synchronous calls run on multiple threads simultaneously

Answer: B

Q845. What is a decorator in Python?

A. A comment that describes a function
B. A tool for formatting output
C. A special syntax for defining classes
D. A function that wraps another function to modify or extend its behavior

Answer: D

Q846. What is the **kwargs parameter in Python used for?

A. It converts arguments to key-value pairs automatically
B. It blocks keyword arguments from being passed
C. It creates two copies of each argument
D. It accepts a variable number of keyword arguments as a dictionary

Answer: D

Q847. What is the purpose of the 'static' keyword for functions in Java?

A. It makes the function run faster
B. It prevents the function from being modified
C. It restricts the function to a single thread
D. It allows the function to be called without creating an instance of the class

Answer: D

Q848. What is an immediately invoked function expression (IIFE)?

A. A function that waits for user input before executing
B. A function that is defined and called at the same time
C. A function that immediately throws an exception
D. A function that is invoked only once per program run

Answer: B

Q849. What is the difference between a procedure and a function in traditional programming terminology?

A. A function returns a value while a procedure performs an action without returning a value
B. A procedure is faster than a function
C. A procedure can only contain one statement
D. A function cannot modify global state

Answer: A

Q850. What is the difference between character streams and byte streams in Java?

A. Character streams are faster than byte streams
B. Byte streams can only read files while character streams can read and write
C. Character streams handle text data using Unicode encoding while byte streams handle raw binary data
D. Character streams cannot handle non-English text

Answer: C

Q851. What is deserialization?

A. Deleting serialized data from disk
B. Compressing data for faster transmission
C. Converting a stored byte sequence back into an object in memory
D. Converting text to binary format

Answer: C

Q852. What is the purpose of a buffer in I/O operations?

A. To permanently store data on disk
B. To compress files before saving
C. To encrypt data during transfer
D. To temporarily hold data in memory to reduce the number of actual I/O operations

Answer: D

Q853. What is the difference between text mode and binary mode when opening a file?

A. Text mode translates line endings and handles encoding while binary mode reads raw bytes without translation
B. Text mode only works with ASCII characters
C. Binary mode is slower than text mode
D. Text mode is only for reading while binary mode is only for writing

Answer: A

Q854. What is the purpose of the StringIO class in Python?

A. It converts strings to integers
B. It encrypts string data for file storage
C. It provides a file-like interface for reading and writing strings in memory
D. It counts the number of I/O operations

Answer: C

Q855. What is the purpose of I/O redirection in command-line programs?

A. It speeds up I/O operations by using faster hardware
B. It redirects standard input, output, or error streams to files or other programs
C. It prevents programs from reading input
D. It converts all output to binary format

Answer: B

Q856. What is XML parsing and what are the two main approaches?

A. DOM is faster than SAX for all file sizes
B. DOM parsing loads the entire document into memory while SAX parsing processes it event by event
C. SAX loads the entire document while DOM reads it line by line
D. XML can only be parsed using regular expressions

Answer: B

Q857. What is the difference between formatted and unformatted I/O?

A. Formatted I/O is always faster than unformatted I/O
B. Unformatted I/O can only handle strings
C. Formatted I/O is only available in C
D. Formatted I/O converts data to human-readable text while unformatted I/O reads and writes raw binary data

Answer: D

Q858. What is the purpose of the encoding parameter when opening a file in Python?

A. It specifies how characters are converted to and from bytes when reading or writing
B. It encrypts the file contents for security
C. It compresses the file to save disk space
D. It sets the maximum file size allowed

Answer: A

Q859. What is the difference between polling and event-driven I/O?

A. Polling is always more efficient than event-driven I/O
B. Polling cannot handle multiple I/O sources
C. Polling repeatedly checks for I/O readiness while event-driven I/O notifies the program when I/O is ready
D. Event-driven I/O blocks the program until data arrives

Answer: C

Q860. What is the difference between String.valueOf() and toString() in Java?

A. valueOf() handles null by returning the string 'null' while toString() throws a NullPointerException on null
B. They are identical methods with different names
C. valueOf() only works with primitive types
D. toString() handles null better than valueOf()

Answer: A

Q861. What is a raw string literal and when is it useful?

A. A string where escape sequences are treated as literal characters, useful for regex and file paths
B. A string that is always stored in raw binary format
C. A string that cannot contain any special characters
D. A string that has not been processed by the compiler

Answer: A

Q862. Why is string concatenation in a loop inefficient for immutable strings?

A. Because each concatenation creates a new string object, copying all existing characters
B. Because the loop runs slower with string operations
C. Because the compiler cannot optimize string operations in loops
D. Because strings in loops are automatically encrypted

Answer: A

Q863. What is the purpose of the join() method for strings?

A. It connects two string variables permanently
B. It links a string to a file on disk
C. It concatenates an iterable of strings using a specified separator
D. It joins a string with a number

Answer: C

Q864. What is the difference between ASCII and extended ASCII?

A. Extended ASCII is a completely different encoding system
B. ASCII defines 128 characters using 7 bits while extended ASCII adds 128 more characters using a full 8 bits
C. Extended ASCII is only used for Asian languages
D. ASCII supports more characters than extended ASCII

Answer: B

Q865. What is a template literal (template string) in JavaScript?

A. A string enclosed in backticks that supports embedded expressions and multi-line text
B. A string that serves as a template for HTML documents
C. A string defined in a separate template file
D. A string that cannot be modified after creation

Answer: A

Q866. What is the purpose of the format() method or f-strings for string formatting?

A. They format the string as valid HTML
B. They change the font and styling of the string
C. They compress the string to reduce memory usage
D. They embed variable values and expressions directly within a string template

Answer: D

Q867. What is the difference between the equals() method and == for string comparison in Java?

A. equals() is faster than == for all cases
B. == compares content while equals() compares references
C. equals() compares content while == compares object references in memory
D. They perform identical comparisons

Answer: C

Q868. What is a character class in regular expressions?

A. A CSS class applied to text characters
B. A function that classifies characters as letters or digits
C. A class in OOP that represents a single character
D. A set of characters enclosed in square brackets that matches any single character from the set

Answer: D

Q869. What is the purpose of the strip() or trim() method beyond removing spaces?

A. It strips the string of all vowels
B. It only removes spaces and nothing else
C. It reduces the string to a fixed length
D. It removes specified leading and trailing characters, not just whitespace

Answer: D

Q870. What is the difference between open addressing and separate chaining for hash table collision resolution?

A. Separate chaining stores colliding elements in linked lists at each bucket while open addressing probes for the next empty slot
B. Separate chaining requires a sorted hash table
C. Open addressing uses linked lists while separate chaining probes for empty slots
D. They always produce identical performance characteristics

Answer: A

Q871. What is the time complexity of inserting an element at the beginning of a linked list?

A. O(log n) because it uses binary search
B. O(n) because all elements must be shifted
C. O(1) because only the head pointer needs to be updated
D. O(n^2) because of nested iterations

Answer: C

Q872. What is a circular buffer (ring buffer)?

A. A buffer that copies data in a circular pattern
B. A buffer shaped like a circle in memory
C. A fixed-size buffer that wraps around when the end is reached, reusing space from the beginning
D. A buffer that stores only circular data structures

Answer: C

Q873. What is the load factor of a hash table?

A. The ratio of stored elements to the total number of buckets
B. The speed at which elements are inserted
C. The maximum number of elements it can store
D. The amount of memory each element consumes

Answer: A

Q874. What is the difference between a min-heap and a max-heap?

A. Max-heaps can only store positive numbers
B. Min-heaps use less memory than max-heaps
C. Min-heaps are smaller than max-heaps
D. In a min-heap the smallest element is at the root while in a max-heap the largest element is at the root

Answer: D

Q875. What is the advantage of using a LinkedHashMap over a regular HashMap?

A. LinkedHashMap is faster for all operations
B. LinkedHashMap uses less memory than HashMap
C. HashMap cannot handle string keys
D. LinkedHashMap maintains insertion order or access order while HashMap does not guarantee order

Answer: D

Q876. What is an immutable collection?

A. A collection that automatically sorts its elements
B. A collection that cannot be modified after creation, providing thread safety and predictability
C. A collection that is stored in read-only memory
D. A collection that can only hold immutable objects

Answer: B

Q877. What is the difference between a stack-based and a queue-based approach to tree traversal?

A. A queue produces depth-first while a stack produces breadth-first
B. A stack traverses faster than a queue
C. A stack produces depth-first traversal while a queue produces breadth-first traversal
D. Both produce the same traversal order

Answer: C

Q878. What is the time complexity of searching for an element in a balanced binary search tree?

A. O(1) because elements are directly accessible by index
B. O(n) because every node must be checked
C. O(n log n) because it requires sorting first
D. O(log n) because the search space halves at each step

Answer: D

Q879. What is the Collections.unmodifiableList() method used for in Java?

A. It sorts the list permanently
B. It converts the list to an array
C. It returns an unmodifiable view of the list that throws an exception on modification attempts
D. It removes all elements from the list

Answer: C

Q880. What is the difference between an interface and an abstract class in Java?

A. They are identical constructs with different keywords
B. Abstract classes support multiple inheritance while interfaces do not
C. Interfaces can have constructors while abstract classes cannot
D. An interface defines method signatures without state while an abstract class can have both abstract and concrete methods with state

Answer: D

Q881. What is the Open/Closed Principle?

A. Software entities should be open for extension but closed for modification
B. All methods should be public (open) and fields should be private (closed)
C. Open source code should not be modified by closed source projects
D. Classes should be open for use and closed for testing

Answer: A

Q882. What is the purpose of the 'protected' access modifier?

A. It makes members accessible only within the same class
B. It allows access within the same class, subclasses, and same package
C. It prevents any access to the member
D. It makes members accessible only through getter methods

Answer: B

Q883. What is method hiding in Java?

A. Making a method invisible to other classes
B. Encrypting method implementations for security
C. Removing a method from a class at runtime
D. When a subclass defines a static method with the same signature as a static method in the superclass

Answer: D

Q884. What is the Single Responsibility Principle?

A. A program should have only one main class
B. A class should have only one reason to change, meaning it should have only one responsibility
C. Each method should have only one parameter
D. Each class should have only one constructor

Answer: B

Q885. What is the difference between shallow copy and deep copy of an object?

A. Shallow copy is always slower than deep copy
B. They produce identical results for all object types
C. Deep copy only copies primitive fields
D. Shallow copy copies only field values (sharing referenced objects) while deep copy recursively copies all referenced objects

Answer: D

Q886. What is a sealed class in Kotlin or Java?

A. A class that cannot have any methods
B. A class that restricts which classes can extend it, defining a closed set of subclasses
C. A class that cannot be imported into other packages
D. A class that seals its data from garbage collection

Answer: B

Q887. What is the Interface Segregation Principle?

A. All interfaces should be in a separate file
B. All classes must implement at least one interface
C. Interfaces should have exactly one method each
D. Clients should not be forced to depend on methods they do not use

Answer: D

Q888. What is the Dependency Inversion Principle?

A. Dependencies should always flow from child to parent classes
B. Methods should be invoked in reverse order of definition
C. All dependencies should be inverted to remove circular references
D. High-level modules should depend on abstractions rather than concrete implementations

Answer: D

Q889. What is a data class in Kotlin or a record in Java?

A. A class that stores only static data
B. A class that stores data in binary format
C. A class that automatically generates equals, hashCode, toString, and copy methods based on constructor properties
D. A class that can only be used for database operations

Answer: C

Q890. What is a memory-mapped file and how does it differ from traditional file I/O?

A. It maps a file directly into the process address space, allowing file access through memory operations instead of read/write calls
B. It stores file data in a special map data structure
C. It maps memory addresses to file names
D. It compresses files to fit in memory

Answer: A

Q891. What is the difference between strong references and weak references?

A. Weak references cannot be used with any data type
B. Strong references prevent garbage collection of the referenced object while weak references allow it
C. Weak references are faster than strong references
D. Strong references use more memory per reference

Answer: B

Q892. What is memory alignment and why does it matter?

A. Aligning data at memory addresses that are multiples of their size for efficient CPU access
B. Sorting data in memory alphabetically
C. Aligning code comments to the same column
D. Matching memory sizes between different programs

Answer: A

Q893. What is the purpose of the finalize() method in Java?

A. It finalizes the class definition preventing changes
B. It is called by the garbage collector before reclaiming an object's memory, but its use is discouraged
C. It permanently saves an object to disk
D. It marks an object as complete and ready for use

Answer: B

Q894. What is a double-free error?

A. Allocating twice the required memory
B. Attempting to free memory that has already been freed, causing undefined behavior
C. Using two different free functions on the same object
D. Freeing two different objects at the same time

Answer: B

Q895. What is a phantom reference in Java?

A. A reference to an object in another program
B. A reference that allows determining when an object has been garbage collected without preventing collection
C. An invisible reference that cannot be accessed
D. A reference to a deleted object

Answer: B

Q896. What is the difference between stack allocation and heap allocation in terms of performance?

A. Stack allocation is slower because it checks for overflow
B. They have identical performance characteristics
C. Heap allocation is always faster than stack allocation
D. Stack allocation is faster because it only requires moving the stack pointer, while heap allocation involves finding suitable free blocks

Answer: D

Q897. What is the purpose of soft references in Java?

A. They allow the garbage collector to reclaim objects only when memory is low, useful for caches
B. They create softly linked objects that cannot be modified
C. They create references that automatically become null after a timeout
D. They reference objects stored on soft storage like SSDs

Answer: A

Q898. What is the concept of ownership in Rust?

A. A design pattern for managing database connections
B. A system where each value has exactly one variable that owns it, controlling when the value is dropped
C. A way to assign copyright to code
D. A security feature that restricts file access

Answer: B

Q899. What is memory compaction in garbage collection?

A. Reducing the virtual memory size of a process
B. Moving live objects together to eliminate fragmentation and create contiguous free space
C. Compressing data to use less memory
D. Deleting all objects from memory

Answer: B

Q900. What is the difference between try-except-else and try-except-finally in Python?

A. The else block runs when an exception occurs while finally runs when it does not
B. The else block runs only when no exception occurs while the finally block runs regardless of whether an exception occurred
C. They are identical in behavior
D. The else block handles exceptions while finally throws them

Answer: B

Q901. Why is catching generic exceptions (like catch Exception) considered a bad practice?

A. Because generic exceptions take more memory than specific ones
B. Because generic exceptions are slower to catch than specific ones
C. Because the compiler does not allow generic exception catches
D. Because it catches all exceptions including ones the code cannot handle, hiding bugs and unexpected errors

Answer: D

Q902. What is exception filtering in C# (when clause)?

A. Adding a condition to a catch block that must be true for the catch to handle the exception
B. Filtering out exceptions based on their message text only
C. Removing exceptions from the program permanently
D. Blocking certain exception types from being thrown

Answer: A

Q903. What is the purpose of the suppressed exceptions feature in Java?

A. It attaches secondary exceptions that were suppressed during cleanup in try-with-resources to the primary exception
B. It reduces the number of exceptions a program can throw
C. It silently ignores all exceptions
D. It prevents exceptions from being logged

Answer: A

Q904. What is the difference between assertion errors and exceptions?

A. Assertions are faster than exceptions
B. Assertions replace the need for all exception handling
C. Exceptions cannot be caught while assertions can
D. Assertions check for programmer errors that should never happen while exceptions handle expected error conditions

Answer: D

Q905. What is the anti-pattern of using exceptions for control flow?

A. Using too few exception handlers in the program
B. Throwing exceptions from constructors
C. Catching exceptions and logging them
D. Using try-catch blocks for normal program flow instead of conditional checks, harming performance and readability

Answer: D

Q906. What is exception wrapping or translation?

A. Wrapping exception handling code in a separate class
B. Catching a low-level exception and throwing a higher-level exception that is more meaningful to the caller
C. Converting exception messages to different languages
D. Converting compile-time errors to runtime exceptions

Answer: B

Q907. What is the purpose of the 'finally' block in resource cleanup?

A. It guarantees that cleanup code (like closing files or connections) executes whether or not an exception occurred
B. It finalizes the data and sends it to the database
C. It prevents the exception from propagating further
D. It runs only when no exception is thrown

Answer: A

Q908. What is the difference between re-throwing and wrapping an exception?

A. Re-throwing is faster while wrapping provides more information
B. Re-throwing propagates the same exception unchanged while wrapping creates a new exception with the original as the cause
C. They are identical operations with different names
D. Re-throwing creates a new exception while wrapping keeps the same one

Answer: B

Q909. What is the role of the getMessage() method on exceptions?

A. It returns a human-readable description of the error that caused the exception
B. It sends an error message to the user via email
C. It creates a new exception with a custom message
D. It translates the exception into a log entry

Answer: A

Q910. What is the purpose of a requirements.txt file in Python?

A. It lists the hardware requirements for the program
B. It lists the required test cases
C. It specifies the Python package dependencies and their versions needed for the project
D. It contains the project documentation

Answer: C

Q911. What is the difference between a bundler and a package manager?

A. A package manager only works with frontend code
B. A package manager installs and manages dependencies while a bundler combines source files into optimized bundles for deployment
C. A bundler installs packages while a package manager bundles code
D. They are identical tools with different names

Answer: B

Q912. What is dependency resolution in package management?

A. Removing unnecessary dependencies from the project
B. Converting dependencies to a different programming language
C. Resolving conflicts between developers about which packages to use
D. The process of determining compatible versions of all direct and transitive dependencies

Answer: D

Q913. What is the purpose of namespace imports in programming?

A. Importing only private functions from a module
B. Creating namespace aliases for the operating system
C. Importing all exports from a module under a single namespace to avoid naming conflicts
D. Creating new names for existing functions

Answer: C

Q914. What is the purpose of the peerDependencies field in package.json?

A. It lists peer-reviewed packages only
B. It defines packages used exclusively during peer programming
C. It specifies packages that should be provided by the consuming project rather than installed as nested dependencies
D. It lists packages that are optional for the project

Answer: C

Q915. What is lazy loading of modules?

A. Deferring the loading of a module until it is actually needed, reducing initial startup time
B. Loading modules slowly to reduce CPU usage
C. Loading only the documentation of a module
D. Loading modules without importing them

Answer: A

Q916. What is the difference between static imports and dynamic imports?

A. Dynamic imports can only load built-in modules
B. Static imports are resolved at compile time while dynamic imports load modules at runtime based on conditions
C. Static imports are faster than dynamic imports in all cases
D. Static imports happen at runtime while dynamic imports happen at compile time

Answer: B

Q917. What is the purpose of a module's public API?

A. The internal implementation details of the module
B. The set of exported functions, classes, and types that external code can use
C. The private methods used for testing
D. The documentation comments in the module

Answer: B

Q918. What is the npm audit command used for?

A. Scanning project dependencies for known security vulnerabilities
B. Checking the financial cost of dependencies
C. Auditing the code for style violations
D. Auditing the performance of installed packages

Answer: A

Q919. What is the difference between a workspace and a standalone package?

A. Standalone packages cannot have dependencies
B. Workspaces can only contain two packages
C. A workspace manages multiple related packages in a single repository with shared dependencies while standalone packages are independent
D. Workspaces are for development only while packages are for production

Answer: C

Q920. What is the Builder pattern and when should it be used?

A. A pattern for building database tables
B. A pattern that compiles code faster
C. A pattern for building user interfaces
D. A pattern that separates object construction from representation, useful for objects with many optional parameters

Answer: D

Q921. What is the Proxy pattern?

A. A pattern that provides a surrogate object that controls access to another object
B. A pattern that hides all methods of an object
C. A pattern for configuring network proxies
D. A pattern that copies objects between servers

Answer: A

Q922. What is memoization in the context of dynamic programming?

A. Caching results of subproblems to avoid redundant computation in recursive algorithms
B. Storing algorithm pseudocode in comments
C. Writing notes about algorithm design
D. Memorizing the source code for the exam

Answer: A

Q923. What is the Chain of Responsibility pattern?

A. A pattern where a request is passed along a chain of handlers until one handles it
B. A pattern for linking database tables
C. A pattern for managing responsibility among team members
D. A pattern that chains function calls together

Answer: A

Q924. What is the difference between space complexity and time complexity?

A. They measure the same thing using different units
B. Space complexity is always larger than time complexity
C. Space complexity measures memory usage growth while time complexity measures execution time growth relative to input size
D. Time complexity only applies to recursive algorithms

Answer: C

Q925. What is the Template Method pattern?

A. A pattern for generating boilerplate code
B. A pattern that defines the skeleton of an algorithm in a base class, letting subclasses override specific steps
C. A pattern that templates all method signatures
D. A pattern for creating HTML templates

Answer: B

Q926. What is the difference between top-down and bottom-up dynamic programming?

A. They always produce different results
B. Top-down is always faster than bottom-up
C. Top-down uses recursion with memoization while bottom-up builds solutions iteratively from smallest subproblems
D. Bottom-up uses recursion while top-down uses iteration

Answer: C

Q927. What is the Mediator pattern?

A. A pattern that converts between data formats
B. A pattern for mediating between the CPU and memory
C. A pattern for middle-tier server components
D. A pattern that defines an object to encapsulate how a set of objects interact, reducing direct dependencies

Answer: D

Q928. What is amortized analysis?

A. A technique that averages the time per operation over a sequence of operations to give a tighter bound than worst-case
B. Analyzing the amount of memory amortized over program lifetime
C. Financial analysis of algorithm development costs
D. Estimating the remaining time for a running algorithm

Answer: A

Q929. What is the difference between a monitor and a semaphore?

A. A monitor provides mutual exclusion with condition variables as a high-level construct while a semaphore is a lower-level counting mechanism
B. Monitors are faster than semaphores
C. They are identical synchronization mechanisms
D. Semaphores provide mutual exclusion while monitors do not

Answer: A

Q930. What is the purpose of the volatile keyword in concurrent programming?

A. It ensures that reads and writes to a variable are visible to all threads by preventing CPU caching of the value
B. It makes a variable faster to access
C. It allocates the variable on the heap instead of the stack
D. It prevents a variable from being modified

Answer: A

Q931. What is a barrier in concurrent programming?

A. A hardware barrier between CPU cores
B. A security barrier preventing unauthorized thread access
C. A synchronization point where all threads must arrive before any can proceed past it
D. A barrier preventing memory leaks

Answer: C

Q932. What is a livelock and how does it differ from a deadlock?

A. A livelock is a deadlock that eventually resolves itself
B. In a livelock, threads actively respond to each other but make no progress, unlike deadlock where threads are blocked indefinitely
C. A deadlock involves active threads while a livelock involves blocked threads
D. A livelock uses more memory than a deadlock

Answer: B

Q933. What is the purpose of a ReadWriteLock?

A. It locks reading while allowing concurrent writes
B. It provides separate locks for different data types
C. It allows multiple threads to read simultaneously but requires exclusive access for writing
D. It separates read and write operations into different files

Answer: C

Q934. What is the difference between a Future and a Promise?

A. A Future is a read-only handle to an asynchronous result while a Promise is the writable side that resolves the value
B. They are identical concepts in all languages
C. Futures are faster than Promises
D. Promises are used only in JavaScript while Futures are used everywhere

Answer: A

Q935. What is the difference between cooperative and preemptive scheduling of threads?

A. They produce identical scheduling outcomes
B. Cooperative scheduling requires threads to voluntarily yield control while preemptive scheduling allows the OS to interrupt threads at any time
C. Preemptive scheduling cannot handle more than four threads
D. Cooperative scheduling is always more efficient

Answer: B

Q936. What is the purpose of a CountDownLatch?

A. It counts down to thread termination
B. It allows one or more threads to wait until a set of operations in other threads completes
C. It limits the number of threads that can run simultaneously
D. It counts the number of threads in the system

Answer: B

Q937. What is the actor model of concurrency?

A. A model where one main actor directs all other threads
B. A model where independent actors communicate only through asynchronous message passing, avoiding shared state
C. A model where actors perform computations in a theater-like environment
D. A model that uses actors to represent database transactions

Answer: B

Q938. What is thread-local storage?

A. A local database connection for each thread
B. Storage on the thread's stack only
C. Storage that is local to the computer running the thread
D. Memory that is private to each thread, where each thread has its own independent copy of the variable

Answer: D

Q939. What is the difference between a stub and a mock in testing?

A. A stub provides predetermined responses to method calls while a mock also verifies that expected interactions occurred
B. They are identical testing concepts
C. Stubs are faster than mocks
D. Mocks provide data while stubs verify behavior

Answer: A

Q940. What is boundary value analysis in testing?

A. Analyzing the boundaries of the codebase
B. Analyzing code coverage boundaries
C. Setting boundaries on test execution time
D. Testing at the edges of input ranges where errors are most likely to occur

Answer: D

Q941. What is the purpose of test doubles?

A. Double-checking test results manually
B. Creating two versions of each test
C. Running each test twice for verification
D. Objects that stand in for real dependencies during testing, providing isolation and control

Answer: D

Q942. What is the arrange-act-assert (AAA) pattern in testing?

A. A pattern for asserting code style compliance
B. A pattern for arranging test files alphabetically
C. A pattern for testing in three environments
D. A pattern that structures tests into setup (arrange), execution (act), and verification (assert) phases

Answer: D

Q943. What is snapshot testing?

A. Testing at a specific snapshot in version control
B. Capturing the output of a component and comparing it against a stored reference snapshot in future test runs
C. Testing memory snapshots for leaks
D. Taking screenshots of the application for visual comparison

Answer: B

Q944. What is the purpose of a code coverage tool?

A. Measuring how much of the source code is executed by the test suite
B. Covering up bugs so they are not visible
C. Covering code with comments for documentation
D. Protecting code from unauthorized access

Answer: A

Q945. What is the difference between end-to-end testing and integration testing?

A. They test the same scope at different speeds
B. Integration tests cover more code than end-to-end tests
C. End-to-end tests verify the entire application workflow from user perspective while integration tests verify interactions between specific components
D. End-to-end tests are faster than integration tests

Answer: C

Q946. What is the purpose of conditional breakpoints in a debugger?

A. Breaking execution conditionally based on CPU temperature
B. Breaking execution only when a programmer-specified condition is true, avoiding stopping at every iteration
C. Creating breakpoints that expire after a certain number of hits
D. Setting breakpoints that only work on certain days

Answer: B

Q947. What is test parameterization?

A. Setting global parameters for all tests
B. Adjusting test framework parameters
C. Parameterizing the test environment
D. Running the same test logic with different input values and expected results

Answer: D

Q948. What is the difference between a smoke test and a sanity test?

A. Sanity tests run before smoke tests in the pipeline
B. They are identical types of tests
C. Smoke tests are longer than sanity tests
D. Smoke tests verify that critical functionality works after a build while sanity tests verify that specific bug fixes or features work correctly

Answer: D

Q949. What is the difference between continuous integration and continuous delivery?

A. They are identical processes with different names
B. CI works with all languages while CD only works with web applications
C. CI deploys code while CD tests it
D. CI automatically builds and tests code changes while CD ensures code is always in a deployable state and can be released at any time

Answer: D

Q950. What is the purpose of a code linter?

A. Compiling code faster than the standard compiler
B. Linking code files together into an executable
C. Removing lint from the physical computer
D. Analyzing source code to find potential errors, style violations, and suspicious patterns before execution

Answer: D

Q951. What is the difference between a hotfix and a regular release?

A. A hotfix is an urgent fix for a critical production issue deployed outside the normal release cycle while a regular release follows the planned schedule
B. Hotfixes are larger than regular releases
C. Regular releases only contain bug fixes while hotfixes contain features
D. Hotfixes do not require testing

Answer: A

Q952. What is the purpose of semantic versioning (SemVer)?

A. Numbering each line of source code
B. Versioning based on release dates
C. Communicating the nature of changes through version numbers (MAJOR.MINOR.PATCH) so users know compatibility impact
D. Assigning version numbers based on team vote

Answer: C

Q953. What is the purpose of a CI/CD pipeline?

A. A tool for managing database migrations
B. A physical pipeline for connecting servers
C. A network protocol for transferring code
D. An automated workflow that builds, tests, and deploys code changes through defined stages

Answer: D

Q954. What is the difference between blue-green deployment and canary deployment?

A. Blue-green is faster than canary
B. They deploy code in the same way but at different times
C. Blue-green switches all traffic between two identical environments while canary gradually routes a percentage of traffic to the new version
D. Canary deployment requires two complete environments while blue-green uses one

Answer: C

Q955. What is the purpose of a code formatter like Prettier or Black?

A. Making code execute faster
B. Converting code between programming languages
C. Formatting code for printing on paper
D. Automatically formatting code to follow consistent style rules, eliminating style debates in code reviews

Answer: D

Q956. What is the purpose of a pre-commit hook?

A. Creating a backup before each commit
B. Sending a notification before each commit
C. Running automated checks (like linting, tests, or formatting) before a git commit is created, preventing bad code from entering the repository
D. Committing code before anyone else

Answer: C

Q957. What is the difference between a monolith and microservices architecture?

A. Monoliths are always slower than microservices
B. A monolith deploys the entire application as a single unit while microservices decompose it into independently deployable services
C. Monoliths do not support any form of modularity
D. Microservices cannot share databases

Answer: B

Q958. What is the difference between eager evaluation and lazy evaluation of function arguments?

A. Lazy evaluation computes all arguments immediately
B. Eager evaluation is always slower than lazy evaluation
C. Eager evaluation only works with pure functions
D. Eager evaluation computes arguments before passing them while lazy evaluation delays computation until the value is actually needed

Answer: D

Q959. What is the difference between stable and unstable sorting algorithms?

A. Stable sorts only work on numeric data
B. Unstable sorts produce incorrect results for duplicate values
C. Stable sorts are always faster than unstable sorts
D. Stable sorts preserve the relative order of equal elements while unstable sorts may rearrange them

Answer: D

Q960. What is the purpose of a definition of done (DoD) in agile development?

A. A list of developers assigned to each task
B. Defining when the project is completely finished
C. A shared checklist of criteria that must be met for a work item to be considered complete
D. A document that defines the final release date

Answer: C

Hard Questions

480 questions

Q961. What is the halting problem in computer science?

A. Optimizing a program execution speed through advanced compiler techniques
B. Determining whether an arbitrary program will finish running or loop forever
C. Finding and reporting memory leaks occurring during a program execution
D. Determining whether a program contains syntax errors in its source code

Answer: B

Q962. In language theory, what is a context-free grammar (CFG)?

A. A grammar where production rules apply regardless of context
B. A grammar equivalent to standard regular expressions
C. A grammar where rules depend on surrounding token context
D. A grammar that is exclusively used in natural languages

Answer: A

Q963. What is tail call optimization (TCO)?

A. Optimizing the first function call in a chain
B. Parallelizing multiple function calls at once
C. Reusing the current stack frame for a tail call
D. Removing unused function calls from the code

Answer: C

Q964. What distinguishes a pure function in functional programming?

A. It returns the same output for the same input always
B. It cannot accept any parameters from the caller
C. It modifies global state and external variables freely
D. It must use recursion as its primary control flow

Answer: A

Q965. What is referential transparency?

A. A function that returns void and produces no output value
B. An expression replaceable with its value without side effects
C. A pointer that references itself in a circular data structure
D. A variable that can be referenced from anywhere in the code

Answer: B

Q966. What is the difference between static and dynamic dispatch?

A. Static dispatch is always slower than dynamic dispatch in practice
B. Static dispatch resolves methods at compile time; dynamic at runtime
C. They are identical concepts in object-oriented programming languages
D. Dynamic dispatch is only available in the C programming language

Answer: B

Q967. What is a closure in programming?

A. A method used to close open file handles
B. A function bundled with its lexical scope
C. A technique to terminate a running loop
D. A destructor that cleans up class state

Answer: B

Q968. What is continuation-passing style (CPS)?

A. A structured method for handling runtime errors
B. Passing control explicitly via continuation functions
C. A design pattern for building user interfaces
D. Passing all arguments by reference to functions

Answer: B

Q969. What is homoiconicity in a programming language?

A. All variables in the program share the same type
B. The code can run on any platform without changes
C. Code structure uses the language own data structures
D. Functions return the exact same type they receive

Answer: C

Q970. What is the Curry-Howard correspondence?

A. A technique for normalizing database tables
B. A method for performing function currying
C. A link between mathematical proofs and programs
D. A protocol for networking between computers

Answer: C

Q971. What is the IEEE 754 standard?

A. A specification for database schemas
B. A standard for security encryption
C. A protocol for computer networking
D. A standard for floating-point numbers

Answer: D

Q972. What is a phantom type?

A. A type providing compile-time safety without storing data
B. A type that has been deprecated from the language standard
C. A type reserved for representing ghost or phantom objects
D. A type that only exists at runtime and not at compile time

Answer: A

Q973. What is the difference between structural typing and nominal typing?

A. They are exactly the same concept with different names
B. Structural typing is only applicable to struct data types
C. Nominal typing is only applicable to numeric data types
D. Structural checks compatibility by shape; nominal by name

Answer: D

Q974. What is type erasure in Java generics?

A. Removing type information at the compile stage
B. Deleting all variables of a particular given type
C. Generic type info removed at runtime as raw types
D. Converting every type to Object at compile time

Answer: C

Q975. What is a dependent type?

A. A type that inherits from another
B. A type that depends on a library
C. A type that is defined by a value
D. A type that requires garbage collection

Answer: C

Q976. What is the bottom type in type theory?

A. A type with no values and subtype of every type
B. The null type representing an absence of a value
C. The base class that all other types inherit from
D. The standard integer type used for whole numbers

Answer: A

Q977. What is variance in type systems?

A. Random and unpredictable behavior in type inference
B. The size variation of different data types in memory
C. How subtyping of complex types relates to components
D. The total range of values a type can hold

Answer: C

Q978. What is an algebraic data type (ADT)?

A. An array type that exclusively stores numeric values
B. A data type designed for mathematical operations only
C. A type used for representing algebraic math expressions
D. A composite type combining sum and product operations

Answer: D

Q979. What is the unit type?

A. A type with exactly one possible value
B. A type used for measurement units
C. The integer constant value of one
D. A class with a single instance only

Answer: A

Q980. What is the purpose of type witnesses in advanced type systems?

A. To provide explicit proof of type relationships
B. To help debug complex type errors in code
C. To convert types dynamically during execution
D. To observe and log type changes at runtime

Answer: A

Q981. What is operator overloading?

A. Defining custom behavior for user-defined types
B. An error caused by incorrect operator usage
C. Increasing the default operator precedence level
D. Using too many operators in one expression

Answer: A

Q982. What is the result of ~5 in a system using 32-bit signed integers?

A. 5
B. -5
C. 6
D. -6

Answer: D

Q983. What is the spaceship operator (<=>)?

A. An assignment operator for setting variables
B. A comparison operator returning -1, 0, or 1
C. A bitwise shift operator for binary values
D. A logical operator for boolean expressions

Answer: B

Q984. What is the difference between arithmetic right shift (>>) and logical right shift (>>>)?

A. There is no difference between them at all
B. >>> preserves the sign bit; >> fills with zeros
C. >> preserves the sign bit; >>> fills with zeros
D. >> always fills the vacant bit positions with zeros

Answer: C

Q985. What is the Elvis operator (?:) in Kotlin?

A. Returns left if non-null, otherwise right
B. A type-checking operator for null safety
C. A standard ternary conditional operator
D. A bitwise operator for binary operations

Answer: A

Q986. How does Python's 'is' operator differ from '=='?

A. is runs faster but produces less accurate results
B. is checks identity; == checks value equality
C. They are functionally identical in all cases
D. == checks identity while is checks equality

Answer: B

Q987. What is the safe navigation operator (?.) in Kotlin/TypeScript?

A. It forces a null pointer exception to be thrown immediately
B. It converts any null value to the integer zero automatically
C. It always returns a null value regardless of the receiver
D. It accesses a member only if the receiver is non-null

Answer: D

Q988. What does the >>> operator do in Java that >> does not?

A. It fills leftmost bits with 0, ignoring the sign
B. It operates more slowly than the >> operator
C. It works on floating-point numbers unlike >>
D. Nothing different from the >> operator at all

Answer: A

Q989. What is destructuring assignment?

A. A type of memory deallocation for objects
B. Deleting variables from the current scope
C. Unpacking values from arrays or objects into variables
D. Assigning null to multiple variables at once

Answer: C

Q990. What is the pipeline operator (|>) proposed for JavaScript?

A. A bitwise OR operation on binary values
B. A comparison operator for ordering two values
C. It passes left result as argument to right function
D. A type conversion operator between data types

Answer: C

Q991. What is tail recursion and how does it relate to loops?

A. Recursion that uses a tail pointer data structure
B. Recursion without any base case for termination
C. A loop structure that invokes itself recursively
D. A tail-position recursive call optimized into a loop

Answer: D

Q992. What is the difference between structured and unstructured control flow?

A. Structured control flow is slower than unstructured control
B. Unstructured control flow is more readable than structured
C. There is no meaningful difference between the two approaches
D. Structured uses blocks like if/while; unstructured uses goto

Answer: D

Q993. What is pattern matching as a control structure?

A. Matching strings with regular expressions
B. A generalized switch that destructures data types
C. A repeating pattern within a loop construct
D. A text search algorithm for finding substrings

Answer: B

Q994. What is the difference between eager and lazy evaluation in control flow?

A. Eager evaluates expressions immediately; lazy defers until value needed
B. Lazy evaluation is always faster than eager evaluation in all cases
C. Eager evaluation is only used in purely functional language paradigms
D. There is no practical difference between the two evaluation strategies

Answer: A

Q995. What is a coroutine and how does it differ from a subroutine?

A. A coroutine is a specialized type of loop iteration control structure
B. A subroutine can be paused and resumed at arbitrary execution points
C. They are the same thing with different names in different languages
D. A coroutine can suspend and resume; a subroutine runs to completion

Answer: D

Q996. What is the purpose of guard clauses?

A. To lock and protect shared resources from concurrent thread access
B. To guard against invalid memory allocation during program startup
C. To protect systems against unauthorized network access attempts
D. To handle edge cases early by returning at the start of functions

Answer: D

Q997. What is a trampoline in the context of recursion?

A. A technique converting recursion into iterative loops
B. A specialized debugging tool for stack overflow
C. A physical device used for hardware testing
D. A tree-based data structure for fast lookup

Answer: A

Q998. What is continuation in control flow?

A. A loop continuation condition for iterations
B. An alternative to the break statement in loops
C. The next sequential statement in the code
D. An abstract representation of remaining computation

Answer: D

Q999. What is the difference between deterministic and non-deterministic control flow?

A. There is no practical difference between them at all
B. Deterministic follows a fixed path; non-deterministic explores many
C. Non-deterministic control flow always causes the program to crash
D. Deterministic control flow always executes more slowly than others

Answer: B

Q1000. What is the 'goto considered harmful' argument?

A. Goto should be used everywhere as the primary control flow method
B. Goto causes memory leaks and resource allocation failures in code
C. Dijkstra argued goto makes programs hard to follow and reason about
D. Goto is always the most efficient control flow mechanism available

Answer: C

Q1001. What is function currying?

A. Overloading a function with additional parameter types
B. Caching the results of previous function calls
C. Optimizing a function for better performance
D. Transforming multi-argument function into single-argument chain

Answer: D

Q1002. What is memoization?

A. Writing descriptive comments inside function body code
B. Caching expensive function results to avoid recomputation
C. Saving a function definition to persistent disk storage
D. Allocating additional memory for function local variables

Answer: B

Q1003. What is the difference between early binding and late binding?

A. Early binding resolves at compile time; late binding at runtime
B. There is no meaningful difference between the two approaches
C. Early binding is always slower than late binding in all cases
D. Late binding only works in the C programming language alone

Answer: A

Q1004. What is a variadic function?

A. A function that has a variable return data type
B. A function that returns multiple values at once
C. A function that accepts no input parameters
D. A function accepting variable number of arguments

Answer: D

Q1005. What is partial application?

A. Fixing some arguments to produce a new function taking the rest
B. Calling a function with completely wrong argument types or values
C. A function that only partially works and fails on some inputs
D. An incomplete function definition that is missing its body code

Answer: A

Q1006. What is a thunk?

A. A runtime error in the application
B. A wrapper delaying computation evaluation
C. A pre-allocated block of heap memory
D. A particular type of variable binding

Answer: B

Q1007. What are first-class functions?

A. Functions assignable to variables and passable as arguments
B. Functions that have the highest execution priority level
C. Functions that belong to the main class of the program
D. Functions that are defined first in source code order

Answer: A

Q1008. What is the difference between a method and a function?

A. Methods cannot return any values while functions always can
B. Functions cannot accept any parameters while methods always can
C. They are always the same thing in every programming language
D. A method belongs to an object or class; a function is standalone

Answer: D

Q1009. What is a generator function?

A. A function that creates and returns other functions
B. A function that yields multiple values over time lazily
C. A function that generates random numbers on demand
D. A constructor that initializes a new class instance

Answer: B

Q1010. What is the Y combinator?

A. A fixed-point combinator enabling anonymous recursion
B. A popular comparison-based sorting algorithm
C. A well-known startup incubator company
D. A commonly used structural design pattern

Answer: A

Q1011. What is memory-mapped I/O?

A. Mapping files into virtual memory address space
B. Encrypting all I/O data during transfer
C. Storing I/O data in the processor cache
D. I/O that uses only RAM for data storage

Answer: A

Q1012. What is the difference between blocking and non-blocking I/O?

A. Non-blocking I/O is always slower than blocking I/O overall
B. There is no meaningful difference between the two modes
C. Blocking I/O never waits for the operation to finish first
D. Blocking waits until completion; non-blocking returns immediately

Answer: D

Q1013. What is asynchronous I/O?

A. I/O that continues without waiting, notified on completion
B. I/O that is identical to synchronous blocking I/O
C. I/O operations that execute in strict sequential order
D. I/O operations that are always extremely fast

Answer: A

Q1014. What is the purpose of the select() system call?

A. Selecting a record from a database table query
B. Selecting files to delete from the file system
C. Monitoring file descriptors for I/O readiness
D. Choosing which thread the scheduler should run

Answer: C

Q1015. What is zero-copy I/O?

A. I/O operations that have zero financial cost
B. I/O operations that transfer no data at all
C. Deleting data as it is being transferred out
D. Transferring data without copying between buffers

Answer: D

Q1016. What is the reactor pattern in I/O?

A. A nuclear-physics computing simulation pattern
B. An event-driven dispatcher that demultiplexes I/O events
C. A chemical reaction simulation programming model
D. A simple loop-based sequential processing pattern

Answer: B

Q1017. What are I/O multiplexing mechanisms like epoll and kqueue?

A. Command-line utilities designed for copying files efficiently
B. Multi-threaded I/O libraries for parallel data processing
C. Mechanisms that duplicate I/O devices for redundancy
D. Scalable kernel interfaces for monitoring many file descriptors

Answer: D

Q1018. What is the difference between character devices and block devices?

A. Character devices transfer byte by byte; block in fixed chunks
B. Block devices are exclusively used for text file operations
C. There is no functional difference between device types
D. Character devices are always faster than block-based devices

Answer: A

Q1019. What is Direct Memory Access (DMA) in the context of I/O?

A. Hardware allowing device-to-memory transfer without CPU
B. A software programming technique for data transfer
C. A debugging method for diagnosing hardware I/O issues
D. A specialized type of dynamic memory allocation method

Answer: A

Q1020. What is scatter-gather I/O (vectored I/O)?

A. Reading/writing multiple non-contiguous buffers at once
B. Distributing I/O operations across network nodes
C. Random I/O patterns across scattered disk sectors
D. Fragmented file storage across multiple disk drives

Answer: A

Q1021. What is the time complexity of naive string matching?

A. O(n*m)
B. O(log n)
C. O(n)
D. O(n^2)

Answer: A

Q1022. What is the KMP (Knuth-Morris-Pratt) string matching algorithm?

A. A comparison-based sorting algorithm for arrays
B. A tree traversal algorithm for binary search trees
C. A linear-time string matcher using a failure function
D. A hashing algorithm for generating unique hash codes

Answer: C

Q1023. What is string interning?

A. Compressing strings to reduce their memory size
B. Storing one copy of each distinct string value
C. Encrypting strings for secure data transmission
D. Storing strings in a temporary memory buffer

Answer: B

Q1024. What is the Rabin-Karp algorithm?

A. A graph traversal algorithm for shortest paths
B. A string matcher using hashing for efficient search
C. A comparison-based sorting algorithm for collections
D. A compression algorithm for reducing data file sizes

Answer: B

Q1025. What is a trie (prefix tree) and how does it relate to strings?

A. A linked list that stores strings in sequential memory order
B. A standard binary search tree data structure
C. A hash table that maps string keys to their computed values
D. A tree where each node represents a character for prefix search

Answer: D

Q1026. What is the difference between UTF-8, UTF-16, and UTF-32?

A. UTF-32 uses the least amount of storage space per character overall
B. They are all identical encoding formats with no differences at all
C. UTF-8 uses 1-4 bytes, UTF-16 uses 2-4 bytes, UTF-32 uses exactly 4
D. UTF-8 is a fixed-width encoding that always uses exactly one byte

Answer: C

Q1027. What is a rope data structure for strings?

A. A contiguous array of individual character values
B. A hash map that stores character frequency counts
C. A singly linked list of character nodes
D. A balanced tree for efficient long string operations

Answer: D

Q1028. What is the Levenshtein distance?

A. The total length of a string in characters
B. The count of matching characters between strings
C. The byte size of a string in memory allocation
D. The minimum single-character edits between strings

Answer: D

Q1029. What is the Boyer-Moore string search algorithm's key insight?

A. Start comparing from the beginning of the pattern each time
B. Use hashing to compare the pattern against every position
C. Compare every single character without skipping any at all
D. Start from pattern end and use heuristics to skip sections

Answer: D

Q1030. What are surrogate pairs in UTF-16?

A. Duplicate characters stored in the encoding table
B. Error codes returned by the UTF-16 encoding process
C. Pairs of 16-bit units for characters beyond the BMP
D. Paired delimiter characters used in string literals

Answer: C

Q1031. What is a bloom filter?

A. A comparison-based sorting algorithm for arrays
B. A space-efficient probabilistic set membership tester
C. An image processing filter for visual enhancement
D. A standard hash table with separate chaining buckets

Answer: B

Q1032. What is a skip list?

A. A filtered array that excludes elements matching a condition
B. A probabilistic multi-layer linked list for O(log n) search
C. A list that skips over certain elements during iteration
D. A type of priority queue that maintains sorted element order

Answer: B

Q1033. What is a concurrent hash map?

A. A hash map using multiple hash functions per key
B. A thread-safe hash map without full map locking
C. A distributed database across network nodes only
D. A regular hash map with no special properties

Answer: B

Q1034. What is the amortized time complexity of adding an element to a dynamic array?

A. O(1) amortized
B. Always O(n)
C. O(log n) each
D. O(n log n) each

Answer: A

Q1035. What is a persistent data structure?

A. An immutable structure that can never be accessed again
B. A structure preserving previous versions on modification
C. A relational database schema structure for queries
D. A structure saved to persistent disk storage

Answer: B

Q1036. What is a B-tree and where is it commonly used?

A. A self-balancing tree with multiple keys per node for databases
B. A hash table variant that uses tree-based collision resolution
C. A standard binary tree with two children per node
D. A linked list type that organizes data in sequential node chains

Answer: A

Q1037. What is consistent hashing?

A. A type of encryption used for securing sensitive data values
B. A collision resolution method for hash table bucket overflow
C. A hashing scheme that minimally redistributes keys on change
D. Regular hashing with standard hash function methods

Answer: C

Q1038. What is a lock-free data structure?

A. An unprotected array that any thread can freely modify
B. A concurrent structure using atomic operations not locks
C. A structure with no synchronization protection at all
D. A read-only data structure that cannot be changed ever

Answer: B

Q1039. What is the difference between external and internal iterators?

A. Internal iterators always consume more memory than external iterators
B. External iterators are always faster than internal iterators in practice
C. There is no meaningful difference between the two iterator types
D. External iterators are client-controlled; internal are collection-controlled

Answer: D

Q1040. What is a disjoint set (union-find) data structure?

A. A structure tracking element partitions with union and find
B. A sorted set that maintains elements in order
C. A hash set variant that uses separate chaining for entries
D. A set that has no intersections with any other set at all

Answer: A

Q1041. What is the SOLID principle in OOP?

A. A coding style guide for naming conventions in code
B. A data structure for organizing elements efficiently
C. Five design principles: SRP, OCP, LSP, ISP, and DIP
D. A testing framework for validating software behavior

Answer: C

Q1042. What is a virtual method table (vtable)?

A. A hash table for caching method results quickly
B. A debug table for tracking method execution and logging
C. A lookup table for resolving dynamic method calls at runtime
D. A method list defined in the source code of the program

Answer: C

Q1043. What is the difference between association, aggregation, and composition?

A. Association is general; aggregation weak; composition strong ownership
B. They are all identical concepts with just different names used
C. Composition is always the better design choice in every single case
D. Only naming differences exist with no behavioral distinctions at all

Answer: A

Q1044. What is a mixin?

A. A class that mixes different data types together randomly
B. A class providing methods to others without being standalone
C. A type of interface that defines a strict method contract
D. A structural design pattern for building complex objects

Answer: B

Q1045. What is double dispatch and how does it differ from single dispatch?

A. Single dispatch uses two methods while double dispatch uses only one
B. There is no practical difference between the two dispatch mechanisms
C. Single dispatch selects by one type; double dispatch by two types
D. Double dispatch is twice as fast in execution as single dispatch

Answer: C

Q1046. What is metaclass programming?

A. Creating graphical user interfaces programmatically
B. Writing classes whose instances are themselves classes
C. A commonly used structural design pattern for code
D. Programming about programs in general terms

Answer: B

Q1047. What is the difference between prototype-based and class-based OOP?

A. Prototype-based OOP is always slower than class-based OOP overall
B. Class-based uses blueprints; prototype-based clones existing objects
C. They are the same approach with no meaningful differences at all
D. Class-based OOP cannot support any form of inheritance at all

Answer: B

Q1048. What is the fragile base class problem?

A. A compilation error from missing method implementations
B. A memory leak caused by improper object disposal code
C. A class that is too small to be useful on its own
D. Base class changes can break derived classes unexpectedly

Answer: D

Q1049. What is covariant return type?

A. A generic type parameter used in collection class definitions
B. Allowing an overriding method to return a more specific subtype
C. Always returning the exact same type as the parent declares
D. A type error that occurs when return types do not match up

Answer: B

Q1050. What is the object slicing problem in C++?

A. Splitting an object into smaller component parts for storage
B. When assigning derived to base variable, derived data is lost
C. A memory allocation issue caused by fragmented heap storage
D. A threading problem from concurrent access to shared objects

Answer: B

Q1051. What is the mark-and-sweep garbage collection algorithm?

A. Marking files for deletion from the disk system
B. A disk cleanup algorithm for removing temporary files
C. A memory allocation method for new object creation
D. A GC that marks reachable objects then frees unmarked

Answer: D

Q1052. What is a generational garbage collector?

A. A manual memory manager without automatic collection
B. A garbage collector for old legacy programs only
C. A GC dividing objects by age, collecting young often
D. A reference counting garbage collection implementation

Answer: C

Q1053. What is the difference between unique_ptr, shared_ptr, and weak_ptr in C++?

A. unique_ptr is exclusive; shared_ptr shares; weak_ptr observes only
B. They are all the same smart pointer type with no differences
C. shared_ptr cannot be copied or assigned to another shared_ptr var
D. unique_ptr is always slower than the other two smart pointer types

Answer: A

Q1054. What is memory-mapped files and how do they work?

A. Files mapped into virtual address space for memory I/O
B. Files that have been encrypted for secure data storage
C. Files that are stored only in volatile RAM memory
D. Files that have been compressed to save disk space used

Answer: A

Q1055. What is a slab allocator?

A. A garbage collector that runs periodically in background
B. A simple wrapper around the standard malloc function call
C. A disk allocator that manages file system block assignments
D. A mechanism pre-allocating and caching object-sized blocks

Answer: D

Q1056. What is escape analysis in JVM?

A. Compiler analysis determining if object references escape scope
B. Network traffic analysis for monitoring data flow patterns
C. Analyzing program exit codes and termination conditions
D. Exception analysis for tracking error propagation in code

Answer: A

Q1057. What is the ABA problem in lock-free memory management?

A. When a location changes A to B to A, making CAS incorrectly succeed
B. A problem related to the alphabet ordering of values
C. A variant of buffer overflow caused by writing past array boundaries
D. A type of memory leak caused by circular reference chains in objects

Answer: A

Q1058. What is a write barrier in garbage collection?

A. A mechanism intercepting pointer writes for GC invariants
B. A memory protection scheme against buffer overflows
C. A disk write cache for improving I/O performance
D. A network firewall for filtering traffic

Answer: A

Q1059. What is Copy-on-Write (COW)?

A. Sharing memory until a write triggers actual copying
B. Copying files manually between directories on disk
C. A garbage collection algorithm for reclaiming memory
D. An error handling technique for runtime exceptions

Answer: A

Q1060. What is address space layout randomization (ASLR)?

A. Randomizing process memory addresses to prevent attacks
B. A memory allocation strategy for dynamic array growth
C. An addressing mode for accessing hardware registers
D. A sorting algorithm for ordering memory addresses

Answer: A

Q1061. What is the difference between fault tolerance and exception handling?

A. Exception handling prevents all errors from ever occurring in the program
B. Exception handling manages code errors; fault tolerance is system resilience
C. They are exactly the same concept with no practical differences at all
D. Fault tolerance is always faster and easier to implement than exceptions

Answer: B

Q1062. What is an exception safety guarantee?

A. A guarantee that no exceptions will ever occur in the program
B. Levels of state guarantees after exception: no-throw, strong, basic
C. A compiler feature that eliminates all runtime exceptions automatically
D. A testing tool that verifies exception handlers work correctly always

Answer: B

Q1063. What is the 'catch-all' anti-pattern?

A. A testing pattern for verifying exception handler correctness
B. Catching all exceptions generically which hides bugs in code
C. Catching only specific well-defined exceptions in the code
D. A recommended design pattern for robust exception handling

Answer: B

Q1064. What is the difference between recoverable and unrecoverable errors?

A. There is no meaningful difference between the two error categories
B. All errors in every program are always recoverable with proper code
C. All errors in every program are always unrecoverable without fixes
D. Recoverable errors can be handled; unrecoverable require termination

Answer: D

Q1065. What is Result/Either type as an alternative to exceptions?

A. A logging mechanism for recording exception messages
B. A type representing either success value or error value
C. A direct replacement for try-catch in all situations
D. A simple boolean type with true or false values

Answer: B

Q1066. What is the problem with exceptions and concurrency?

A. Exceptions in one thread may not be visible in another thread
B. No problems exist between exceptions and concurrency at all
C. Concurrency completely prevents all exceptions from occurring
D. Exceptions are inherently thread-safe in all implementations

Answer: A

Q1067. What is a panic in Rust/Go?

A. A regular exception like any other in the language
B. A warning message that can be safely ignored by developers
C. An unrecoverable error that unwinds the stack and terminates
D. A debug log entry recorded during program execution review

Answer: C

Q1068. What is structured concurrency's approach to exception handling?

A. Ignoring all exceptions thrown by child tasks completely
B. Propagating child exceptions to parent and cancelling siblings
C. Logging exceptions only without any propagation to parents
D. Catching all exceptions silently without any further action

Answer: B

Q1069. What is the difference between resumable and non-resumable exceptions?

A. Resumable allows continuing at throw point; non-resumable unwinds
B. There is no meaningful difference between the two exception types
C. Non-resumable exceptions can be resumed after the handler finishes
D. Resumable exceptions are always slower than non-resumable exceptions

Answer: A

Q1070. What is the 'exception specification' problem in C++?

A. Exception specifications are completely reliable and always enforced
B. They always improve performance of exception handling in all cases
C. They are required by the C++ standard for every function definition
D. C++ throw() specs were problematic and replaced by noexcept keyword

Answer: D

Q1071. What is dependency injection?

A. Installing package dependencies from the package registry
B. A package manager for downloading and installing third-party libs
C. Providing dependencies from outside rather than creating internally
D. An import statement for including external module code in file

Answer: C

Q1072. What is the OSGi framework?

A. A build tool for compiling and packaging Java applications
B. A dynamic Java module system supporting runtime bundle mgmt
C. An operating system for mobile devices and desktops
D. A testing framework for writing and running automated tests

Answer: B

Q1073. What is the difference between CommonJS and ES Modules?

A. They are identical module systems with no meaningful differences
B. ES Modules do not support any form of async module loading
C. CommonJS uses require synchronously; ESM uses import/export
D. CommonJS is the newer standard developed after ES Modules

Answer: C

Q1074. What is the diamond dependency problem in package management?

A. Two dependencies requiring different versions of the same package
B. A class inheritance issue in object-oriented programming code
C. A circular dependency between two modules in the same project
D. A build error caused by missing configuration files in project

Answer: A

Q1075. What is a plugin architecture?

A. A fixed codebase that cannot be extended or modified at all
B. A build system for compiling and packaging application binaries
C. A testing framework for automated integration and unit testing
D. A design allowing runtime extension via external loaded modules

Answer: D

Q1076. What is the Inversion of Control (IoC) container?

A. A framework component managing object creation and dependencies
B. A data structure for storing elements in a specific order only
C. A reversed loop that iterates in backward order through data
D. A version control system for tracking changes to source code

Answer: A

Q1077. What is ABI (Application Binary Interface) compatibility?

A. A network protocol for communication between distributed services
B. The same as API compatibility with no additional constraints
C. Binary-level interface ensuring compiled modules interact correctly
D. A data format for serializing objects to disk or network transfer

Answer: C

Q1078. What is the Service Provider Interface (SPI) pattern?

A. A web service for hosting API endpoints on the internet
B. A physical network interface for connecting hardware devices
C. A design where providers implement a service-defined interface
D. A build tool for automating compilation and deployment tasks

Answer: C

Q1079. What is reproducible builds and why is it important for packages?

A. Building projects as fast as possible to save developer time
B. Building multiple modules in parallel across available cores
C. Building the same project twice in a row for verification
D. Ensuring same source always produces bit-identical output binary

Answer: D

Q1080. What is the difference between peer dependencies and regular dependencies?

A. Regular dependencies are always shared between all project packages
B. There is no meaningful difference between the two dependency types
C. Peer dependencies require the consumer to provide a shared version
D. Peer dependencies are completely optional and never need installing

Answer: C

Q1081. What is the Visitor pattern and when would you use it?

A. Defining new operations on objects without changing their classes
B. An authentication pattern for verifying user identity securely
C. A logging pattern for recording all application events to files
D. Tracking website visitors for analytics and usage metrics

Answer: A

Q1082. What is a monad in functional programming?

A. A single immutable value in the program code
B. A pattern wrapping values for chaining operations with effects
C. A primitive data type for storing basic values in memory
D. A loop construct for iterating through collection elements

Answer: B

Q1083. What is the difference between a microkernel and a monolithic architecture?

A. Microkernel architecture is always larger in size than monolithic
B. Monolithic architecture is always more modular than microkernel
C. Microkernel has minimal core with plugins; monolithic is all-in-one
D. There is no meaningful difference between the two architecture styles

Answer: C

Q1084. What is reactive programming?

A. Procedural programming with sequential statement execution
B. A paradigm focused on async data streams and change propagation
C. Programming with user interface reactions and animations
D. Event-driven programming using callbacks and event listeners

Answer: B

Q1085. What is the Command Query Responsibility Segregation (CQRS) pattern?

A. An authentication method for verifying user credentials
B. A database query for selecting records from tables
C. A sorting algorithm for ordering elements in collections
D. Separating read and write operations into different models

Answer: D

Q1086. What is a domain-specific language (DSL)?

A. A general-purpose programming language for all domains
B. A scripting language for automating repetitive tasks
C. A specialized language designed for a particular domain
D. An assembly language for low-level hardware programming

Answer: C

Q1087. What is the actor model?

A. Independent units communicating via asynchronous message passing
B. A relational database model for organizing table schemas
C. A user interface pattern for laying out visual components
D. A design pattern for video game character development

Answer: A

Q1088. What is aspect-oriented programming (AOP)?

A. A subtype of object-oriented programming methodology
B. Separating cross-cutting concerns from business logic via aspects
C. A user interface pattern for organizing visual layout components
D. A testing approach for validating application behavior thoroughly

Answer: B

Q1089. What is the hexagonal architecture (ports and adapters)?

A. A network topology for connecting distributed system nodes
B. A six-sided geometric design for user interfaces
C. Isolating core domain from externals through ports and adapters
D. A database design for organizing tables and relationships

Answer: C

Q1090. What is algebraic effects as a programming concept?

A. Expressing side effects as values handled by effect handlers separately
B. An optimization technique for improving numerical computation speed
C. Mathematical operations performed on algebraic expressions
D. A specialized data structure for storing algebraic computation results

Answer: A

Q1091. What is the difference between optimistic and pessimistic concurrency control?

A. There is no meaningful difference between the two control approaches
B. Pessimistic concurrency control never uses any locks on resources
C. Optimistic concurrency control uses locks to prevent all conflicts
D. Optimistic assumes no conflicts and validates; pessimistic locks first

Answer: D

Q1092. What is the Compare-and-Swap (CAS) operation?

A. A file swap operation for exchanging file contents
B. A string operation for replacing character subsequences
C. A sorting operation for ordering elements in arrays
D. An atomic operation comparing and swapping memory values

Answer: D

Q1093. What is the Java Memory Model (JMM)?

A. A memory leak detector for finding unreleased allocations
B. A garbage collection model for reclaiming unused object memory
C. A specification defining thread memory interaction and visibility
D. How Java manages memory allocation for objects on the heap

Answer: C

Q1094. What is work stealing in thread pool implementations?

A. A security vulnerability where threads access unauthorized data
B. Interrupting running threads to reassign their execution priority
C. Idle threads stealing tasks from busy threads for load balancing
D. Taking another thread source code and executing it directly

Answer: C

Q1095. What is a memory barrier/fence?

A. A hardware instruction enforcing memory operation ordering for CPUs
B. A memory protection boundary between user and kernel space
C. A memory limit restricting maximum allocation size for processes
D. A garbage collection trigger that starts when memory is low

Answer: A

Q1096. What is the happens-before relationship in concurrency?

A. A scheduling algorithm for determining thread execution priority
B. Simple chronological ordering of events by their timestamps
C. A debugging concept for tracing the order of function calls
D. A guarantee that writes by one action are visible to another

Answer: D

Q1097. What is a read-write lock (ReentrantReadWriteLock)?

A. A file lock for preventing concurrent file access
B. A database lock for serializing transaction execution
C. A lock allowing multiple readers but exclusive writers
D. A mutex variant supporting recursive lock acquisition

Answer: C

Q1098. What is lock-free programming?

A. A type of race condition caused by missing synchronization
B. Programming without any synchronization mechanisms at all
C. Single-threaded programming without any concurrent access
D. Using atomic operations instead of locks for thread progress

Answer: D

Q1099. What is the difference between green threads and OS threads?

A. Green threads are runtime-scheduled; OS threads are kernel-managed
B. Green threads are always faster than OS threads in every scenario
C. They are exactly the same with no meaningful differences at all
D. OS threads use less memory than green threads in all situations

Answer: A

Q1100. What is Software Transactional Memory (STM)?

A. A concurrency mechanism using transactions for shared memory with retry
B. A memory allocation strategy for managing heap memory efficiently
C. A thread scheduler for determining execution order of all threads
D. A relational database system for persistent data storage

Answer: A

Q1101. What is property-based testing?

A. Specifying invariants that hold for all generated inputs
B. Testing object properties for correct values and types
C. Testing CSS properties for correct visual rendering
D. Testing configuration properties for application behavior

Answer: A

Q1102. What is fuzzing?

A. Making code intentionally unclear and hard to understand
B. Feeding random or malformed data to find vulnerabilities
C. A string operation for making text fuzzy and unreadable
D. A sorting technique that introduces randomness into order

Answer: B

Q1103. What is mutation testing?

A. Testing DNA sequences for mutations in bioinformatics apps
B. Introducing small code changes to check if tests catch them
C. Testing genetic algorithms for correct evolution behavior
D. Testing code changes for backward compatibility with old code

Answer: B

Q1104. What is the difference between static and dynamic analysis?

A. Dynamic analysis is performed before compilation of the source code
B. Static examines code without executing; dynamic examines during execution
C. There is no meaningful difference between the two analysis approaches
D. Static analysis is performed at runtime while the program is executing

Answer: B

Q1105. What is a flame graph in performance analysis?

A. An error graph displaying exception frequency over time
B. A fire hazard visualization for safety monitoring apps
C. A visualization of call stacks showing CPU time by code path
D. A heat map showing temperature distribution across hardware

Answer: C

Q1106. What is symbolic execution?

A. Executing code with special symbol characters as input data
B. Executing code with symbolic values to explore all possible paths
C. A debugging mode that shows variable names instead of values
D. Running code with special characters to test encoding correctness

Answer: B

Q1107. What is the difference between correctness testing and performance testing?

A. They are exactly the same type of testing with no differences
B. Correctness verifies behavior; performance measures speed and resources
C. Correctness testing measures execution speed like performance does
D. Performance testing checks for bugs just like correctness testing

Answer: B

Q1108. What is chaos engineering?

A. Random testing with no specific goals or expected outcomes
B. Unstructured development without any planning or methods
C. Creating intentionally buggy software for entertainment
D. Intentionally introducing failures to test system resilience

Answer: D

Q1109. What is taint analysis?

A. Analyzing memory allocation patterns for optimization purposes
B. Tracking untrusted input flow to find security vulnerabilities
C. Checking code for bugs using standard debugging tools
D. Formatting code according to standard style guide conventions

Answer: B

Q1110. What is the difference between benchmarking and profiling?

A. They are identical performance analysis methods with no differences
B. Benchmarking measures overall metrics; profiling identifies bottlenecks
C. Profiling gives overall performance metrics; benchmarking is specific
D. Benchmarking provides more detailed analysis than profiling overall

Answer: B

Q1111. What is the Twelve-Factor App methodology?

A. Twelve testing phases that every software project must go through
B. A math-based approach to solving programming problems algorithmically
C. Twelve best practices for building scalable cloud-native applications
D. Twelve programming languages that every developer should learn fully

Answer: C

Q1112. What is trunk-based development?

A. A database design pattern for normalizing relational table schemas
B. A branching strategy with single shared branch and short-lived branches
C. Using tree data structures for organizing hierarchical information
D. A testing strategy for validating software behavior systematically

Answer: B

Q1113. What is the difference between monorepo and polyrepo strategies?

A. Polyrepo is only suitable for small projects with limited source code
B. Monorepo is always the better choice regardless of project requirements
C. There is no meaningful difference between the two repository strategies
D. Monorepo stores all projects in one repo; polyrepo uses separate repos

Answer: D

Q1114. What is Infrastructure as Code (IaC)?

A. Managing infrastructure through configuration files not manual processes
B. A programming paradigm for organizing code into functional modules
C. Writing source code directly on production servers for deployment
D. A coding standard for formatting and naming conventions in code

Answer: A

Q1115. What is the concept of 'shift left' in testing?

A. Moving code directly to production without any testing or review
B. Left-aligning all source code for consistent formatting readability
C. Performing testing earlier in the development lifecycle for detection
D. Moving test files to the left side of the project directory tree

Answer: C

Q1116. What is feature flagging/toggling?

A. A branching strategy for organizing feature development in Git
B. A testing tool for validating feature behavior before release
C. Deleting features permanently from the codebase and all traces
D. Enabling or disabling features at runtime without deploying code

Answer: D

Q1117. What is the strangler fig pattern for legacy modernization?

A. Gradually replacing legacy systems by building new functionality around
B. A refactoring tool for automatically restructuring legacy code
C. A tree-related data structure pattern for hierarchical organization
D. Deleting all old code at once and replacing with a new system

Answer: A

Q1118. What is observability in software systems?

A. A debugging tool for stepping through code execution line by line
B. Watching code run step by step in a debugger tool window
C. Monitoring only server uptime without any additional data collection
D. Understanding system state from external outputs like logs and metrics

Answer: D

Q1119. What is the CAP theorem and how does it affect development practices?

A. A design pattern for organizing code into reusable modular components
B. In distributed systems, only two of Consistency, Availability, Partition tolerance
C. A coding principle for writing clean and maintainable source code
D. A testing theorem for determining the minimum number of test cases

Answer: B

Q1120. What is the concept of 'toil' in Site Reliability Engineering (SRE)?

A. Testing software to validate that it meets all specified requirements
B. Manual repetitive automatable work scaling linearly with service size
C. Hard work that requires significant effort and long hours from team
D. Debugging code to find and fix errors in the application source

Answer: B

Q1121. A program compiles without errors but produces incorrect output for certain edge cases. What type of error is this?

A. A runtime error causing immediate termination
B. A linker error from missing library references
C. A syntax error caught by the compiler during build
D. A logical error in the algorithm for edge cases

Answer: D

Q1122. In a multi-pass compiler, what is the primary advantage of separating analysis into multiple passes?

A. It allows forward references and better optimization
B. It makes the compiler run faster by skipping checks
C. It eliminates the need for any optimization phase
D. It removes the requirement for a symbol table

Answer: A

Q1123. A program works on the developer machine but fails in production. Which factor is most likely the cause?

A. The source code changed itself during the deployment
B. Environment differences such as OS or library versions
C. The language behaves differently on each machine
D. The compiler randomly generates different code each run

Answer: B

Q1124. What is the key difference between ahead-of-time (AOT) and just-in-time (JIT) compilation?

A. AOT requires an interpreter; JIT does not need one
B. JIT always produces slower code than AOT in every case
C. AOT compiles before execution; JIT compiles during runtime
D. AOT is only for scripting; JIT only for systems languages

Answer: C

Q1125. Why might a developer choose a dynamically typed language for rapid prototyping?

A. Dynamic typing makes programs run faster on all hardware
B. Dynamic typing enables faster iteration by skipping types
C. Dynamic typing always produces more efficient machine code
D. Dynamic typing prevents all runtime errors from occurring

Answer: B

Q1126. What problem does the halting problem describe in computer science?

A. The difficulty of stopping an infinite loop in a program
B. The challenge of compiling large programs within limits
C. The problem of optimizing code for memory constraints
D. The impossibility of deciding if any program will halt

Answer: D

Q1127. A language uses type inference to determine variable types. What is the main benefit?

A. It makes the language dynamically typed at runtime
B. It prevents the compiler from performing optimization
C. It combines static type safety with reduced verbosity
D. It eliminates type safety entirely from the language

Answer: C

Q1128. In language design, what does homoiconicity mean?

A. The language can only represent one data type
B. Code and data share the same representation
C. The language does not support metaprogramming
D. The compiler and interpreter are identical

Answer: B

Q1129. What is the primary purpose of an abstract syntax tree (AST) in compilation?

A. To store the final machine code for the target
B. To handle input and output during execution
C. To represent hierarchical syntactic structure
D. To manage memory allocation during runtime

Answer: C

Q1130. A team debates compiled vs interpreted for a real-time system. What is the strongest argument for compiled?

A. Compiled languages always have simpler syntax
B. Compiled languages are easier for beginners
C. Compiled languages need no testing before use
D. Compiled languages produce optimized native code

Answer: D

Q1131. A financial app uses float for currency and has rounding errors. What is the recommended fix?

A. Convert all values to strings before performing calculations
B. Use double type since it eliminates all rounding errors
C. Increase float precision by adding more decimal places
D. Use integer types storing values in cents to avoid issues

Answer: D

Q1132. What is the concept of type erasure in Java generics?

A. Type information is stored in a separate metadata file
B. The compiler removes all type info making code untyped
C. The JVM dynamically creates new types for each generic use
D. Generic type parameters are replaced with bounds at compile

Answer: D

Q1133. A developer encounters NaN in JavaScript. Which operation could produce this result?

A. Multiplying a string that cannot be parsed as a number
B. Comparing two floating-point numbers with strict equality
C. Concatenating two numeric strings together into one
D. Dividing a positive integer by a negative integer value

Answer: A

Q1134. Why does comparing floating-point numbers with exact equality often fail?

A. The equality operator does not work with numeric types
B. Binary representation introduces tiny precision errors
C. Floating-point numbers are stored as strings internally
D. Floating-point comparisons are not supported by CPUs

Answer: B

Q1135. In Rust, what is the purpose of the Option<T> type?

A. It provides thread-safe access to shared mutable data
B. It stores multiple values of different types at once
C. It safely represents a value that may or may not exist
D. It optimizes memory allocation for large structures only

Answer: C

Q1136. What is a phantom type in languages with advanced type systems?

A. A type that cannot be instantiated under any circumstances
B. A type that stores data in an encrypted format always
C. A type that automatically converts between other data types
D. A type existing only at compile time using no runtime data

Answer: D

Q1137. A C program assigns a negative signed integer to an unsigned variable. What is the likely outcome?

A. The runtime throws an overflow exception and terminates
B. The program will refuse to compile due to a type mismatch
C. The unsigned variable automatically stores zero as default
D. The value becomes a large positive number via reinterpretation

Answer: D

Q1138. What problem do algebraic data types (ADTs) solve in functional programming?

A. They eliminate the need for functions in functional code
B. They provide automatic serialization to network formats
C. They optimize memory by compressing data automatically
D. They model complex data as combinations of sum and product

Answer: D

Q1139. Why does JavaScript treat typeof null as object instead of null?

A. Because null and objects share the same memory structure
B. Because null is implemented as an empty object internally
C. Because of a legacy bug in the original JS implementation
D. Because the spec intentionally groups null with objects

Answer: C

Q1140. What is the difference between structural typing and nominal typing?

A. Structural checks type names; nominal checks type shape
B. Nominal typing does not support inheritance between types
C. Structural checks type shape; nominal checks type names
D. Structural typing is slower than nominal in every case

Answer: C

Q1141. A developer uses bitwise AND to check if a number is odd. Which expression correctly identifies odd numbers?

A. (n & 2) == 1 returns true for odd numbers in binary
B. (n & 1) == 0 returns true because odd clears last bit
C. (n & 0) == 1 returns true for odd numbers correctly
D. (n & 1) == 1 returns true because odd last bit is set

Answer: D

Q1142. In C++, what potential issue arises from overloading the comma operator?

A. It may break the guaranteed left-to-right evaluation
B. It causes compilation errors in all template-based code
C. The comma operator cannot be overloaded in C++ standard
D. It automatically converts all expressions to void type

Answer: A

Q1143. What does the unsigned right shift (>>>) do differently from signed right shift (>>)?

A. It shifts bits right and preserves the sign bit in position
B. It rotates bits right wrapping them around to the left side
C. It shifts bits left instead of right despite arrow direction
D. It shifts bits right and fills with zeros regardless of sign

Answer: D

Q1144. A Python developer writes 'a is b' vs 'a == b'. When do these give different results?

A. When a and b reference the exact same object in memory
B. When a and b are the same integer literal used repeatedly
C. When a and b have equal values but are different objects
D. When both a and b are None compared in an if statement

Answer: C

Q1145. What is the result of applying De Morgan's law to !(A && B)?

A. (!A || !B) distributes the negation and flips the operator
B. (!A && !B) negation of each operand combined with AND
C. (!A && B) negates only the first operand in expression
D. (A || B) simply removes the negation from the expression

Answer: A

Q1146. Why should developers avoid relying on operator precedence in complex expressions?

A. Because parentheses are not allowed with operator precedence
B. Because complex precedence reduces readability and causes bugs
C. Because operator precedence varies between every language
D. Because operator precedence only applies to arithmetic ops

Answer: B

Q1147. What is the purpose of the pipe operator (|>) in functional languages like Elixir?

A. It passes the result of one expression as input to next
B. It redirects standard output to a file on the system
C. It creates a new parallel process for each function call
D. It performs bitwise OR operations on integer operands

Answer: A

Q1148. In JavaScript, what is the result of {} + [] when evaluated as a statement?

A. A type error because objects and arrays cannot be added
B. The string '{}[]' because both operands convert to string
C. An empty object combined with empty array resulting in NaN
D. The number 0 because {} is empty block and +[] coerces to 0

Answer: D

Q1149. What advantage does the walrus operator (:=) provide in Python 3.8+?

A. It performs parallel assignment of multiple variables at once
B. It creates constants that cannot be reassigned after init
C. It defines default parameter values for function declarations
D. It assigns a value and returns it within a single expression

Answer: D

Q1150. How does operator precedence interact with associativity when precedence is equal?

A. Associativity determines evaluation order for same precedence
B. The compiler randomly selects which operator to evaluate first
C. Operators with equal precedence always evaluate left to right
D. Operators with equal precedence cannot appear in one expression

Answer: A

Q1151. A function uses recursion with no base case. What will happen during execution?

A. The compiler automatically adds a base case to stop
B. The call stack overflows causing a stack overflow error
C. The function returns null after one recursive call
D. The function loops exactly once then exits normally

Answer: B

Q1152. In a switch with intentional fall-through, what is the best practice for maintainability?

A. Convert the entire switch to unrelated if statements
B. Remove all break statements for code consistency
C. Add explicit comments documenting intentional fall-through
D. Use goto statements instead of fall-through for clarity

Answer: C

Q1153. A developer replaces nested if-else with a strategy pattern. What is the main benefit?

A. Memory usage is reduced because fewer objects are created
B. The code executes faster because polymorphism is quicker
C. The number of classes in the codebase decreases greatly
D. Each strategy is encapsulated making code more extensible

Answer: D

Q1154. What is the time complexity of a triply nested loop where each iterates n times?

A. O(n cubed) because each nesting multiplies by factor n
B. O(n) because only the innermost loop affects complexity
C. O(n squared) because the outer loop does not count
D. O(log n) because nested loops divide problem recursively

Answer: A

Q1155. How does continuation-passing style (CPS) change program control flow?

A. It uses goto statements to jump between code sections
B. It eliminates all function calls using only loops instead
C. It converts recursive functions into iterative ones always
D. It passes a callback representing the next step to each

Answer: D

Q1156. A loop uses floating-point comparison as its termination condition. Why is this risky?

A. The loop will always terminate early before full completion
B. Floating-point loops always execute one extra iteration
C. Floating-point numbers cannot be used in loop conditions
D. Precision errors may prevent termination condition triggering

Answer: D

Q1157. What is the advantage of iterators over index-based loops for data structures?

A. Iterators use less memory than index-based loops in all cases
B. Iterators always process elements faster than index access
C. Iterators prevent any modifications to underlying collection
D. Iterators provide a uniform interface across data structures

Answer: D

Q1158. What problem does the loop-and-a-half pattern solve?

A. It creates loops executing exactly one and a half times
B. It converts while loops into for loops for performance
C. It handles loops needing to check condition in middle
D. It optimizes loops to run at half the time of regular

Answer: C

Q1159. How do higher-order functions like map and filter replace loops in functional programming?

A. They convert loops into goto statements for faster speed
B. They apply transformations declaratively without iteration
C. They eliminate the need for conditional logic in programs
D. They only work with numeric data not other types at all

Answer: B

Q1160. Why might a compiler transform a recursive function into an iterative loop?

A. Because iteration uses more memory improving caching
B. Because recursive functions always produce wrong output
C. Because iteration avoids stack overhead and overflow
D. Because recursion is not supported by modern CPUs

Answer: C

Q1161. A recursive function for Fibonacci has exponential time complexity. How can this be optimized?

A. By converting the function parameters to global variables
B. By using memoization to cache previously computed values
C. By adding more recursive calls to speed up computation
D. By removing the base case to reduce function calls

Answer: B

Q1162. What is the Y combinator and what problem does it solve in lambda calculus?

A. It validates the types of arguments passed to functions
B. It combines two functions into a single optimized one
C. It enables recursion in languages without named functions
D. It converts iterative loops into recursive function calls

Answer: C

Q1163. A function modifies a global variable as a side effect. Why is this considered poor practice?

A. Because side effects always cause the program to crash immediately
B. Because side effects make functions unpredictable and harder to test
C. Because global variables use too much memory in programs
D. Because programming languages prohibit modifying global variables

Answer: B

Q1164. What is the difference between currying and partial application?

A. Currying combines multiple functions; partial splits them into pieces
B. Currying splits a function into single-arg chain; partial fixes some args
C. Currying and partial application are exactly the same technique always
D. Currying only works with two arguments; partial works with any count

Answer: B

Q1165. How does tail call optimization (TCO) prevent stack overflow in recursive functions?

A. It moves recursive calls to a separate memory heap location
B. It converts all recursive calls into asynchronous operations
C. It limits the depth of recursion to a fixed maximum count
D. It reuses the current stack frame for the next recursive call

Answer: D

Q1166. What is a thunk in the context of lazy evaluation?

A. A deferred computation wrapped in a zero-argument function
B. A data structure that stores function parameter information
C. A compiled machine code fragment stored in memory cache
D. An error handler that catches exceptions in lazy code blocks

Answer: A

Q1167. A developer uses function composition f(g(x)). What is the main benefit of this approach?

A. It builds complex operations from simple reusable functions
B. It eliminates the need for any error handling in code
C. It guarantees the composed functions execute in parallel
D. It ensures both functions always return the same data type

Answer: A

Q1168. What is a functor in functional programming?

A. An object that stores multiple functions in an internal array
B. A function that can only be called once during execution
C. A special variable that holds a reference to a function pointer
D. A type that can be mapped over with a structure-preserving function

Answer: D

Q1169. Why are first-class functions important for functional programming paradigms?

A. They enable functions to be passed, returned, and assigned like data
B. They prevent functions from having any parameters or return values
C. They allow functions to be stored in databases for persistence
D. They make all functions execute faster than in other paradigms

Answer: A

Q1170. What problem does the trampolining technique solve for recursive functions?

A. It automatically parallelizes recursive calls across CPU cores
B. It prevents stack overflow by converting recursion to loop steps
C. It speeds up recursive calculations by using hardware acceleration
D. It eliminates the need for base cases in recursive algorithms

Answer: B

Q1171. A program reads a large file line by line instead of loading it all at once. What is the main advantage?

A. It prevents the file from being modified by other processes
B. Line-by-line reading is always faster than loading at once
C. It automatically handles character encoding conversion issues
D. It reduces memory usage by processing one line at a time

Answer: D

Q1172. What is memory-mapped I/O and when is it beneficial?

A. Allocating extra RAM to speed up sequential file reading
B. Mapping files into virtual memory for faster random access
C. Storing all program variables in external files on disk
D. Converting memory addresses to file paths automatically

Answer: B

Q1173. A multi-threaded program writes to the same file simultaneously. What problem can occur?

A. Race conditions cause corrupted or interleaved file data
B. Each thread creates its own separate copy of the file
C. The file automatically becomes read-only preventing writes
D. The operating system duplicates the file for each thread

Answer: A

Q1174. What is the difference between blocking and non-blocking I/O in server design?

A. Blocking handles more connections; non-blocking handles fewer
B. Non-blocking cannot read from network sockets at all ever
C. Blocking waits for completion; non-blocking returns immediately
D. Blocking is only for files; non-blocking only for networks

Answer: C

Q1175. What is the purpose of a write-ahead log (WAL) in database I/O?

A. It logs all read operations for performance monitoring only
B. It encrypts data before writing to prevent unauthorized access
C. It compresses write operations to reduce disk space usage
D. It records changes before applying them for crash recovery

Answer: D

Q1176. A developer needs to parse a 10GB XML file. Why would a SAX parser be preferred over DOM?

A. DOM parsers cannot handle files containing nested elements
B. SAX processes XML as a stream without loading it all in memory
C. SAX is more accurate at parsing XML structure than DOM
D. SAX automatically validates XML against its schema definition

Answer: B

Q1177. What is the reactor pattern in asynchronous I/O programming?

A. A technique for converting synchronous code to asynchronous
B. A single-threaded event loop dispatching I/O events to handlers
C. A method of buffering all output until program termination
D. A pattern that creates a new thread for every I/O request

Answer: B

Q1178. How does zero-copy I/O improve performance compared to traditional I/O?

A. It avoids copying data between kernel and user space buffers
B. It compresses data to zero bytes before transferring it out
C. It uses zero system calls by directly accessing hardware
D. It eliminates the need for any disk storage by using RAM only

Answer: A

Q1179. What is the purpose of the fsync system call?

A. It synchronizes file access between multiple running threads
B. It converts file system format between different OS types
C. It synchronizes file timestamps with the system clock value
D. It ensures file data is physically written to storage device

Answer: D

Q1180. A program handles both file I/O and network I/O. Why might it use an event-driven architecture?

A. Event-driven guarantees faster disk I/O than any alternative
B. Event-driven efficiently handles many concurrent I/O sources
C. Event-driven converts network data to file format automatically
D. Event-driven eliminates all errors in I/O operations always

Answer: B

Q1181. A program compares user passwords using == instead of a constant-time comparison. What is the security risk?

A. The program will crash when processing long password strings
B. The comparison may return incorrect results for all input
C. The passwords will be stored in plain text in memory always
D. Timing attacks can reveal password characters one by one

Answer: D

Q1182. Why is the KMP algorithm preferred over naive string matching for large texts?

A. KMP requires the pattern to be sorted before searching starts
B. KMP uses more memory but is simpler to implement correctly
C. KMP avoids rechecking characters using a failure function table
D. KMP only works with numeric strings not alphabetic text data

Answer: C

Q1183. What is the purpose of string interning and when can it cause problems?

A. It translates strings between languages causing encoding issues
B. It encrypts strings for security but slows down all comparisons
C. It caches unique strings saving memory but can cause leaks if overused
D. It compresses strings for storage but prevents substring extraction

Answer: C

Q1184. How does the Rabin-Karp algorithm use hashing for string matching?

A. It creates a hash table of all possible patterns before searching
B. It hashes the entire text once and compares with pattern hash
C. It uses rolling hash to efficiently compare pattern with substrings
D. It hashes each character individually and sorts them for matching

Answer: C

Q1185. What is a rope data structure and when is it better than a standard string?

A. A rope compresses strings to use minimal memory allocation
B. A rope converts strings to binary for faster CPU processing
C. A rope stores strings in encrypted form for better security
D. A rope is a tree of small strings efficient for large text edits

Answer: D

Q1186. What is the difference between UTF-8 and UTF-16 encoding?

A. UTF-8 uses 8 bits per character always; UTF-16 uses 16 always
B. UTF-8 only supports English; UTF-16 supports all world languages
C. UTF-8 uses variable 1-4 bytes; UTF-16 uses 2 or 4 byte units
D. UTF-8 is binary format; UTF-16 is a text-based format only

Answer: C

Q1187. A developer finds that reversing a string containing emoji produces corrupted output. What is the cause?

A. The reversal algorithm has a bug unrelated to character type
B. Emoji use multi-code-unit sequences broken by naive reversal
C. Emoji characters are not valid in any programming language
D. The string library does not support any non-ASCII characters

Answer: B

Q1188. What is the purpose of normalization forms (NFC, NFD) in Unicode string processing?

A. They ensure equivalent character sequences have canonical form
B. They convert strings from Unicode to ASCII encoding format
C. They validate that strings contain only printable characters
D. They compress Unicode strings to reduce their storage size

Answer: A

Q1189. Why might a trie data structure be preferred for autocomplete over a hash map of strings?

A. Tries automatically sort strings in reverse alphabetical order
B. Tries use less memory than hash maps for all data sizes
C. Tries are faster for exact match lookups than hash maps
D. Tries support efficient prefix-based searching and traversal

Answer: D

Q1190. What is a homoglyph attack in the context of string security?

A. An attack that overflows string buffers with excess data
B. An attack that deletes all string data from the database
C. An attack that encrypts strings making them unreadable
D. An attack using visually similar characters to deceive users

Answer: D

Q1191. A developer uses a HashMap with mutable objects as keys. What problem can occur?

A. The map automatically prevents mutation of all key objects
B. The map will throw an error when any key is first inserted
C. Mutable keys cause the map to sort entries in reverse order
D. Mutating keys changes their hash, making entries unretrievable

Answer: D

Q1192. What is the amortized time complexity of adding an element to a dynamic array (ArrayList)?

A. O(log n) because dynamic arrays use binary search for insert
B. O(n squared) because each insertion triggers a full array sort
C. O(n) because every insertion requires copying all elements
D. O(1) amortized because resizing cost is spread over operations

Answer: D

Q1193. What is a bloom filter and what tradeoff does it make?

A. A compression technique that trades accuracy for file size
B. A probabilistic structure allowing false positives not negatives
C. A sorting algorithm that trades stability for speed
D. A search algorithm trading memory for faster query execution

Answer: B

Q1194. Why is a red-black tree preferred over a simple BST for implementing TreeMap?

A. Red-black trees guarantee O(log n) by maintaining balance always
B. Red-black trees use less memory than simple BSTs in all cases
C. Red-black trees do not require any comparison between elements
D. Simple BSTs cannot store key-value pairs only single values

Answer: A

Q1195. What is the difference between a B-tree and a binary search tree?

A. B-trees cannot be balanced while binary search trees can be
B. B-trees have multiple children per node optimized for disk I/O
C. B-trees store only two children per node like binary trees
D. B-trees only work with string keys not with numeric values

Answer: B

Q1196. A concurrent program uses a regular HashMap and encounters data corruption. What is the fix?

A. Use ConcurrentHashMap which provides thread-safe operations
B. Use a larger initial capacity for the HashMap allocation
C. Add more entries to the HashMap to prevent hash collisions
D. Convert the HashMap to a TreeMap for sorted data access

Answer: A

Q1197. What is a skip list and what advantage does it offer?

A. A list that skips every other element for faster traversal
B. A list that automatically removes duplicate elements on insert
C. A probabilistic structure providing O(log n) search in linked list
D. A list optimized for storing very large string values only

Answer: C

Q1198. What is the difference between a weak reference map and a regular map?

A. Weak maps allow garbage collection of keys not strongly referenced
B. Weak maps are faster for lookup operations than regular maps
C. Weak maps cannot store null values while regular maps can
D. Weak maps store values with less precision than regular maps

Answer: A

Q1199. What is cache-oblivious algorithm design and how does it relate to array processing?

A. It disables caching to ensure consistent performance results
B. It ignores CPU cache entirely for simpler algorithm design
C. It fills the CPU cache completely before processing any data
D. It optimizes memory access patterns without knowing cache size

Answer: D

Q1200. Why does a hash table with a poor hash function degrade to O(n) lookup time?

A. Because many collisions cluster entries into the same bucket
B. Because poor hash functions produce negative index values
C. Because poor hash functions prevent the table from resizing
D. Because the hash table automatically switches to linear search

Answer: A

Q1201. A developer violates the Single Responsibility Principle by creating a class that handles both data validation and database storage. What is the consequence?

A. The class automatically becomes thread-safe for concurrent use
B. The compiler will refuse to compile the class with both concerns
C. The class will execute faster due to fewer method calls needed
D. Changes to validation logic may unintentionally break storage code

Answer: D

Q1202. What is the visitor pattern and when should it be used?

A. A pattern for creating multiple instances of a single class
B. A pattern to add operations to objects without modifying classes
C. A pattern for sorting objects in a collection by their type
D. A pattern for managing user sessions in web applications

Answer: B

Q1203. How does the SOLID Open/Closed Principle guide class design?

A. Classes should be open for both extension and modification always
B. Classes should be closed for both extension and modification
C. Classes should be open for extension and closed for modification
D. Classes should be open for modification and closed for extension

Answer: C

Q1204. What is the difference between covariance and contravariance in generic types?

A. Both covariance and contravariance preserve subtype ordering
B. Neither covariance nor contravariance affects subtype ordering
C. Covariance preserves subtype ordering; contravariance reverses it
D. Covariance reverses subtype ordering; contravariance preserves it

Answer: C

Q1205. What is the template method pattern and how does it use inheritance?

A. It generates code templates for creating new class files quickly
B. It replaces all inheritance with composition for better flexibility
C. It creates multiple templates of the same class for different uses
D. It defines algorithm skeleton in base class deferring steps to sub

Answer: D

Q1206. Why should developers prefer composition over inheritance in most cases?

A. Because inheritance is not supported in modern languages anymore
B. Because composition provides more flexibility and looser coupling
C. Because composition always uses less memory than inheritance does
D. Because inheritance cannot be used with interfaces in any language

Answer: B

Q1207. What is double dispatch and how does the visitor pattern implement it?

A. Resolving a method call based on two object types at runtime
B. Dispatching a method call to two different servers simultaneously
C. Creating two instances of the same class for parallel processing
D. Calling the same method twice with different arguments each time

Answer: A

Q1208. What problem does the fragile base class issue describe?

A. Changes to base class may unexpectedly break derived classes
B. Base classes consume too much memory in the inheritance chain
C. Base classes cannot have more than one level of inheritance
D. Derived classes always override every method in the base class

Answer: A

Q1209. How does the CQRS pattern separate read and write operations in OOP design?

A. It combines all reads and writes into a single unified model
B. It uses separate models for reading queries and writing commands
C. It prevents any reading operations during write transactions
D. It uses different programming languages for reads and writes

Answer: B

Q1210. What is the expression problem and how do OOP and FP approach it differently?

A. Both OOP and FP solve the expression problem in identical ways
B. OOP easily adds new operations; FP easily adds new types always
C. Neither OOP nor FP can add new types or operations at all ever
D. OOP easily adds new types; FP easily adds new operations on types

Answer: D

Q1211. A program has a memory leak that only manifests after running for several days. What approach best identifies the leak?

A. Adding more RAM to the server to delay the out-of-memory crash
B. Using a memory profiler to track allocation growth over time
C. Reviewing source code manually for every allocation and free
D. Restarting the program more frequently to prevent the leak

Answer: B

Q1212. What is the generational hypothesis in garbage collection?

A. Older objects are more likely to be collected than newer ones
B. Most objects die young so collecting young generation is efficient
C. Garbage collection should run only once per program execution
D. All objects have the same lifetime regardless of creation time

Answer: B

Q1213. What is a use-after-free vulnerability and how can it be exploited?

A. Freeing memory twice causing the allocator to add extra memory
B. Accessing freed memory that may contain attacker-controlled data
C. Using free software after a trial period has expired completely
D. Using a variable after its scope ends causing a compiler error

Answer: B

Q1214. How does Rust's ownership system prevent memory safety bugs at compile time?

A. Each value has one owner and memory is freed when owner goes out of scope
B. It prevents all heap allocation forcing everything onto the stack
C. It uses garbage collection that runs at compile time instead
D. It requires programmers to manually free every allocation made

Answer: A

Q1215. What is the difference between mark-and-sweep and mark-and-compact garbage collection?

A. Mark-sweep leaves objects in place; mark-compact moves them together
B. Mark-sweep moves live objects; mark-compact leaves them in place
C. Mark-compact cannot handle circular references unlike mark-sweep
D. Mark-sweep is faster than mark-compact in all possible scenarios

Answer: A

Q1216. What is a slab allocator and when is it used in operating systems?

A. An allocator that only works with very large memory blocks
B. An allocator that stores data on disk instead of in memory
C. An allocator caching fixed-size objects for frequent kernel allocations
D. An allocator that compresses memory to fit more data in RAM

Answer: C

Q1217. How does address space layout randomization (ASLR) improve memory security?

A. It encrypts all memory contents to prevent unauthorized reading
B. It randomizes memory layout making exploit addresses unpredictable
C. It prevents any program from allocating memory on the heap
D. It locks memory pages so they cannot be written to by code

Answer: B

Q1218. What is false sharing in multi-threaded programs and how does it affect performance?

A. Threads falsely reporting memory as free when it is still in use
B. Sharing read-only data between threads causing unnecessary copying
C. Multiple threads sharing the same variable causing data corruption
D. Threads on different cores invalidate cache lines they do not truly share

Answer: D

Q1219. What is the purpose of memory-mapped files and how do they differ from traditional file I/O?

A. They compress files in memory to reduce the total memory used
B. They encrypt files in memory to prevent unauthorized data reading
C. They store files entirely in CPU registers for faster access
D. They map file contents to virtual memory allowing direct pointer access

Answer: D

Q1220. What is a garbage collection pause and why is it problematic for real-time applications?

A. A delay caused by the GC writing collection logs to disk
B. A delay when the GC starts running for the very first time
C. A stop-the-world pause halting application threads during collection
D. A pause occurring only when the program first allocates memory

Answer: C

Q1221. A developer wraps every method in try-catch blocks that silently swallow exceptions. What is the consequence?

A. Memory usage decreases because exceptions are not stored
B. The application runs faster due to fewer exception propagations
C. The application becomes completely immune to all runtime errors
D. Errors are hidden making debugging extremely difficult and slow

Answer: D

Q1222. What is the difference between fault tolerance and error recovery in system design?

A. Fault tolerance continues despite faults; recovery restores state
B. Fault tolerance prevents all errors; recovery fixes them after
C. Fault tolerance is for hardware; recovery is for software only
D. Both terms mean exactly the same thing in all system contexts

Answer: A

Q1223. How does the circuit breaker pattern prevent cascading failures in distributed systems?

A. It duplicates every request to ensure at least one succeeds
B. It physically disconnects failing servers from the network
C. It stops calling a failing service after threshold allowing recovery
D. It encrypts all communication to prevent transmission errors

Answer: C

Q1224. What is the difference between fail-fast and fail-safe error handling strategies?

A. Fail-fast is for testing; fail-safe is for production environments
B. Both strategies handle errors in exactly the same manner always
C. Fail-fast ignores errors; fail-safe reports them immediately
D. Fail-fast reports errors immediately; fail-safe continues operating

Answer: D

Q1225. A microservice architecture experiences a cascading failure. Which pattern best prevents this?

A. Increasing timeout values to give failing services more time
B. Using bulkhead isolation to contain failures within boundaries
C. Adding more servers to handle the increased error load
D. Removing all error handling to let services fail naturally

Answer: B

Q1226. What is the purpose of structured error handling using Result types instead of exceptions?

A. Result types are faster to execute than exception handling code
B. Result types prevent any errors from occurring during runtime
C. Result types make error handling explicit in function signatures
D. Result types automatically fix errors without developer action

Answer: C

Q1227. How does the retry pattern with exponential backoff improve error handling?

A. It retries immediately without any delay between each attempt
B. It retries a fixed number of times with constant delay between
C. It decreases delay between retries to resolve errors faster
D. It increases delay between retries reducing load on failing service

Answer: D

Q1228. What is the difference between recoverable and unrecoverable errors in Rust?

A. Recoverable use Result type; unrecoverable use panic macro
B. Rust does not distinguish between recoverable and unrecoverable
C. Recoverable use panic macro; unrecoverable use Result type
D. Both types are handled using the same mechanism in Rust

Answer: A

Q1229. What is the supervisor pattern in Erlang/OTP for fault tolerance?

A. A design that prevents any process from ever failing at all
B. A thread that monitors CPU usage and kills slow processes
C. A process that monitors child processes and restarts on failure
D. A function that validates all input before processing it

Answer: C

Q1230. Why is it important to include contextual information when throwing custom exceptions?

A. Context helps developers quickly diagnose the root cause of errors
B. Context information makes exceptions run faster in production
C. Context reduces the memory footprint of exception objects created
D. Context automatically fixes the error without human intervention

Answer: A

Q1231. A project has conflicting versions of the same dependency required by two libraries. How is this typically resolved?

A. The project must choose one library and remove the other
B. Dependency resolution algorithms select a compatible version
C. Both versions are deleted and the dependency is removed
D. All dependencies are updated to the latest version available

Answer: B

Q1232. What is the difference between a plugin architecture and a monolithic architecture?

A. Plugins require more memory than monolithic architecture always
B. Monolithic supports more features than plugin architecture can
C. Plugins allow extending functionality without modifying core code
D. Plugins are slower; monolithic is always faster in execution

Answer: C

Q1233. What is dependency hell and what strategies mitigate it?

A. Dependencies that are too large consuming excessive disk space
B. Conflicting version requirements across the dependency graph
C. Having too few dependencies causing limited functionality
D. Dependencies that require payment to use in production apps

Answer: B

Q1234. How does lazy loading of modules improve application performance?

A. It defers loading modules until they are actually needed
B. It compresses modules to reduce their memory footprint
C. It loads all modules at startup to prevent later delays
D. It caches all modules permanently in browser local storage

Answer: A

Q1235. What is the difference between a micro-frontend architecture and a traditional monolithic frontend?

A. Micro-frontends split UI into independently deployable modules
B. Monolithic frontends load faster than micro-frontend solutions
C. Micro-frontends cannot share any state between components
D. Micro-frontends use only one framework; monolithic uses many

Answer: A

Q1236. What is a supply chain attack in the context of package management?

A. Compromising a dependency to inject malicious code downstream
B. An attack that corrupts the package manager configuration file
C. An attack that prevents packages from being downloaded online
D. An attack that targets the physical delivery of hardware

Answer: A

Q1237. What is the purpose of code splitting in modern web applications?

A. Separating code comments from executable statements in files
B. Splitting code across multiple programming languages for speed
C. Dividing code equally among team members for parallel coding
D. Breaking application code into chunks loaded on demand for speed

Answer: D

Q1238. How does the facade pattern simplify complex module interactions?

A. It removes dependencies between modules by duplicating all code
B. It hides all modules and prevents any external access to them
C. It converts all module functions into a single large function
D. It provides a simplified interface to a complex subsystem of modules

Answer: D

Q1239. What is the hexagonal architecture and how does it relate to module organization?

A. An architecture using exactly six modules in the application
B. An architecture isolating core logic from external adapters and ports
C. An architecture requiring six layers of abstraction in modules
D. An architecture that arranges modules in a hexagonal grid pattern

Answer: B

Q1240. What is the difference between hard and soft dependencies in module design?

A. Hard dependencies are faster; soft dependencies are always slower
B. Hard dependencies are for production; soft are for development only
C. Hard dependencies are required; soft are optional or gracefully absent
D. Both hard and soft dependencies behave identically in all cases

Answer: C

Q1241. What is the difference between greedy algorithms and dynamic programming approaches?

A. Both approaches always produce identical results for all problems
B. Greedy makes locally optimal choices; DP considers all subproblems
C. Greedy always finds the optimal solution; DP finds approximate
D. DP makes locally optimal choices; greedy considers all subproblems

Answer: B

Q1242. What is a monad in functional programming and what problem does it solve?

A. A pattern for chaining operations while managing side effects
B. A single unit of memory allocation in functional programs
C. A function that converts between different data type formats
D. A data type that can only hold one value at any given time

Answer: A

Q1243. How does the A* algorithm combine features of Dijkstra and greedy best-first search?

A. A* runs both algorithms separately and picks the better result
B. A* ignores heuristics and relies only on actual path cost found
C. A* uses actual cost plus heuristic estimate to guide its search
D. A* uses Dijkstra for graphs and greedy for trees exclusively

Answer: C

Q1244. What is the CAP theorem and how does it affect distributed system design?

A. It states only two of consistency, availability, partition tolerance
B. It applies only to single-machine systems not distributed systems
C. It guarantees all three properties simultaneously in all systems
D. It states systems can achieve consistency, availability, and partitions

Answer: A

Q1245. What is the difference between a trie and a balanced BST for string operations?

A. Tries cannot store strings with common prefixes unlike BSTs
B. Tries are slower for prefix queries than balanced BSTs always
C. Tries provide O(m) lookup by key length; BSTs provide O(m log n)
D. BSTs provide faster prefix-based searching than tries in all cases

Answer: C

Q1246. What is the actor model for concurrent programming?

A. A model that uses only one thread for all program operations
B. A model that prevents all concurrency to avoid race conditions
C. A model where threads share mutable state through locks
D. A model where actors communicate through asynchronous messages

Answer: D

Q1247. What problem does the flyweight pattern solve in software design?

A. It prevents objects from growing beyond a certain memory limit
B. It shares common state among many objects to reduce memory use
C. It makes heavy objects lighter by removing unnecessary methods
D. It converts heavyweight processes into lightweight threads only

Answer: B

Q1248. How does consistent hashing improve distributed hash table scalability?

A. It minimizes key redistribution when nodes are added or removed
B. It sorts all keys alphabetically across all nodes in the system
C. It eliminates all hash collisions in the distributed hash table
D. It ensures all hash values are consecutive integers in sequence

Answer: A

Q1249. What is the difference between optimistic and pessimistic concurrency control?

A. Optimistic locks before access; pessimistic assumes no conflict
B. Optimistic is for reads only; pessimistic is for writes exclusively
C. Both approaches use identical locking mechanisms for all access
D. Optimistic assumes no conflict and validates; pessimistic locks first

Answer: D

Q1250. What is event sourcing and how does it differ from traditional CRUD?

A. Event sourcing uses events only for logging not for state recovery
B. Event sourcing stores only the current state like CRUD does
C. Event sourcing stores all state changes as immutable event sequence
D. Event sourcing deletes old data automatically unlike CRUD approach

Answer: C

Q1251. What is lock-free programming and why is it challenging to implement correctly?

A. A technique that prevents any thread from ever being interrupted
B. A style that avoids all shared memory between concurrent threads
C. Using atomic operations for thread safety without traditional locks
D. Programming without any synchronization for maximum speed always

Answer: C

Q1252. What is the ABA problem in lock-free programming?

A. When a thread reads value A but another writes B before commit
B. When two threads both try to set the same value simultaneously
C. When atomic operations fail due to hardware memory limitations
D. When a value changes from A to B to A, making change undetectable

Answer: D

Q1253. How does the fork-join framework improve parallel computation?

A. It recursively splits tasks and merges results for parallelism
B. It joins multiple programs into one before executing them all
C. It creates a new thread for every single instruction to execute
D. It forks the program into separate processes that never rejoin

Answer: A

Q1254. What is the happens-before relationship in the Java memory model?

A. A rule preventing any reordering of instructions by the compiler
B. A requirement that all code must execute in written source order
C. A guarantee that one action's effects are visible to another action
D. A constraint requiring all threads to start in a specific order

Answer: C

Q1255. What is a memory barrier or fence instruction?

A. A mechanism for allocating memory in a thread-safe manner
B. A technique for protecting memory from unauthorized access
C. A hardware limit on the total amount of memory a program uses
D. An instruction ensuring memory operations complete in order

Answer: D

Q1256. How does software transactional memory (STM) simplify concurrent programming?

A. It allows composable atomic memory transactions without locks
B. It automatically detects and removes all race conditions found
C. It stores all thread data in a transactional database system
D. It converts multi-threaded code into single-threaded execution

Answer: A

Q1257. What is the thundering herd problem in concurrent systems?

A. When threads are distributed unevenly across available CPU cores
B. When a single thread consumes all available CPU resources alone
C. When many waiting threads wake simultaneously overwhelming resources
D. When many threads are created causing the system to slow down

Answer: C

Q1258. What is work stealing in parallel task scheduling?

A. Idle threads take tasks from busy threads' queues for balance
B. Threads duplicating work done by other threads for redundancy
C. Threads stealing CPU time from other threads for faster execution
D. The scheduler removing low-priority tasks to free up resources

Answer: A

Q1259. How does the read-copy-update (RCU) mechanism work in concurrent systems?

A. Readers lock data before reading; writers wait for all readers
B. All operations require exclusive locks for both reads and writes
C. Readers access data without locks; writers create updated copies
D. Data is copied to each thread's private memory before any access

Answer: C

Q1260. What is the difference between data parallelism and task parallelism?

A. Both data and task parallelism are identical concepts in practice
B. Task parallelism applies different operations to same data always
C. Data parallelism is for CPUs only; task parallelism for GPUs only
D. Data parallelism applies same operation to different data portions

Answer: D

Q1261. A test suite has 95% code coverage but critical bugs still reach production. What does this indicate?

A. That code coverage is a useless metric that should be abandoned
B. That coverage alone does not guarantee test quality or edge cases
C. That 95% coverage is too low and 100% is always required
D. That the remaining 5% of uncovered code contains all the bugs

Answer: B

Q1262. What is mutation testing and how does it evaluate test suite quality?

A. It tests the program with mutated or corrupted input data files
B. It modifies the test execution order to find order dependencies
C. It introduces small code changes to check if tests detect them
D. It randomly changes test inputs to find flaky test failures

Answer: C

Q1263. What is the difference between performance testing and load testing?

A. Performance tests measure speed; load tests measure correctness
B. Performance measures response under conditions; load tests capacity
C. Performance testing is manual; load testing is always automated
D. Both terms describe identical testing activities in all contexts

Answer: B

Q1264. A developer uses a memory profiler and finds objects that are referenced but never used. What is this problem called?

A. A segmentation fault caused by accessing invalid memory address
B. A buffer overflow caused by writing past array boundaries
C. A logical memory leak where references prevent garbage collection
D. A stack overflow caused by deep recursive function calls

Answer: C

Q1265. What is property-based testing and how does it differ from example-based testing?

A. Property-based verifies invariants with generated inputs randomly
B. Both approaches use identical methodologies for test generation
C. Property-based is slower and less effective than example-based
D. Property-based uses fixed examples; example-based uses random ones

Answer: A

Q1266. What is chaos engineering and how does it improve system reliability?

A. It intentionally injects failures to discover system weaknesses
B. It randomly deletes source code files to test backup systems
C. It introduces random code changes to test developer adaptability
D. It creates chaotic user interfaces to test user error handling

Answer: A

Q1267. What is the difference between static analysis and dynamic analysis?

A. Static runs code to find bugs; dynamic examines code without running
B. Static examines code without running; dynamic analyzes during execution
C. Static is for compiled languages only; dynamic for interpreted only
D. Both static and dynamic analysis require running the program code

Answer: B

Q1268. How does delta debugging systematically minimize a failing test case?

A. It deletes random lines until the test no longer demonstrates failure
B. It adds more test cases until the failure becomes easier to understand
C. It compares failing test with passing test to find code differences
D. It systematically removes portions to find minimal failure-inducing input

Answer: D

Q1269. What is the purpose of fuzzing as a testing technique?

A. Generating clear and readable test cases for documentation
B. Measuring the fuzziness or uncertainty of test result accuracy
C. Feeding random or malformed input to find crashes and vulnerabilities
D. Testing the user interface with blurred or fuzzy screen rendering

Answer: C

Q1270. What is the testing pyramid and what does it recommend for test distribution?

A. More unit tests at base, fewer integration, fewest UI tests at top
B. More UI tests at top, fewer unit tests at bottom of pyramid
C. Only unit tests with no integration or end-to-end tests at all
D. Equal numbers of each test type across all levels of the pyramid

Answer: A

Q1271. A team has a large codebase with no tests. What is the recommended approach to introduce testing?

A. Write tests for all code at once before making any changes
B. Hire a separate testing team to write all tests independently
C. Start by adding tests around code that changes most frequently
D. Delete all existing code and rewrite it with tests from scratch

Answer: C

Q1272. What is the strangler fig pattern for legacy system modernization?

A. Gradually replacing legacy components while keeping system running
B. Wrapping the legacy system in a new interface without changes
C. Immediately replacing the entire legacy system with new code
D. Documenting the legacy system before shutting it down entirely

Answer: A

Q1273. How does trunk-based development differ from GitFlow?

A. Both approaches use identical branching and merging strategies
B. Trunk-based uses short-lived branches merging to main frequently
C. Trunk-based does not use version control; GitFlow requires it
D. Trunk-based uses long-lived branches; GitFlow uses short-lived ones

Answer: B

Q1274. What is the twelve-factor app methodology?

A. A methodology requiring exactly twelve source code files per app
B. A set of best practices for building scalable cloud-native apps
C. A testing methodology requiring twelve types of tests minimum
D. A security framework with twelve levels of access permissions

Answer: B

Q1275. What is observability in software systems and how does it differ from monitoring?

A. Observability infers internal state from outputs; monitoring tracks metrics
B. Observability and monitoring are identical concepts in practice
C. Monitoring provides deeper insight than observability in all cases
D. Observability only tracks errors; monitoring tracks all system events

Answer: A

Q1276. What is the purpose of feature flags in software deployment?

A. They indicate which programming language features are being used
B. They enable toggling features on or off without code deployment
C. They flag code that contains bugs for developers to fix later
D. They mark features as complete in the project management tool

Answer: B

Q1277. How does domain-driven design (DDD) improve complex software projects?

A. It restricts development to a single programming language only
B. It aligns code structure with business domain using ubiquitous language
C. It eliminates the need for any testing in the development process
D. It focuses on optimizing database domain and query performance

Answer: B

Q1278. What is the difference between horizontal and vertical scaling strategies?

A. Horizontal adds more machines; vertical adds resources to existing
B. Horizontal adds resources to one machine; vertical adds more machines
C. Horizontal is for databases only; vertical is for applications only
D. Both strategies add resources in exactly the same way to systems

Answer: A

Q1279. What is the concept of eventual consistency in distributed systems?

A. All nodes always have the exact same data at every point in time
B. Only the primary node has consistent data; replicas do not
C. Data consistency is never achieved in any distributed system
D. All nodes will converge to the same state given enough time

Answer: D

Q1280. What is the purpose of a service mesh in microservice architectures?

A. It handles service-to-service communication, security, and observability
B. It combines all microservices into a single monolithic application
C. It stores all microservice data in a single shared database
D. It generates source code for new microservices automatically

Answer: A

Q1281. How does the LLVM compiler infrastructure enable language-independent optimization?

A. By using a common intermediate representation that multiple language frontends can target
B. By interpreting all programs at runtime without compilation
C. By compiling all languages directly to x86 machine code
D. By requiring all languages to use the same syntax

Answer: A

Q1282. What is the difference between structural and nominal type equivalence?

A. Structural equivalence is used only in compiled languages
B. Nominal equivalence ignores all type information
C. They produce identical results in every programming language
D. Structural equivalence compares type structure while nominal equivalence compares type names

Answer: D

Q1283. What is the Church-Turing thesis and its significance for programming?

A. It proves that all programs will eventually halt
B. It states that any computable function can be computed by a Turing machine
C. It defines the maximum speed of any computer program
D. It establishes that compiled code is always faster than interpreted code

Answer: B

Q1284. What is the expression problem in programming language design?

A. The challenge of parsing mathematical expressions efficiently
B. The problem of evaluating boolean expressions with short-circuit logic
C. The difficulty of adding both new data types and new operations without modifying existing code
D. The difficulty of handling operator precedence in complex expressions

Answer: C

Q1285. How does dependent typing differ from traditional static typing?

A. Dependent typing eliminates the need for any type annotations
B. Dependent types only work with integer values
C. Dependent typing is another name for dynamic typing
D. Dependent types allow types to depend on runtime values, enabling more precise type specifications

Answer: D

Q1286. What is the difference between call-by-name and call-by-need evaluation strategies?

A. Call-by-name re-evaluates arguments each time they are used while call-by-need caches the result after first evaluation
B. Call-by-need always evaluates arguments before passing them
C. Call-by-name prevents all side effects in function calls
D. Call-by-name is the same as call-by-value

Answer: A

Q1287. What is effect typing and how does it extend traditional type systems?

A. It measures the performance impact of type checking
B. It tracks computational side effects in the type system alongside value types
C. It replaces all runtime errors with compile-time type errors
D. It only applies to visual effects in graphical programming

Answer: B

Q1288. What is the significance of the lambda calculus as a foundation for programming languages?

A. It establishes the rules for object-oriented inheritance hierarchies
B. It provides a minimal formal system that can express any computable function using only function abstraction and application
C. It is a specialized calculus for mathematical computation only
D. It defines how compilers optimize machine code instructions

Answer: B

Q1289. What is gradual typing and what problem does it solve?

A. It allows mixing statically typed and dynamically typed code in the same program
B. It gradually removes type annotations as the program matures
C. It automatically converts all types to strings at runtime
D. It types variables based on the order they are declared

Answer: A

Q1290. What is the difference between total and partial functions in programming language theory?

A. Total functions are only found in imperative languages
B. Total functions handle all data types while partial functions handle only one
C. Partial functions are faster because they skip some computations
D. Total functions return a result for every possible input while partial functions may not terminate or may be undefined for some inputs

Answer: D

Q1291. What is a higher-kinded type and which languages support it?

A. A type reserved for advanced mathematical computations
B. A type that is always at the top of the inheritance hierarchy
C. A type constructor that takes other type constructors as parameters, supported by Haskell and Scala
D. A type that can store higher-precision decimal numbers

Answer: C

Q1292. What is the difference between covariance and contravariance in generic type parameters?

A. Covariance applies to input parameters and contravariance to output parameters
B. They both produce identical behavior in all type systems
C. Covariance preserves the type ordering while contravariance reverses it
D. Covariance is used only in dynamically typed languages

Answer: C

Q1293. What is a refinement type in programming languages?

A. A type that restricts a base type with a predicate specifying allowed values
B. A type that refines a class through inheritance only
C. A type that can only be used in pattern matching
D. A type that has been optimized for faster runtime performance

Answer: A

Q1294. How does Rust's ownership model interact with data types to ensure memory safety?

A. Rust uses garbage collection for all heap-allocated types
B. Rust copies all values automatically without restriction
C. All data types in Rust are allocated on the stack
D. Each value has exactly one owner, and the value is dropped when the owner goes out of scope

Answer: D

Q1295. What is the difference between existential types and universal types?

A. Existential types are used only for numbers while universal types handle strings
B. Universal types must be defined at compile time but existential at runtime
C. They are different names for the same concept
D. Universal types work for all types (forall) while existential types hide the concrete type (there exists)

Answer: D

Q1296. What is type-level programming and how is it used in practice?

A. Using types only for documentation purposes
B. Automatically generating types from database schemas
C. Programming without any type annotations
D. Writing programs that execute entirely at the type level during compilation to enforce complex constraints

Answer: D

Q1297. What is the difference between nominal subtyping and structural subtyping for interfaces?

A. Structural subtyping requires classes to declare they implement an interface
B. Nominal subtyping is only used in interpreted languages
C. Nominal subtyping requires explicit declaration of subtype relationships while structural subtyping checks compatibility by shape
D. They produce identical behavior in all object-oriented languages

Answer: C

Q1298. What is a linear type system and what guarantee does it provide?

A. A type system that arranges all types in a linear hierarchy
B. A type system that evaluates type expressions in linear time
C. A type system that only supports linear data structures
D. A type system that ensures every value is used exactly once, preventing resource leaks

Answer: D

Q1299. What is a GADT (Generalized Algebraic Data Type)?

A. A data type that can only hold algebraic expressions
B. An algebraic data type where constructors can specify different type parameters for the resulting type
C. A type alias for complex generic signatures
D. A standard generic type available in all languages

Answer: B

Q1300. What is the bottom type (Nothing/Never) and how is it used in type systems?

A. A type that is the subtype of all types and has no possible values, used for functions that never return
B. The default type assigned to uninitialized variables
C. A type used exclusively for error codes
D. A type representing the smallest possible integer value

Answer: A

Q1301. Why can operator overloading lead to maintenance problems in large codebases?

A. Because operator overloading always causes runtime performance degradation
B. Because most languages prohibit any form of operator overloading
C. Because overloaded operators cannot be used with primitive types
D. Because overloaded operators may behave in unexpected ways that differ from their conventional meaning

Answer: D

Q1302. How can bitwise operators be used to swap two variables without a temporary variable?

A. Bitwise operators cannot be used to swap variables
B. By using AND: a &= b; b &= a; a &= b swaps the values
C. By using XOR: a ^= b; b ^= a; a ^= b swaps the values
D. By using OR: a |= b; b |= a; a |= b swaps the values

Answer: C

Q1303. What is the purpose of the three-way comparison (spaceship) operator in modern languages?

A. It returns negative, zero, or positive to indicate less than, equal, or greater than in a single operation
B. It performs three sequential comparisons automatically
C. It compares three variables simultaneously
D. It creates a three-element tuple from two values

Answer: A

Q1304. What is the significance of De Morgan's laws for simplifying Boolean expressions?

A. They state that AND and OR operators are always interchangeable
B. They define the order of evaluation for nested Boolean expressions
C. They prove that all Boolean expressions can be reduced to a single operator
D. They state that NOT(A AND B) equals (NOT A) OR (NOT B) and NOT(A OR B) equals (NOT A) AND (NOT B)

Answer: D

Q1305. How does the nullish coalescing operator (??) differ from the logical OR (||) in JavaScript?

A. ?? only considers null and undefined as falsy while || considers 0, empty string, and false as falsy too
B. ?? works only with strings while || works with all types
C. They behave identically for all input values
D. || only considers null as falsy while ?? considers all falsy values

Answer: A

Q1306. What is the role of the comma operator in C and JavaScript?

A. It evaluates both operands from left to right and returns the value of the rightmost operand
B. It concatenates two strings with a comma separator
C. It splits a string into multiple parts
D. It creates an array containing both operands

Answer: A

Q1307. Why is using bit shifting for multiplication and division by powers of two considered an optimization?

A. Because bit shifting produces more accurate results than multiplication
B. Because multiplication operators are not supported for integers
C. Because bit shifting operations are executed in a single CPU cycle while multiplication may take multiple cycles
D. Because bit shifting works on all data types including strings

Answer: C

Q1308. What problem does the double-checked locking pattern solve using comparison operators?

A. It compares two objects twice to ensure deep equality
B. It validates input parameters using two different comparison methods
C. It reduces the overhead of synchronization by checking a condition before and after acquiring a lock
D. It performs two sequential comparisons for sorting stability

Answer: C

Q1309. What is the difference between strict equality and abstract equality in type-coercing languages?

A. Abstract equality never performs type conversion
B. Strict equality is only available in statically typed languages
C. Strict equality is slower than abstract equality in all cases
D. Strict equality compares value and type without coercion while abstract equality converts types before comparing

Answer: D

Q1310. How does the Elvis operator (?:) in Kotlin simplify null handling compared to traditional null checks?

A. It automatically throws an exception when null is encountered
B. It provides a default value when the left expression is null in a single concise expression
C. It converts null values to zero for arithmetic operations
D. It eliminates all null values from the program

Answer: B

Q1311. How does the visitor pattern implement double dispatch to avoid complex conditional logic?

A. By using reflection to dynamically invoke methods
B. By storing all behavior in a single switch statement
C. By using two virtual method calls to determine behavior based on both the visitor and element types at runtime
D. By using nested if-else chains with type checking

Answer: C

Q1312. What is the difference between structured exception handling and setjmp/longjmp in C?

A. Structured exceptions cannot cross function boundaries
B. setjmp/longjmp automatically free allocated memory during the jump
C. Structured exceptions use language constructs with proper stack unwinding while setjmp/longjmp perform non-local jumps without cleanup
D. setjmp/longjmp provide better type safety than structured exceptions

Answer: C

Q1313. How does Rust's match expression enforce exhaustive pattern matching?

A. It uses a default catch-all case automatically
B. The compiler requires all possible patterns to be covered, refusing to compile if any case is missing
C. It only works with integer values
D. It throws a runtime exception for unmatched patterns

Answer: B

Q1314. What is the difference between cooperative and preemptive multitasking in terms of control flow?

A. Cooperative multitasking automatically prevents deadlocks
B. Preemptive multitasking cannot handle more than two tasks
C. Cooperative multitasking requires tasks to explicitly yield control while preemptive multitasking allows the scheduler to interrupt tasks
D. Cooperative multitasking is always faster than preemptive

Answer: C

Q1315. How do algebraic effects provide a more composable alternative to traditional control flow mechanisms?

A. They eliminate all side effects from programs
B. They replace all loops with recursive functions
C. They separate the description of an effect from its handler, allowing different interpretations of the same effectful code
D. They automatically parallelize sequential code

Answer: C

Q1316. What is the purpose of the computed goto (or indirect branch) in low-level optimization?

A. It computes the result of a mathematical expression before branching
B. It creates multiple goto statements in a single line
C. It jumps to a target address computed at runtime, enabling efficient dispatch tables for interpreters
D. It optimizes recursive function calls by removing the call stack

Answer: C

Q1317. What is the relationship between monadic composition and control flow in functional programming?

A. Monads eliminate the need for conditional statements
B. Monads chain computations where each step depends on the previous result, encoding control flow as data transformations
C. Monads replace all loops with simple function calls
D. Monads are only used for mathematical computations

Answer: B

Q1318. How does speculative execution in modern CPUs interact with branch prediction for control structures?

A. The CPU always executes both branches simultaneously
B. The CPU predicts which branch will be taken and speculatively executes instructions along that path before the condition is resolved
C. Branch prediction only works with simple if-else structures
D. Speculative execution skips all conditional checks for performance

Answer: B

Q1319. What is the difference between call/cc (call with current continuation) and standard exception handling?

A. They are identical mechanisms with different names
B. call/cc is simpler to implement than exception handling
C. call/cc captures the entire remaining computation as a first-class value that can be invoked multiple times, while exceptions only unwind the stack
D. Exception handling can capture continuations but call/cc cannot

Answer: C

Q1320. Why is the halting problem relevant to static analysis of loops and control flow?

A. It proves that recursive functions always terminate
B. It shows that all loops will eventually halt if written correctly
C. It proves that no algorithm can determine in general whether an arbitrary loop will terminate, limiting what static analyzers can guarantee
D. It only applies to infinite loops in interpreted languages

Answer: C

Q1321. How does memoization improve the performance of recursive functions?

A. By executing recursive calls in parallel threads
B. By reducing the number of parameters a function accepts
C. By converting all recursive calls into iterative loops automatically
D. By caching previously computed results and returning them for repeated inputs instead of recomputing

Answer: D

Q1322. What is the difference between currying and partial application in functional programming?

A. Currying fixes arguments while partial application chains functions
B. They are identical techniques with different names
C. Currying transforms a function with multiple arguments into a chain of single-argument functions, while partial application fixes some arguments of a function
D. Currying only works with two arguments

Answer: C

Q1323. What is a continuation and how is it used in functional programming?

A. A representation of the rest of the computation at a given point, passed as an argument to control execution flow
B. A function that continues executing after an error
C. A loop that never terminates
D. A variable that persists between function calls

Answer: A

Q1324. What is the difference between a free variable and a bound variable in a closure?

A. Free variables are faster to access than bound variables
B. Free variables must be global variables
C. A bound variable is defined within the function while a free variable is defined in an enclosing scope and captured by the closure
D. Bound variables cannot be modified inside the function

Answer: C

Q1325. What is a functor in functional programming and how does it relate to function application?

A. A type that implements a map operation, allowing a function to be applied to wrapped values without unwrapping
B. A debugging tool for tracking function calls
C. A special type of constructor in OOP
D. A function that creates other functions

Answer: A

Q1326. How does the trampolining technique prevent stack overflow in languages without tail call optimization?

A. It increases the stack size dynamically at runtime
B. It wraps recursive calls in thunks and uses a loop to evaluate them iteratively, keeping the stack flat
C. It converts all functions to asynchronous operations
D. It splits recursive calls across multiple threads

Answer: B

Q1327. What is referential transparency and why is it important for reasoning about functions?

A. It ensures functions can reference variables in any scope
B. It requires all functions to be visible from any part of the program
C. It means all function names must clearly describe their purpose
D. An expression is referentially transparent if it can be replaced with its value without changing program behavior

Answer: D

Q1328. What is the difference between applicative functors and monads in terms of function application?

A. Applicative functors apply independent wrapped functions to wrapped values while monads allow dependent sequential computations
B. Applicative functors are always faster than monads
C. Applicative functors require mutable state
D. Monads cannot be used for error handling

Answer: A

Q1329. What is a fixed-point combinator and what role does it play in lambda calculus?

A. It finds the optimal parameters for a function
B. It converts any function into a constant function
C. It enables recursion in lambda calculus where functions cannot refer to themselves by name
D. It fixes errors in function definitions automatically

Answer: C

Q1330. How does defunctionalization transform higher-order functions into first-order programs?

A. By converting functions into global variables
B. By removing all function definitions from the program
C. By replacing function values with data constructors and using an apply function to dispatch on them
D. By inlining all function calls at compile time

Answer: C

Q1331. How does the proactor pattern differ from the reactor pattern in asynchronous I/O?

A. The reactor only works with network I/O while the proactor works with file I/O
B. The proactor blocks on I/O while the reactor does not
C. They are identical patterns with different names
D. The reactor demultiplexes events and dispatches to handlers while the proactor initiates operations and handles completion notifications

Answer: D

Q1332. What is the purpose of the sendfile() system call and how does it optimize file transfer?

A. It sends files as email attachments
B. It transfers data between file descriptors directly in kernel space without copying to user space
C. It encrypts files before sending them over the network
D. It splits large files into smaller chunks for parallel transfer

Answer: B

Q1333. What is the io_uring interface in Linux and why is it significant?

A. It encrypts I/O data using ring signatures
B. It is a circular buffer for logging I/O errors
C. It manages USB device connections in a ring topology
D. It provides a high-performance asynchronous I/O interface using shared ring buffers between user and kernel space

Answer: D

Q1334. What is the write amplification problem in storage I/O?

A. When the actual amount of data written to storage is significantly larger than the logical data being written
B. When multiple programs write to the same file simultaneously
C. When write operations are slower than read operations
D. When data is written in the wrong encoding format

Answer: A

Q1335. How does memory-mapped I/O differ from port-mapped I/O at the hardware level?

A. Port-mapped I/O uses the same address space as main memory
B. Memory-mapped I/O uses the same address space for memory and I/O devices while port-mapped I/O uses a separate address space
C. Memory-mapped I/O is slower in all cases
D. Memory-mapped I/O is only available on ARM processors

Answer: B

Q1336. What is the purpose of the O_DIRECT flag when opening files on Linux?

A. It opens the file in a special directory
B. It encrypts all data written to the file
C. It bypasses the operating system page cache and performs direct I/O between user buffers and the storage device
D. It directs output to the console instead of the file

Answer: C

Q1337. What is the difference between level-triggered and edge-triggered I/O notification?

A. Level-triggered reports readiness as long as the condition holds while edge-triggered reports only when the state changes
B. Edge-triggered is always more efficient than level-triggered
C. Level-triggered only works with read operations
D. Edge-triggered requires blocking I/O

Answer: A

Q1338. What is the purpose of vectored I/O (scatter-gather) and when is it beneficial?

A. It reads or writes data from multiple non-contiguous buffers in a single system call, reducing overhead
B. It gathers I/O statistics from multiple processes
C. It splits a single buffer into multiple files
D. It encrypts data by scattering bits across storage

Answer: A

Q1339. What is the durability guarantee provided by fsync versus fdatasync?

A. fsync flushes both file data and metadata to disk while fdatasync flushes only the file data and essential metadata
B. They provide identical durability guarantees in all cases
C. fdatasync provides stronger guarantees than fsync
D. fsync only flushes metadata while fdatasync flushes data

Answer: A

Q1340. What is the thundering herd problem in the context of I/O multiplexing with accept()?

A. When multiple threads waiting on the same listening socket all wake up for a single connection, but only one succeeds
B. When too many connections arrive simultaneously and crash the server
C. When I/O buffers overflow due to rapid data arrival
D. When multiple files are opened simultaneously causing disk contention

Answer: A

Q1341. What is the Aho-Corasick algorithm and when is it used for string matching?

A. It searches for multiple patterns simultaneously in a text by building an automaton from all patterns
B. It sorts strings in alphabetical order efficiently
C. It matches a single pattern against multiple texts
D. It compresses strings using pattern-based encoding

Answer: A

Q1342. What is the suffix array data structure and how does it improve string operations?

A. A linked list of all string prefixes
B. An array that stores only the last character of each string
C. A hash table mapping string suffixes to their positions
D. A sorted array of all suffixes of a string that enables efficient substring search, longest common prefix computation, and other operations

Answer: D

Q1343. How does the Burrows-Wheeler Transform improve text compression?

A. It rearranges characters so that similar characters are grouped together, making the result more compressible
B. It removes duplicate characters from the text
C. It encrypts the text to reduce its size
D. It replaces all characters with shorter binary codes

Answer: A

Q1344. What is the difference between greedy and lazy quantifiers in regular expressions?

A. Greedy quantifiers match as much text as possible while lazy quantifiers match as little as possible
B. Lazy quantifiers match more text than greedy quantifiers
C. They produce identical matches in all cases
D. Greedy quantifiers are faster than lazy quantifiers

Answer: A

Q1345. What is catastrophic backtracking in regular expressions?

A. When a regex engine takes exponential time due to ambiguous patterns that cause excessive backtracking
B. When backtracking causes the program to run out of memory
C. When a regex accidentally matches the wrong string
D. When a regex fails to match any string

Answer: A

Q1346. What is the Z-algorithm for string matching?

A. It constructs a Z-array where Z[i] is the length of the longest substring starting at position i that matches a prefix of the string
B. It finds the last occurrence of a pattern in a string
C. It sorts characters in a string alphabetically
D. It compresses strings using a zigzag encoding

Answer: A

Q1347. How do persistent string data structures maintain previous versions efficiently?

A. By compressing all previous versions into a single archive
B. By keeping complete copies of every version in memory
C. By storing only the most recent version and discarding older ones
D. By sharing unchanged portions between versions and only allocating new memory for changed parts

Answer: D

Q1348. What is Unicode normalization and why can two visually identical strings have different byte representations?

A. Unicode only affects non-English text
B. Different byte representations indicate file corruption
C. Normalization removes all special characters from a string
D. Characters can be represented as single code points or as combining character sequences, and normalization converts between these forms

Answer: D

Q1349. What is the difference between DFA and NFA-based regex engines?

A. DFA engines are always slower than NFA engines
B. DFA engines guarantee linear time matching but cannot support backreferences while NFA engines support backreferences but may have exponential worst case
C. NFA engines cannot match basic patterns
D. DFA engines support all regex features including backreferences

Answer: B

Q1350. What is the longest common subsequence (LCS) problem and its string application?

A. Determining which string is longer
B. Finding the longest palindrome within a string
C. Finding the longest sequence of characters common to two strings in the same order but not necessarily contiguous, used in diff algorithms
D. Finding the longest substring that appears in both strings at the same position

Answer: C

Q1351. What is a cuckoo hash table and how does it achieve O(1) worst-case lookups?

A. It stores elements in sorted order for binary search
B. It uses a single hash function with no collisions
C. It uses two hash functions and two tables, displacing existing elements on collision, guaranteeing each key is in one of two possible locations
D. It uses three hash functions and three tables

Answer: C

Q1352. What is the difference between an AVL tree and a red-black tree in terms of balancing?

A. They maintain identical balance guarantees
B. AVL trees maintain stricter balance with at most 1 height difference, while red-black trees allow up to 2x height difference, making AVL faster for lookups but slower for insertions
C. AVL trees do not support deletion operations
D. Red-black trees are always perfectly balanced

Answer: B

Q1353. What is a Fibonacci heap and why is it used in graph algorithms?

A. A heap with exactly 5 children per node
B. A heap used only for sorting Fibonacci sequences
C. A heap that stores Fibonacci numbers
D. A heap with amortized O(1) decrease-key operation, making it optimal for algorithms like Dijkstra's shortest path

Answer: D

Q1354. What is the difference between an R-tree and a B-tree for indexing?

A. B-trees can index spatial data better than R-trees
B. R-trees index multi-dimensional spatial data using bounding rectangles while B-trees index one-dimensional ordered data
C. R-trees are used for text search while B-trees are for spatial data
D. They provide identical functionality for all data types

Answer: B

Q1355. How does a count-min sketch provide approximate frequency counts with bounded error?

A. It uses multiple hash functions mapping to a matrix of counters, providing an upper bound estimate with configurable error probability
B. It stores only the most frequent elements exactly
C. It counts exactly by using unlimited memory
D. It compresses all data into a single counter

Answer: A

Q1356. What is a treap and how does it combine properties of trees and heaps?

A. A special array that combines tree and heap indices
B. A heap that supports tree traversal operations
C. A tree that stores elements in heap order only
D. A BST where each node has a random priority, maintaining BST property on keys and heap property on priorities for probabilistic balance

Answer: D

Q1357. What is the difference between LSM trees and B-trees for write-heavy workloads?

A. LSM trees do not support read operations
B. They perform identically for all workload types
C. B-trees are always faster than LSM trees for writes
D. LSM trees buffer writes in memory and merge sorted runs sequentially, optimizing write throughput, while B-trees update pages in place with random I/O

Answer: D

Q1358. What is a HAT-trie and why is it more cache-friendly than a standard trie?

A. It stores multiple characters per node in contiguous arrays, reducing pointer chasing and improving CPU cache utilization
B. It stores the entire trie in a single array
C. It hashes all keys before insertion into the trie
D. It uses hardware acceleration for trie operations

Answer: A

Q1359. What is the difference between wait-free and lock-free concurrent data structures?

A. They provide identical progress guarantees
B. Wait-free structures are slower than lock-free structures always
C. Lock-free guarantees system-wide progress while wait-free guarantees per-thread progress in bounded steps
D. Lock-free structures use locks internally while wait-free does not

Answer: C

Q1360. What is a CRDT (Conflict-free Replicated Data Type) and why is it important for distributed collections?

A. A compressed data type for reducing network bandwidth
B. A data type that prevents all conflicts by using locks
C. A data type designed so that concurrent updates on different replicas always converge to the same state without coordination
D. A data type that replicates data by copying entire collections

Answer: C

Q1361. How does the bridge pattern separate abstraction from implementation?

A. By using two separate inheritance hierarchies connected by composition, allowing them to vary independently
B. By merging all implementations into a single abstract class
C. By using global variables to connect unrelated classes
D. By creating a bridge between two interfaces using static methods

Answer: A

Q1362. What is the Liskov Substitution Principle's impact on exception handling in subclasses?

A. Subclasses must throw the exact same exceptions as the superclass
B. Subclasses should never throw any exceptions
C. Subclass methods should not throw broader exceptions than the superclass method specifies to maintain substitutability
D. The LSP does not apply to exception handling

Answer: C

Q1363. What is the difference between the state pattern and the strategy pattern?

A. The state pattern is for concurrency while the strategy pattern is for sorting
B. They are identical patterns with different names
C. The state pattern encapsulates state-dependent behavior with transitions between states while the strategy pattern selects an algorithm without transitions
D. The strategy pattern manages state transitions while the state pattern selects algorithms

Answer: C

Q1364. What is the object pool pattern and when is it beneficial?

A. A pattern that distributes objects across multiple servers
B. A pattern that creates objects in a swimming pool metaphor
C. A pattern that pools all objects into a single collection for easy access
D. A pattern that maintains a pool of reusable objects to avoid expensive creation and destruction costs

Answer: D

Q1365. What is the problem with inheritance-based reuse known as the 'yo-yo problem'?

A. When multiple inheritance creates an infinite loop
B. When understanding a class requires tracing through many levels of inheritance, bouncing up and down the hierarchy
C. When objects repeatedly create and destroy themselves
D. When a class inherits from both a parent and grandchild class

Answer: B

Q1366. How does the specification pattern encapsulate business rules as recombinable objects?

A. By storing rules as string expressions evaluated at runtime
B. By hardcoding rules into if-else chains
C. By writing all business rules as database queries
D. By representing each business rule as an object with an isSatisfiedBy method that can be combined using AND, OR, and NOT operations

Answer: D

Q1367. What is the difference between type classes in Haskell and interfaces in Java?

A. Type classes are slower than interfaces
B. They are identical concepts in different languages
C. Java interfaces support more features than type classes
D. Type classes can add behavior to existing types without modifying them while Java interfaces require implementation at class definition time

Answer: D

Q1368. What is the memento pattern and how does it support undo functionality?

A. It stores all objects in persistent memory
B. It creates memory snapshots of the entire application
C. It memorizes method call history in a static log
D. It captures an object's internal state in a separate memento object that can be used to restore the state later

Answer: D

Q1369. How does the entity-component-system (ECS) architecture differ from traditional OOP?

A. ECS requires all entities to share the same base class
B. ECS is just another name for the MVC pattern
C. ECS uses deep inheritance hierarchies for entity behavior
D. ECS separates data (components) from behavior (systems) and uses composition over inheritance, favoring cache-friendly memory layouts

Answer: D

Q1370. What is the expression problem and how do OOP and functional programming approach it differently?

A. OOP cannot handle any expressions while FP handles all expressions
B. Both paradigms solve the expression problem identically
C. FP cannot define new types while OOP cannot define new functions
D. OOP easily adds new types but struggles to add new operations, while FP easily adds new operations but struggles to add new types

Answer: D

Q1371. What is the tri-color marking algorithm in garbage collection?

A. It colors memory pages for security purposes
B. It marks objects based on their data type using three categories
C. It uses three passes to determine which objects to collect
D. It classifies objects as white (unreachable), gray (to be scanned), and black (scanned) to safely identify garbage during concurrent collection

Answer: D

Q1372. How does the Boehm-Demers-Weiser conservative garbage collector work for C programs?

A. It requires the programmer to annotate all pointers
B. It only collects objects on the stack
C. It converts C programs to Java for garbage collection
D. It treats any value in memory that could be a pointer as a potential reference, preventing collection of possibly-referenced objects

Answer: D

Q1373. What is the difference between stop-the-world and concurrent garbage collection?

A. Concurrent GC always takes longer than stop-the-world
B. Stop-the-world only pauses one thread at a time
C. They produce identical pause times in all scenarios
D. Stop-the-world pauses all application threads during collection while concurrent GC runs alongside the application with minimal pauses

Answer: D

Q1374. What is the ZGC garbage collector in Java designed to achieve?

A. Maximum throughput at the expense of longer pauses
B. Sub-millisecond pause times regardless of heap size by using colored pointers and load barriers
C. Smaller memory footprint than serial GC
D. Compatibility with C code through JNI

Answer: B

Q1375. What is the purpose of Rust's borrow checker?

A. It enforces rules at compile time ensuring references do not outlive the data they point to and preventing data races
B. It monitors runtime memory usage for debugging
C. It automatically frees memory when borrowing ends
D. It checks if the program borrows external libraries correctly

Answer: A

Q1376. What is the difference between arena allocation and general-purpose heap allocation?

A. Arena allocation is slower than heap allocation for all use cases
B. Arena allocation requires garbage collection to function
C. Arena allocation groups objects with similar lifetimes and frees them all at once, avoiding individual deallocation overhead
D. General-purpose allocation always uses less memory than arena allocation

Answer: C

Q1377. What is the Cheney's algorithm for copying garbage collection?

A. It copies garbage to a separate memory region for analysis
B. It creates backup copies of all objects for safety
C. It uses three memory spaces in a rotating pattern
D. It uses two equal-sized semi-spaces, copying live objects from one to the other, compacting memory and eliminating fragmentation

Answer: D

Q1378. What is the problem of false sharing in concurrent programs and how is it mitigated?

A. When multiple threads allocate memory from the same pool
B. When different threads modify variables on the same cache line, causing unnecessary cache invalidation and performance degradation
C. When shared memory is falsely reported as freed
D. When threads share incorrect data due to synchronization bugs

Answer: B

Q1379. What is the Shenandoah GC's approach to concurrent compaction?

A. It compacts memory only during application idle time
B. It uses stop-the-world pauses for all compaction operations
C. It relies on the operating system for compaction
D. It uses Brooks forwarding pointers to redirect references to relocated objects, enabling compaction concurrent with the application

Answer: D

Q1380. What is the concept of escape analysis and how does it optimize memory allocation?

A. It analyzes memory escape routes during buffer overflow attacks
B. It determines when variables escape from their declared scope
C. It detects when objects escape the program's memory space
D. It analyzes whether an object escapes the method or thread that created it, enabling stack allocation or elimination for non-escaping objects

Answer: D

Q1381. What is the difference between the Result type in Rust and traditional exception handling?

A. Result encodes success or failure in the return type, forcing callers to handle errors explicitly at compile time, while exceptions propagate implicitly
B. Result types can only represent two types of errors
C. Exceptions provide better type safety than Result
D. Result types are slower than exception handling

Answer: A

Q1382. What is the concept of exception safety in C++ and its three guarantee levels?

A. Low, medium, and high exception severity levels
B. Compile-time safety, runtime safety, and thread safety
C. Type safety, memory safety, and null safety
D. Basic guarantee (no leaks), strong guarantee (transaction-like rollback), and no-throw guarantee (never throws exceptions)

Answer: D

Q1383. How does Kotlin's approach to checked exceptions differ from Java's?

A. Kotlin uses a different keyword for checked exceptions
B. Kotlin requires all exceptions to be checked
C. Kotlin only allows checked exceptions in coroutines
D. Kotlin does not have checked exceptions, treating all exceptions as unchecked, because studies showed checked exceptions often lead to empty catch blocks

Answer: D

Q1384. What is the relationship between exceptions and monadic error handling using Either/Result?

A. Monadic error handling replaces all try-catch blocks with recursion
B. Exceptions are a subset of monadic error handling
C. Monadic error handling cannot represent multiple error types
D. Both represent computations that may fail, but monadic error handling makes failure explicit in types and composable through flatMap/bind

Answer: D

Q1385. What is the zero-cost exception handling model used in modern C++ compilers?

A. It eliminates all exception-related code at compile time
B. Exceptions are completely free even when thrown
C. It replaces exceptions with return codes automatically
D. It adds no performance overhead when exceptions are not thrown, using table-based stack unwinding only when exceptions occur

Answer: D

Q1386. What is the problem with exceptions in asynchronous or concurrent code?

A. Exceptions cannot cross thread or async boundaries naturally and may be lost if not properly captured and propagated
B. Concurrent code cannot throw any exceptions
C. Exceptions are always thread-safe and work identically in concurrent code
D. Exceptions in threads automatically propagate to the main thread

Answer: A

Q1387. How does structured concurrency improve exception handling compared to unstructured async code?

A. It handles exceptions by restarting failed tasks indefinitely
B. It ensures that all concurrent tasks are scoped to a parent, and exceptions from child tasks automatically cancel siblings and propagate to the parent
C. It defers all exceptions until the program exits
D. It eliminates all exceptions in concurrent code

Answer: B

Q1388. What is the effect handler approach to exception handling and how does it generalize try-catch?

A. Effect handlers define how effects (including exceptions) are interpreted, allowing the same code to have different error handling strategies depending on the handler
B. They add hardware-level exception handling support
C. Effect handlers only work in statically typed languages
D. Effect handlers replace all exception types with a single error type

Answer: A

Q1389. What is the supervisor pattern in actor systems for handling failures?

A. A thread monitor that prevents all exceptions from occurring
B. A design pattern that replaces all try-catch with logging
C. A parent actor monitors child actors and applies a restart, stop, or escalate strategy when a child fails with an exception
D. A senior developer supervises code reviews for exception handling

Answer: C

Q1390. What is the module resolution algorithm in Node.js and how does it find modules?

A. It uses a random search through all directories
B. It only searches the current working directory
C. It always looks in the global installation directory first
D. It searches node_modules directories ascending from the requiring file's location, then global locations, checking for files, directories with index.js, or package.json main fields

Answer: D

Q1391. What is the purpose of import maps in modern JavaScript?

A. They provide a map visualization of module dependencies
B. They create documentation maps for modules
C. They allow controlling how module specifiers are resolved in the browser without a build step
D. They map imports to different programming languages

Answer: C

Q1392. What is the difference between side-effect-free and side-effectful module imports?

A. Side-effect-free imports are slower
B. They produce identical behavior in all bundlers
C. Side-effectful imports cannot export any values
D. Side-effect-free imports only use the exported values while side-effectful imports execute code that modifies global state on import

Answer: D

Q1393. What is the challenge of diamond dependency conflicts in package management?

A. When a package depends on itself recursively
B. When two packages have identical names in different registries
C. When a dependency graph forms a perfect diamond shape
D. When two dependencies require different incompatible versions of the same transitive dependency

Answer: D

Q1394. How does the Java Platform Module System (JPMS) differ from traditional classpath-based loading?

A. Classpath provides stronger encapsulation than JPMS
B. JPMS uses module declarations to explicitly define dependencies and encapsulated packages, unlike the flat classpath that exposes everything
C. JPMS is slower than classpath loading
D. JPMS only works with Java 7 and earlier

Answer: B

Q1395. What is supply chain security in the context of package management?

A. Protecting against malicious code injected into dependencies through compromised packages or maintainer accounts
B. Managing the physical supply chain of hardware components
C. Ensuring packages are delivered in the correct order
D. Securing the network connection used to download packages

Answer: A

Q1396. What is the purpose of a Software Bill of Materials (SBOM) for packages?

A. It describes the hardware requirements for the software
B. It lists the financial cost of each dependency
C. It provides a comprehensive inventory of all components, libraries, and dependencies in a software project for security and compliance
D. It documents the development timeline of each package

Answer: C

Q1397. How does Deno's module system differ from Node.js in terms of dependency management?

A. Deno uses the same npm and node_modules system as Node.js
B. Deno requires all dependencies to be bundled before execution
C. Deno does not support external modules
D. Deno imports modules directly from URLs without a centralized package manager or node_modules directory

Answer: D

Q1398. What is the purpose of module federation in webpack 5?

A. Allowing multiple independently deployed applications to share modules at runtime without rebuilding
B. Federating modules across different government agencies
C. Merging all modules into a single federated package
D. Creating federal backups of all modules

Answer: A

Q1399. What is the difference between vendoring and using a package manager for dependencies?

A. Vendoring copies dependency source code directly into the project repository while package managers download them on demand
B. Package managers copy all code into the repository
C. Vendoring only works with interpreted languages
D. Vendoring is always faster than using a package manager

Answer: A

Q1400. What is the difference between the Abstract Factory and Factory Method patterns?

A. Abstract Factory is a simpler version of Factory Method
B. Abstract Factory creates families of related objects through an interface while Factory Method delegates creation of a single object to subclasses
C. They are identical patterns used in different programming languages
D. Factory Method creates multiple objects while Abstract Factory creates one

Answer: B

Q1401. How does the CQRS pattern improve scalability in distributed systems?

A. By combining all queries and commands into a single optimized model
B. By caching all queries on the client side
C. By separating read and write models, allowing each to be scaled, optimized, and evolved independently
D. By reducing the number of database tables

Answer: C

Q1402. What is the difference between compile-time and runtime metaprogramming?

A. Compile-time metaprogramming is always slower
B. They both execute at the same phase
C. Compile-time metaprogramming generates or transforms code during compilation while runtime metaprogramming modifies program structure during execution
D. Runtime metaprogramming cannot create new functions

Answer: C

Q1403. What is the difference between event sourcing and traditional state storage?

A. Event sourcing stores every state change as an immutable event, deriving current state by replaying events, while traditional storage saves only current state
B. Traditional storage keeps complete history while event sourcing does not
C. They store data in identical formats
D. Event sourcing only stores the final state

Answer: A

Q1404. How does the saga pattern manage distributed transactions across microservices?

A. It retries all failed transactions indefinitely
B. It uses a single distributed lock across all services
C. It orchestrates a sequence of local transactions with compensating actions to undo completed steps if a later step fails
D. It rolls back all services using two-phase commit

Answer: C

Q1405. What is the difference between the interpreter pattern and an actual language interpreter?

A. The interpreter pattern compiles code while a language interpreter interprets it
B. The interpreter pattern defines a grammar representation and an interpreter for sentences in that grammar, used for simple DSLs, while a language interpreter is a full program execution engine
C. A language interpreter is simpler than the interpreter pattern
D. They are identical in scope and implementation

Answer: B

Q1406. What is the purpose of the circuit breaker pattern in distributed systems?

A. It manages electrical circuits in server hardware
B. It circuits all requests through a central router
C. It breaks circular dependencies between services
D. It prevents cascading failures by stopping requests to a failing service and allowing it time to recover

Answer: D

Q1407. What is the difference between a red-black tree and a 2-3-4 tree?

A. A red-black tree is a binary representation of a 2-3-4 tree where red links represent nodes merged within 2-3-4 tree nodes
B. Red-black trees have more nodes than 2-3-4 trees for the same data
C. They are completely unrelated data structures
D. 2-3-4 trees are always faster for all operations

Answer: A

Q1408. What is backpressure in reactive programming and why is it important?

A. Pressure applied to undo the last operation
B. Network congestion that slows down all requests
C. Memory pressure from running out of heap space
D. A mechanism that allows consumers to signal producers to slow down when they cannot keep up with the data flow rate

Answer: D

Q1409. What is the difference between optimistic and pessimistic concurrency control in application design?

A. They are identical strategies with different names
B. Pessimistic control never uses any locks
C. Optimistic control assumes conflicts are rare and detects them at commit time, while pessimistic control prevents conflicts by acquiring locks before access
D. Optimistic control is always faster than pessimistic control

Answer: C

Q1410. What is the difference between lock-free and wait-free algorithms?

A. Lock-free algorithms are always faster than wait-free
B. Lock-free guarantees system-wide progress while wait-free guarantees bounded completion for every thread
C. They provide identical progress guarantees
D. Wait-free algorithms use locks while lock-free algorithms do not

Answer: B

Q1411. What is the epoch-based reclamation (EBR) technique for concurrent memory management?

A. It reclaims memory at fixed time intervals
B. It creates backup copies of memory at each epoch
C. It uses epoch timestamps to sort memory allocations
D. It divides time into epochs and defers memory reclamation until all threads have advanced past the epoch when the memory was freed

Answer: D

Q1412. How does the Disruptor pattern achieve high-performance inter-thread communication?

A. It disrupts thread scheduling to gain priority
B. It uses a lock-free ring buffer with sequencing to enable producer-consumer communication with mechanical sympathy for CPU caches
C. It uses multiple locks for each data element
D. It stores all data in a shared database

Answer: B

Q1413. What is the difference between linearizability and serializability?

A. They are identical consistency models
B. Linearizability requires operations to appear atomic at a single point in real time while serializability requires the result to be equivalent to some serial execution
C. Serializability is stronger than linearizability
D. Linearizability only applies to single-threaded programs

Answer: B

Q1414. What is the purpose of hazard pointers in lock-free data structures?

A. They mark memory locations that are unsafe to write to
B. They protect nodes from being reclaimed while other threads still reference them, solving the safe memory reclamation problem
C. They point to hazardous code sections that should not be executed
D. They point to threads that may cause deadlocks

Answer: B

Q1415. What is the difference between strong and weak memory ordering on modern CPUs?

A. Strong ordering prevents all concurrency bugs automatically
B. Weak ordering is always faster than strong ordering
C. Strong ordering guarantees operations appear in program order to all threads while weak ordering allows reordering for performance, requiring explicit barriers
D. They only affect floating-point operations

Answer: C

Q1416. What is structured concurrency and why is it considered safer than unstructured thread creation?

A. It requires all threads to run the same structured code
B. It only allows creating threads in specific code structures
C. It ensures concurrent tasks are scoped to a defined lifetime, guaranteeing all tasks complete or are cancelled before the scope exits
D. It structures threads in a hierarchical tree permanently

Answer: C

Q1417. What is the consensus problem in distributed concurrent systems?

A. Agreeing on the number of threads to allocate
B. Getting multiple processes to agree on a single value despite failures, foundational for distributed coordination
C. Getting all stakeholders to agree on the software architecture
D. Reaching consensus on which programming language to use

Answer: B

Q1418. What is the FLP impossibility result and its significance for concurrent systems?

A. It demonstrates that threads cannot share memory safely
B. It shows that lock-free algorithms are impossible
C. It proves that deterministic consensus is impossible in an asynchronous system where even one process can crash
D. It proves that all concurrent programs eventually deadlock

Answer: C

Q1419. What is the difference between virtual threads (Project Loom) and platform threads in Java?

A. Virtual threads cannot perform I/O operations
B. Platform threads use less memory than virtual threads
C. Virtual threads are lightweight user-mode threads managed by the JVM that can be created in millions, while platform threads are OS threads with higher overhead
D. Virtual threads are faster for all workloads

Answer: C

Q1420. What is concolic testing and how does it combine concrete and symbolic execution?

A. It tests concurrent code with symbolic debuggers
B. It uses concrete test data and symbolic names for variables
C. It runs concrete execution alongside symbolic execution, using concrete values to guide path exploration and symbolic constraints for input generation
D. It combines console output testing with logic testing

Answer: C

Q1421. What is the difference between deterministic and non-deterministic testing of concurrent programs?

A. They produce identical results for concurrent code
B. Deterministic testing controls thread scheduling to explore specific interleavings while non-deterministic relies on random scheduling
C. Non-deterministic testing is more thorough than deterministic
D. Deterministic tests always pass while non-deterministic may fail

Answer: B

Q1422. What is metamorphic testing and when is it useful?

A. Testing where outputs of related test inputs are compared using known relationships when expected outputs are hard to determine
B. Testing the metadata of test files
C. Testing morphological transformations of data structures
D. Testing that code changes (morphs) correctly across versions

Answer: A

Q1423. How does reverse debugging (time-travel debugging) work?

A. It debugs code before it is compiled
B. It reverses the order of test execution
C. It runs the program in reverse to undo changes
D. It records the entire program execution state, allowing developers to step backward through execution to find the root cause of bugs

Answer: D

Q1424. What is the difference between generation-based and mutation-based fuzzing?

A. Generation-based fuzzing creates inputs from grammar specifications while mutation-based fuzzing modifies existing valid inputs
B. They produce identical test coverage
C. Mutation-based fuzzing requires manual test case writing
D. Generation-based fuzzing is always faster

Answer: A

Q1425. What is the purpose of formal verification compared to testing?

A. Formal verification is faster than testing
B. Testing provides stronger guarantees than formal verification
C. Formal verification replaces the need for all testing
D. Formal verification mathematically proves correctness for all possible inputs while testing can only check a finite number of cases

Answer: D

Q1426. What is coverage-guided fuzzing and how does it improve upon random fuzzing?

A. It focuses only on covering previously tested paths
B. It uses code coverage feedback to guide input mutations toward unexplored code paths, achieving deeper code coverage
C. It guides tests toward code that already has high coverage
D. It reduces coverage to speed up fuzzing

Answer: B

Q1427. What is the oracle problem in software testing?

A. The difficulty of choosing the right testing framework
B. A problem with Oracle database testing
C. The challenge of determining the correct expected output for a given test input, especially for complex systems
D. A problem with predicting which tests will fail

Answer: C

Q1428. What is the difference between white-box fuzzing and black-box fuzzing?

A. White-box fuzzing only tests visible code while black-box tests hidden code
B. They explore the same paths using different input formats
C. White-box fuzzing uses program analysis to systematically generate inputs that explore specific paths while black-box fuzzing treats the program as opaque
D. Black-box fuzzing requires source code access while white-box does not

Answer: C

Q1429. What is the purpose of differential testing (differential fuzzing)?

A. Testing the difference between old and new code versions only
B. Testing the performance difference between two algorithms
C. Testing equivalent implementations of the same specification against each other to find discrepancies indicating bugs
D. Testing differences between development and production environments

Answer: C

Q1430. What is the purpose of chaos engineering and how does Netflix's Chaos Monkey work?

A. Chaos Monkey introduces syntax errors into code
B. It randomly assigns developers to different teams
C. Chaos Monkey randomly terminates production instances to verify that the system can tolerate failures, building confidence in resilience
D. It chaotically reorders database records for testing

Answer: C

Q1431. What is the difference between observability and monitoring?

A. They are identical practices with different names
B. Monitoring tracks predefined metrics and alerts while observability enables exploring and understanding system behavior from its outputs without predefined queries
C. Monitoring provides more insight than observability
D. Observability is simpler than monitoring

Answer: B

Q1432. How does GitOps extend DevOps practices using Git as the single source of truth?

A. GitOps uses Git repositories as the declarative source of truth for infrastructure and application configuration, with automated reconciliation
B. GitOps eliminates the need for CI/CD pipelines
C. GitOps only applies to frontend development
D. GitOps replaces Git with a new version control system

Answer: A

Q1433. What is the purpose of a service mesh and how does it relate to microservice development?

A. It creates a mesh network between physical servers
B. It provides a dedicated infrastructure layer handling service-to-service communication including load balancing, encryption, and observability without changing application code
C. It provides networking hardware for services
D. It meshes all services into a single application

Answer: B

Q1434. What is the concept of developer experience (DX) and why does it matter?

A. DX focuses on making tools, APIs, documentation, and workflows efficient and pleasant for developers, directly impacting productivity and code quality
B. DX only applies to designing developer-facing UI
C. DX measures how many years of experience a developer has
D. DX is about decorating the developer workspace

Answer: A

Q1435. What is the difference between Platform Engineering and traditional DevOps?

A. Platform Engineering builds internal developer platforms as a product to enable self-service capabilities, while DevOps focuses on culture and practices for collaboration
B. DevOps provides self-service while Platform Engineering requires manual ops
C. They are identical approaches to infrastructure management
D. Platform Engineering replaces DevOps entirely

Answer: A

Q1436. What is the DORA metrics framework for measuring software delivery performance?

A. It measures developer satisfaction and code quality only
B. It measures server uptime and network latency
C. It measures the number of lines of code written per day
D. It measures deployment frequency, lead time for changes, change failure rate, and time to restore service

Answer: D

Q1437. What is the purpose of dependency injection containers and how do they implement inversion of control?

A. They invert the direction of data flow in the application
B. They contain all project dependencies in a single file
C. They inject security vulnerabilities for penetration testing
D. They manage object creation and lifecycle, automatically resolving and injecting dependencies based on configuration, inverting control of dependency creation from application code to the container

Answer: D

Q1438. What is the concept of immutable infrastructure?

A. Infrastructure that cannot be deleted once created
B. Infrastructure that uses immutable data structures only
C. Infrastructure where servers are never modified after deployment; changes require building and deploying entirely new instances
D. Infrastructure that never changes its IP addresses

Answer: C

Q1439. What is the difference between vertical and horizontal scaling and when is each appropriate?

A. Horizontal scaling requires a single powerful server
B. Vertical scaling is always better than horizontal
C. Vertical scaling adds more resources to a single machine while horizontal scaling adds more machines, with horizontal being better for distributed workloads and vertical for simpler architectures
D. They achieve identical results in all scenarios

Answer: C

Q1440. How does the error kernel pattern improve fault tolerance in distributed systems?

A. It creates a kernel-level exception handler for all processes
B. It prevents all errors by running code in a secure kernel
C. It logs all errors to a central kernel database
D. It isolates error-prone code in separate processes supervised by a minimal, highly reliable kernel that handles restarts

Answer: D