UNIT -III

Q1. What is a Class in Java ? Discuss.

Ans :

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:

1. **Modifiers** : A class can be public or has default access
2. **Class name:** The name should begin with a initial letter (capitalized by convention).
3. **Superclass(if any):** The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.
4. **Interfaces(if any):** A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
5. **Body:** The class body surrounded by braces, { }.

……………………………………………………………………………………..

Q.2. Define Object in Java.

Ans : OBJECT is an instance of a class. An object is nothing but a self-contained component which consists of methods and properties to make a particular type of data useful. For example color name, table, bag, barking. When we send a message to an object, we are asking the object to invoke or execute one of its methods as defined in the class.

From a programming point of view, an object can include a data structure, a variable, or a function. It has a memory location allocated. The object is designed as class hierarchies.

Syntax

ClassName ReferenceVariable = new ClassName();

An object has three characteristics:

- **State:** represents the data (value) of an object.
- **Behavior:** represents the behavior (functionality) of an object such as deposit, withdraw, etc.
- **Identity:** An object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. However, it is used internally by the JVM to identify each object uniquely.

……………………………………………………..

Q.3 What is the Difference Between Object & Class?

Ans:

A **class** is a **blueprint or prototype** that defines the variables and the methods (functions) common to all objects of a certain kind.

An **object** is a specimen of a class. Software objects are often used to model real-world objects we find in everyday life.

………………………………………………………………………………..

Q.4. Define Constructor in Java.

Ans : Constructor is a block of code that initializes the newly created object. A constructor resembles an instance method in java but it’s not a method as it doesn’t have a return type. In short constructor and method are different. People often refer constructor as special type of method in Java.

Constructor has same name as the class and looks like this in a java code.

```
public class MyClass

{

  //This is the constructor

  MyClass(){

  }

  .}
```

Note that the constructor name matches with the class name and it doesn’t have a return type.

…………………………………………………………..

Q.5. Define different types of Constructor in Java.

Ans : There are two types of constructors in Java:

1. Default constructor (no-arg constructor)
2. Parameterized constructor

Java Default Constructor

A constructor is called "Default Constructor" when it doesn't have any parameter.

Syntax of default constructor:

```
<class_name>()
{
}
```

Example of default constructor

In this example, we are creating the no-arg constructor in the Bike class. It will be invoked at the time of object creation

```
class Bike1
{
Bike1()
{
System.out.println("Bike is created");
}
//main method
public static void main(String args[])
{
//calling a default constructor
Bike1 b=new Bike1();
}
}
```

Java Parameterized Constructor

A constructor which has a specific number of parameters is called a parameterized constructor.

The parameterized constructor is used to provide different values to distinct objects. However, we can provide the same values also.

Example of parameterized constructor

In this example, we have created the constructor of Student class that have two parameters. We can have any number of parameters in the constructor.

```
class Student4
{
    int id;
    String name;
    //creating a parameterized constructor
    Student4(int i,String n)
   {
    id = i;
    name = n;
    }
    //method to display the values
    void display()
{
System.out.println(id+" "+name);
}
public static void main(String args[])
{
    //creating objects and passing values
    Student4 s1 = new Student4(111,"Karan");
    Student4 s2 = new Student4(222,"Aryan");
    //calling method to display the values of object
    s1.display();
    s2.display();
   }

}
```

…………………………………………

.