Q.1: Object-Oriented Programming (OOP) is a fundamental paradigm in Java and many other programming languages. It is a way of designing and organizing code based on the concept of objects, which represent real-world entities or concepts. Java supports several key OOP concepts, which are as follows:

1. Classes and Objects:

- A class is a blueprint or template for creating objects.
- An object is an instance of a class, representing a specific entity or concept.
- Classes define the structure (attributes/fields) and behavior (methods/functions) of objects.

2. Encapsulation:

- Encapsulation refers to the practice of bundling data (attributes) and the methods (functions) that operate on that data into a single unit, known as a class.
- It provides access control mechanisms to restrict the direct access to some of an object's components, protecting the integrity of the data.

3. Inheritance:

- Inheritance is a mechanism that allows a class to inherit properties and behaviors (attributes and methods) from another class (superclass or parent class).
- It promotes code reuse and the creation of a hierarchical structure among classes.

4. Polymorphism:

- Polymorphism enables objects of different classes to be treated as objects of a common superclass.
- It allows methods to be defined in a general way in a superclass and then overridden in subclasses to provide specific implementations.
- Polymorphism helps achieve flexibility and dynamic behavior in code.

5. Abstraction:

- Abstraction involves simplifying complex systems by breaking them into smaller, more manageable parts.
- In Java, abstract classes and interfaces are used to define abstractions, providing a blueprint for concrete subclasses to implement.

6. Method Overloading and Overriding:

- Method overloading allows a class to have multiple methods with the same name but different parameter lists, based on the number or types of parameters.

- Method overriding occurs when a subclass provides a specific implementation of a method that is already defined in its superclass.

7. Association, Aggregation, and Composition:

- These are relationships between classes:

  - Association represents a simple relationship between two classes.

  - Aggregation is a "has-a" relationship where one class contains another class as a part.

  - Composition is a stronger form of aggregation where one class is composed of other classes, and the lifecycle of the composed objects is tightly tied to the container class.

8. Access Modifiers:

- Java provides access modifiers like public, private, protected, and default (package-private) to control the visibility and accessibility of class members.

In Java, OOP concepts help in building modular, maintainable, and extensible software. They promote code reusability, separation of concerns, and a more intuitive representation of real-world entities within the codebase. Java's OOP features make it a popular choice for developing a wide range of applications, from desktop software to web applications and mobile apps.

MULTIPLE CHOICE QUESTIONS:

1. B
2. A
3. B
4. A
5. A
6. C
7. A
8. B
9. C
10. Na
11. “Base ::show() called”

12. “derived ::show() called”
13. Derived class
14. Test class : name
15. Na
16. Na
17. This wont compile
18. X and y values will be printed
19. 5
20. 7
21. 2
22. 2
23. 2
24. 1 2
25. Obj1.a=4 obj1.b=3
    Obj2.a =4 obj.b = 3