

1. Explain any five OOP concepts.
2. Elaborate on JRE, JDK, JVM, JAVA APPLET, JIT COMPILER
3. What is a parameterized constructor? Is it possible to create an object for a class A using `A obj = new A ()`, if the class contains only a parameterized constructor? Justify
4. Summarize the need of constructors in Java and list the different types of constructors allowed in Java programs
5. Implement compile-time polymorphism in Java to calculate the area of different shapes namely circle and rectangle
6. How is platform independence achieved in Java?
7. Distinguish between Composition and Aggregation relations that exist in class diagram
8. What are the advantages of using UML? Sketch the UML class diagram for an entity 'book'.
9. Why is the 'main' method in Java qualified as public, static, and void?
10. Compare type conversion and type casting, and explain their potential side effects with examples.
11. Write a java program that illustrates how 'this' keyword can be used to resolve the ambiguity between formal parameters and instance variables.
12. Construct a UML Activity diagram for an online Hotel Reservation System, which shows the flow of activities for booking rooms at a hotel.
13. Explain how garbage collection is done in Java.
14. Detail the types of inheritance using diagrams
15. Consider the problem of a Service Station which provides three types of services to its customers: refuelling, vehicle maintenance, and parking. Customer can pay using cash, card or cheque. The pricing for vehicle maintenance depends on the cost of parts and labour. Parking areas are rented according to weekly and monthly rates. Construct a UML class diagram for the above problem by identifying at least six entities in the system which can be represented using classes and show the relationship between them.
16. Explain any five buzzwords using which Java can be characterized.
17. Construct a UML Class diagram for Online Movie Ticket Booking System. The various entities involved in the system are Admin, Registered User, Visitor / Guest User, Movie, Book Ticket, Make Payment
18. Detail out the six basic object-oriented programming concepts
19. Differentiate between the two main approaches of software design.
20. Draw a UML Activity Diagram for Food Ordering System, which shows the flows between the activity of Order, Delivery, Food, Item, Category, and Payment.

21. Demystify static and dynamic polymorphism. Illustrate with sample programs.
22. Write a Java program to reverse digits of a given integer.
23. Explain the buzzwords of Java that define the Java programming language.
24. Write a Java program to find the frequency (count the occurrence) of each element in an integer array.
25. Write a java program to show the significance of method overriding in achieving run time polymorphism. Discuss difference between method overriding and method overloading.
26. Explain inheritance. Illustrate types of inheritance using sample programs.
27. Explain Java precedence rule. Evaluate output of below expression by demonstrating precedence rule.

```
int x = 2;
int result = x++ + x++ * --x / x++ - --x + 3 >> 1 | 2;
System.out.println("result = " + result);
System.out.println("X = " + x);
```

28. Write a Java program by creating a 'Student' class having the following data members: rollNumber, name, mathMarks, phyMarks, chemMarks, and methods getRequiredDetails() – to get required input and displayAverage() – to calculate average marks and display it. In class 'Implement' create an object of the Student class and get the required details from user and display the average marks of that student.
29. illustrates how the 'this' keyword can resolve the ambiguity between formal parameters and instance variables.
30. Provide a recursive program to calculate the factorial of a number and explain how the call mechanism works using the system stack
31. Write a Java program by creating a 'Student' class having the following data members: rollNumber, name, mathMarks, phyMarks, chemMarks and initialize the student details using constructor and using displayAverage() calculate average marks and display it. In class 'Implement' create an object of the Student class and get the required details from user and display the average marks of that student.
32. Describe the role of 'super' keyword in the context of inheritance in Java with appropriate examples.
33. Create a class Person with attributes name, age, address and a method display() to display the details. Create a subclass Student with attributes rollno, mark1, mark2 and mark3. Override display() method and calculate the grade. Create another subclass Faculty with attributes faculty\_id, department, basic\_pay and DA. Override display() method to calculate total salary as (basic\_pay+DA)+70% of (basic\_pay+DA). Create instances of Student and Faculty and display the details.
34. Explain different types of inheritance in Java with suitable examples.
35. You are given an array of student information: ID, FirstName, and CGPA. Your

task is to rearrange them according to their CGPA in decreasing order. If two students have the same CGPA, then arrange them according to their first name in alphabetical order.

**Sample input:**

5  
33 Rumpa 3.68  
85 Ashis 3.85  
56 Samiha 3.75  
19 Samara 3.75  
22 Fahim 3.76

**Sample output:**

Ashis  
Fahim  
Samara  
Samiha  
Rumpa

36. Write a program to find the LCM of two numbers

Sample input:

Number1: 12

Number2: 15

Sample output:

LCM: 60