

# OBJECT ORIENTED PROGRAMMING USING JAVA

## ASSIGNMENT - 1

1) Explain OOPs concepts.

→ Object oriented programming is a fundamental programming concept which developers use for binding the data and function together using the concepts of classes & objects.

i) OBJECT :- An entity that has state & behaviour is known as an object. It is an instance of class. An object has the three characteristics :-

1) STATE : represents the data of an object.

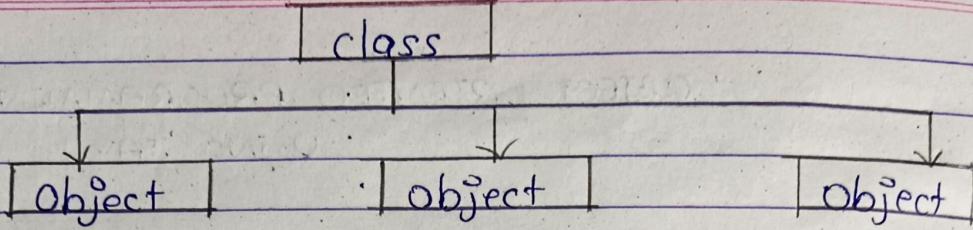
2) BEHAVIOUR : represents functionality of an object.

3) IDENTITY :- An object identity is typically implemented through a unique ID.

Example:- An is an object its name is Reynolds; color is white; known as its state; it is used to write so writing is behaviour.

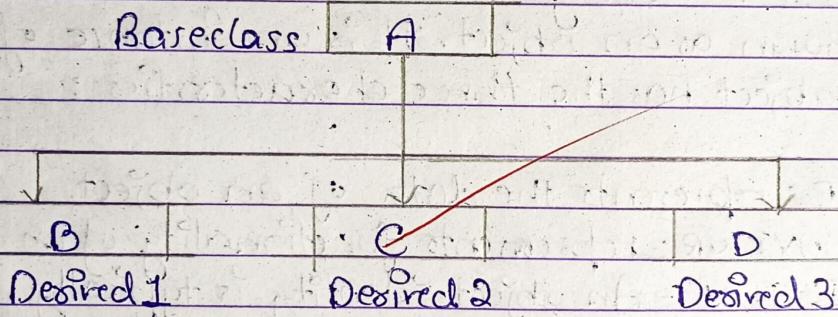
2) CLASS :- A class is a group of objects which share common properties.

- It is a template/blueprint from which objects are created.
- Class doesn't consume any space.



### 3) INHERITANCE

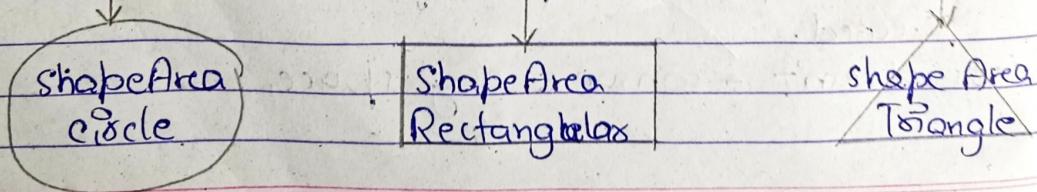
- Inheritance is one of the key features of oop that allowed us to create a new class from an existing class.
- The new class that is created is known as subclass.
- The existing class from where the child class is derived is known as superclass.
- To inherit from a class use the extend keyword.



### 4) POLYMORPHISM:

- Polymorphism means "many forms".
- If one task is performed in different ways it is known as polymorphism.
- It simply means more than one form.

Example:-



## 5) ABSTRACTION :-

- Hiding Internal details & showing functionality is known as Abstraction.
- In simple words ; the process of hiding certain details & showing only essential information to the user.
- Example: Sending SMS where you type the text & send the message. You don't know the internal processing about the message delivery.

### Abstract Keyword

↓  
Abstract class

↓  
Abstract method

## 6) ENCAPSULATION :

- Wrapping up of the code & data together into a single unit are known as encapsulation.
- The meaning of encapsulation is to make sure that "sensitive" data is hidden from user.
- Example: A capsule, it is wrapped with different medicine.

## 2) List and explain features of Java.

- (i) OBJECT ORIENTED :- Everything in Java is an object
- It means we organise our software as a combination of different types of objects that incorporate both data and behaviour.

(ii) SIMPLE :- Java is very simple and easy to learn and its syntax is similar to C, C++.

- (i) PLATFORM INDEPENDENT :- Java code can be run on multiple platforms. e.g. - Windows, Linux, Mac OS etc. multiple platforms i.e. write once and run anywhere (WORA).
- (ii) SECURED :- With Java we can develop virus free systems. It is secured because it has no explicit pointers, it runs inside JVM, class loader, byte code verifier & security manager.
- (iii) ROBUST :- Because of strong memory management, automatic garbage collection, exception handling and type checking mechanism. It is resistant to failures.
- (iv) ARCHITECTURAL NEUTRAL :- Because there is no implementation dependent features : size of primitive type is fixed.
- (v) PORTABLE :- Because it facilitates you to carry the Java bytecode to any platform. It doesn't require any type of implementation.
- (vi) HIGH PERFORMANCE :- Java is faster than other traditional interpreted programming languages because java bytecode is close to native code.
- (vii) INTERPRETED :- Java interpreted is high quantity high performance and less space occupying making it one of the fastest in current scenario.

3) DISTRIBUTED:- Because it facilitates user to create distributed applications in Java. RMI and EJS are used for creating distributed applications.

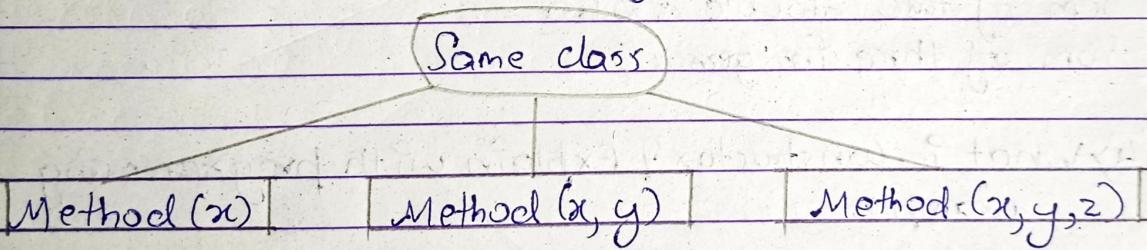
3) Explain method overloading with programming example.

→ When a class has two or more methods by the same name but different parameters, it is known as method overloading.

• Multiple methods can have the same name with different parameters.

- Method overloading is achieved by either :-
  - changing the number of arguments
  - or changing the data type of arguments.

### Method Overloading in Java



Example:-

```
public class Example {  
    public int add (int a, int b)  
    {  
        return a+b;  
    }  
}
```

```
public double add (double a, double b)  
{  
    return a+b;  
}
```

```
public int add (int a, int b, int c)
{
    return a+b+c;
}

public static void main (String [] args) {
    Example example = new Example ();
    System.out.println ("sum of two integers " + example.add (10, 20));
    System.out.println ("sum of two doubles " + example.add (10.0, 20.0));
    System.out.println ("sum of three integers " + example.add (10, 20, 30));
}
```

3  
}

OUTPUT :-

Sum of two integers : 30

Sum of two double : 31.0

Sum of three integers : 60

4) What is constructor? Explain with programming example.

- A constructor is a special member function of a class with the same name as the class name but has no return type.
- Whenever an object of a class is created using a new keyword, it invokes a constructor of that class.

## RULES :-

- i) The name of the constructor must be the same as the class name.
- ii) Java constructors do not have a return type.

## TYPES :-

### Types of constructors



Default constructor

Parameterized constructor

Default Constructors: If a constructor does not accept any parameters it is known as default constructor.

Parameterized Constructors: It takes arguments to initialize the objects attributes.

## SYNTAX:-

public class ClassName  
{

    public className() // constructor

{

    // Constructor's body

}

}

EXAMPLE :-

```
class Bike 1  
{
```

Bike 1() // creating a default constructor

```
{
```

```
} System.out.println ("Bike is created");
```

```
{
```

public static void main (String args [])

```
{
```

```
} Bike b = new Bike 1();
```

```
}
```

OUTPUT:- Bike is created.

5) Explain the java program structure.

→ DOCUMENTATION SECTION :-

- It includes basic information about a Java program.
- The information includes the name, date of creation, version, program name, company name & the description of the program

PACKAGE STATEMENT:

- The package statement identifies the package that the class belongs to
- We put classes into sensible groups with the help of packages.

## IMPORT STATEMENTS:-

- The import statements import other classes into the current class.
- This allows the current class to use the methods and variables of the imported classes.

## CLASS DEFINITION :-

- The class definition defines the class
- A class definition is composed of
  - i) class name
  - ii) class variables and methods
  - iii) Constructors

## MAIN METHOD :-

- The Java program starts with main Method

## BODY OF THE CLASS :-

- The body of the class contains the code for the class.
- The code inside a class can be divided into methods, constructors and variables.

## EXAMPLE :-

```
class sample
{
    public static void main (String [])
    {
        System.out.println ("Welcome to Java");
    }
}
```

Documentation Section

Package statement

Import statement

Interface statement

Class definition

Main Method class

{

}

1) Main Method definition

Ans: