



Codethon

Basic Java Questions

1. Hello World Program

- Write a Java program that prints "Hello, World!" to the console.
- Write a Java program to swap 2 numbers without using third variable.

2. Data Types and Variables

- Write a program that declares variables of different data types (int, float, char, boolean) and prints their values.

3. Control Structures

- Write a Java program that checks if a number is even or odd using an if-else statement.
- Write a Java program to check the divisibility of entered number with 8 and 5 using else if ladder.

4. Switch Statement

- Create a program that takes a day number (1-7) as input and prints the corresponding day of the week.

5. Loops

- Write a program that prints the first 10 natural numbers using a for loop.
- Write a program to print hollow pyramid with * symbol.

6. While Loop

- Write a program that takes a positive integer input from the user and calculates the sum of all integers from 1 to that number using a while loop.

7. Arrays

- Write a program that initializes an array of integers and finds the largest number in the array.

8. String Manipulation



- Write a program that takes a string input from the user and prints the string in reverse.

9. Methods

- Create a method that takes two integers as parameters and returns their sum. Write a main method to test it.

10. Overloading Methods

- Write two overloaded methods named multiply that can take either two integers or two double values and return the product.

11. Inheritance Basics

- Define a class Animal with a method sound(). Create a subclass Dog that overrides the sound() method to print "Woof".

12. Constructors in Inheritance

- Create a superclass Vehicle with a constructor that takes a String parameter. Create a subclass Car that calls the superclass constructor.

13. Is-a Relationship

- Create a class hierarchy for a simple animal kingdom. Define a base class Animal and derive classes like Dog, Cat, and Bird. Demonstrate polymorphism by creating a method in Animal called speak() that is overridden in each derived class.

14. Has-a Relationship

- Design a Library class that has a collection of Book objects. Each Book should have attributes like title, author, and ISBN. Implement methods to add, remove, and list books in the library.

15. Association

- Implement a Student class and a Course class where students can enroll in multiple courses, and each course can have multiple students. Illustrate how you would model this association.

16. Aggregation

- Create a Department class that aggregates Employee objects. Each department should have a name and a list of employees. Show how employees can exist independently of the department they belong to.



17. Composition

- Design a Car class that has a composition relationship with Engine. An Engine cannot exist without a Car. Define methods to start and stop the car, demonstrating how the engine is started and stopped.

18. Typecasting

- Create a base class Shape and derived classes Circle and Rectangle. Write a method that takes a Shape reference and uses typecasting to determine the specific type of shape at runtime. Demonstrate this with a simple example.

19. Real-World Scenario

- Imagine a school system with Person, Teacher, and Student classes. Show how you would implement these classes to demonstrate both inheritance (is-a) and association (has-a), such as Teacher has-a Subject.

20. Composite Design Pattern

- Implement the Composite design pattern using a File and Folder class. A Folder can contain both files and other folders. Create methods to