**1. OOPS Concepts in Java**

Java is based on Object Oriented Programming Concepts, following are some of the OOPS concepts implemented in java programming.

**1.Abstraction:** Abstraction is the concept of *hiding the internal details and describing things in simple terms*.

For example, a method that adds two integers. The internal processing of the method is hidden from the outer world. There are many ways to achieve abstraction in object-oriented programming’s, such as encapsulation and inheritance.

**2.Encapsulation:** Encapsulation is the *technique used to implement abstraction* in object-oriented programming. Encapsulation is used for *access restriction to class members and methods*. Access modifier keywords are used for encapsulation in object oriented programming.

For example, encapsulation in java is achieved using private, protected and public keywords.

**3.Polymorphism:** Polymorphism is the concept where an *object behaves differently in different situations*. There are two types of polymorphism – *compile time polymorphism and runtime polymorphism*.

NOTE: Runtime polymorphism is implemented when we have “IS-A” relationship between objects. This is also called as method overriding because subclass has to override the superclass method for runtime polymorphism.

**4.Inheritance:** Inheritance is the object oriented programming concept where an *object is based on another object.* Inheritance is the *mechanism of code reuse*. The object that is getting inherited is called superclass and the object that inherits the superclass is called subclass.

**5.Association:** Association is the OOPS concept to define the *relationship between objects. Association defines the multiplicity between objects.*

For example, Teacher and Student objects. There is one to many relationships between a teacher and students. Similarly, a student can have one to many relationships with teacher objects. However, both student and teacher objects are independent of each other

**6.Aggregation:** Aggregation is a s*pecial type of association. In aggregation, objects have their own life cycle but there is an ownership.* Whenever we have “HAS-A” relationship between objects and ownership then it’s a case of aggregation.

**7.Composition**: Composition is a *special case of aggregation*. Composition is a more *restrictive form of aggregation.* When the contained object in “HAS-A” relationship can’t exist on its own, then it’s a case of composition.

For example, House has-a Room. Here room can’t exist without house. Read more about it at Composition in Java. Composition is said to be better than inheritance, read more at Composition Vs Inheritance.

**2.WHAT IS A CLASS AND OBJECT?**

**Object−** Objects *have states and behaviors*. Example: A dog has states - color, name, breed as well as behaviors – wagging the tail, barking, eating. An object is an instance of a class.

**Class−** A class can be defined as a template/blueprint that describes the behavior/state that the object of its type support.

**3.what is a wrapper class?**

Java wrapper classes are the Object representation of eight primitive types in java. All the wrapper classes in java are immutable and final

**4.Can we have multiple classes in single java source file?**

We can’t have more *than one public class in a single java source file*. A single source file *can have multiple classes that are not public*.

5. Source code to Byte code

6. how executable files created

7.Difference between C & Java

8. Primitive Data Types

9. concept of project

**Cloud storage** is a **cloud** computing model that stores data on the Internet through a **cloud** computing provider who manages and operates data **storage** as a service. It's delivered on demand with just-in-time capacity and costs, and eliminates buying and managing your own data **storage** infrastructure.