Java Theory Questions and Programming Concepts (BCA Exam)

Java Theory Questions (Most Probable for Exam)

1. Explain Object-Oriented Programming principles with examples (Inheritance, Encapsulation, Polymorphism, Abstraction).

2. Describe the difference between method overloading and method overriding.

3. Explain the use of 'this' and 'super' keywords with examples.

4. What are constructors in Java? Explain types with examples.

5. Explain the difference between interface and abstract class with real-life examples.

6. What is exception handling? Explain try, catch, throw, throws, and finally with examples.

7. What are packages in Java? How are user-defined packages created?

8. Compare String, StringBuffer, and StringBuilder classes.

9. Explain Java file handling with a program to read/write a text file.

10. What is multithreading in Java? Explain Thread class and Runnable interface.

11. Difference between compile-time and run-time polymorphism with examples.

12. Explain access modifiers (public, private, protected, default) with example programs.

13. Describe the lifecycle of a thread in Java.

14. What is garbage collection in Java? How does it work internally?

Java Theory Questions and Programming Concepts (BCA Exam)

15. Write a program to demonstrate the use of inheritance and method overriding.

Java Programming Concepts (Basic to Advanced)

-> Basic Programs:

- Print a message, find the largest number, check even/odd.

- Calculate factorial, reverse a number, sum of digits.

-> OOP Programs:

- Class & object example, constructor overloading.

- Single and multilevel inheritance, method overriding.

- Abstraction using abstract class and interface.

-> Exception Handling:

- Handle divide by zero using try-catch.

- Use of throw and throws keywords.

-> File Handling:

- Read from and write to a file using FileReader/FileWriter.

-> Arrays & Strings:

- Program on 2D and jagged arrays.

Java Theory Questions and Programming Concepts (BCA Exam)

- Palindrome string, count vowels, string reversal.

-> Multithreading:

- Create threads using Thread and Runnable class.

- Synchronization example.