

Assignment-3

Q.1) Distinguish between method overloading and overriding with suitable example.

→ The difference between method overloading and overriding are given below:

Method Overloading	Method Overriding
<ul style="list-style-type: none"> Using overloading, we can implement the compile time polymorphism. It is performed within a class. Parameter must be different in overloading. Overloading is used to increase the readability of a program. 	<ul style="list-style-type: none"> Using overriding, we can implement the runtime polymorphism. It is performed with two classes, i.e. parent class method overrides in the child class. Parameter must be same in overriding. Overriding is used for implementing parent class method in the child class with different functionalities.

Example's

(A) Method Overloading.

```

class Addition {
    int add (int a, int b) {
        return a+b;
    }
    int add (int a, int b, int c) {
        return a+b+c;
    }
    double add (double a, double b, double c) {
        return a+b+c;
    }
}

class MethodOverloading {
    public static void main (String args[]) {
        Addition A = new Addition();
        int x;
        double y;
        x = A.add (12, 13);
        S.O.P ("add" = " + x);
        x = A.add (12, 13, 14);
        S.O.P ("add" = " + x);
        y = A.add (22.5, 33.5, 24.5);
        S.O.P ("add" = " + y);
    }
}

OUTPUT:
add = 25
add = 39
add = 80.5
    
```

(B) Method overriding

```

class Vehicle {
    void run() {
        S.O.P("vehicle is running");
    }
    void colour() {
        S.O.P("vehicle colour is black");
    }
}
class Car extends Vehicle {
    void run() {
        S.O.P("car is running");
    }
    void colour() {
        super.colour();
    }
}
class MethodOverriding {
    public static void main (String args[]) {
        Car C1 = new Car();
        C1.run();
        C1.colour();
    }
}

```

OUTPUT:

Car is running
The vehical colour is black.

Q.2] Describe Inheritance and its types with suitable Example.

⇒ Inheritance is an important concept in OOP which allows us to define a

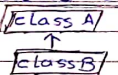
new class from an-existing class.

- with the help of inheritance a variable or method in Java code can inherit any attributes from its parent to access both the data and functionality.
- The main feature of inheritance is code reusability in which you can reuse the members of the parent class and there is no need to define the member again. Therefore, it reduces the code.

• Types of Inheritance

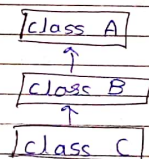
(a) Single Inheritance

- when a child class inherits the property from a single parent, it is called a single inheritance.



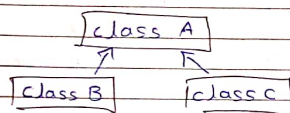
(b) Multilevel Inheritance

- If A is a parent class and B is the child class of A and inherits the property from A. C is a child class of B and inherits the property from B.
- This type of inheritance is called multilevel inheritance.



(c) Hierarchical Inheritance.

- When two or more classes inherit the same class, it is called hierarchical inheritance. For example, if B & C are inherited from the same class A or we can say A is the parent of both B & C.



(d) Hybrid inheritance.

- In this type of inheritance, there are no order of inheritance.
- All the members of the parent class are inherited in the subclass except the constructor of the parent class.
- However, a parent class constructor can be invoked from a child class.

Q-3] Explain class, object and constructor.

⇒ Class : A class is a very important concept of java because every line of code is written within a class.
It is declared using the class keyword and class name.
A class is a collection of objects that are of similar type.
For example, ice-cream, pizza, burger & chips are members of the same class - food.

Object : An object is the basic unit of OOP that has properties and actions.

An object contains a copy of all the members of its class and therefore it is called the instance of the class. In simple words, an object is anything that exists.

For example : table, car, pencil, etc.

So, anything that does not exist (or is not tangible) is not an object, such as an idea.

Constructor : A constructor is similar to a method which is used to initialize the instance variables.

It initializes the instance variables at the time of object creation.

When we do not define constructor in a class, the java compiler implicitly adds a constructor in the class at the time of compilation.

A constructor follows the following two rules:

(1) The name of a constructor is same as the class name.

(2) A constructor does not have any return type because it is a class by default.

- There are two type of constructor

which is (a) default constructor and (b) parameterized constructor.

Q.4] Define Exception, Error and Bug in Java.

⇒ Exception: An exception is an unwanted event that disturbs the normal flow of execution of a program. When exception occurs, the program terminates abnormally.

When a program is normally terminated, the resource it uses is released first.

However, if the program is abnormally terminated, the resource is not released, thereby causing more resource consumption.

Therefore, an exception is a run-time error. It is a must to handle an exception.

Error: There are three types of error in java program.

- Compile time Error: These are basically syntax errors.

For e.g., when you are typing a code, you forget to include the closing parentheses at the end of code.

This will result in compile-time error.

- Run-time Error: These error occur when you execute a java program

and because of insufficient memory of your computer system, the program fails to execute.

- Logical Errors: These error occur because you might have written the wrong program.

For example, this error might occur when you type wrong formula to calculate sum of two numbers.

Bug: A bug is a coding error in a computer program. The process of finding bugs before program users do is called debugging.

Debugging starts after the code is first written and continues in successive stages as code is combined with other units of programming to form a software product, such as an operating system or an application.

After a product is released or during public beta testing, bugs are still apt to be discovered.

When this occurs, users have to either find a way to avoid using the "buggy" code or get a patch from the originator of the code.

Q.5] Explain nested class in java.

=> In java, it is possible to define a class within another class, such classes are known as nested classes. They enable you to logically group classes that are only used in one place, thus this increases the use of encapsulation, and create more readable and maintainable code.

- A nested class is also a member of its enclosing class.

- As a member of its enclosing class, a nested class can be declared private, public, protected, or package private.

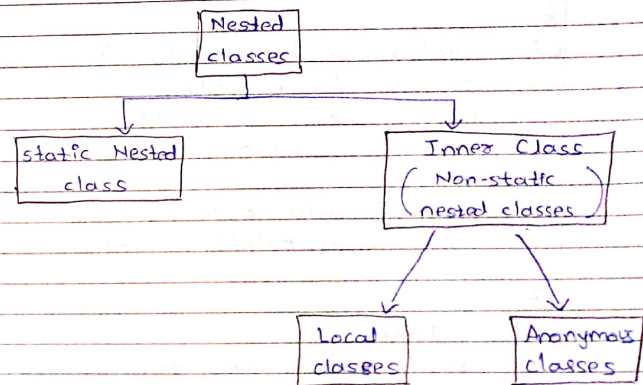
- Nested classes are divided into two categories:

1. Static nested class: Nested classes that are declared static are called static nested classes.

2. Inner class: An inner class is a non-static nested class.

Syntax:

```
class OuterClass
{
    ...
    class NestedClass
    {
        ...
    }
}
```



Q.6] Write a program to calculate area and volume of the classroom using inheritance.

```
=> import java.util.*;
class Value {
    protected float r;
    public void read (float r)
    {
        r = x;
    }
}

class Area extends Value {
    protected float area;
    public void calculate() {
        area = 3.14 * r * r * r;
    }
    public void display() {
        System.out.println("Area = " + area);
    }
}
```

```

    }
}
class Volume extends Area {
    private float Volume;
    public void Computer() {
        Volume = area *  $\pi$  *  $\frac{4}{3}$ ;
    }
    public void output() {
        System.out.println("Volume=" + Volume);
    }
}

class Inheri {
    public static void main(String args[]) {
        float k;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the Radius");
        x = sc.nextFloat();
        Volume a = new Volume();
        a.read(x);
        a.calculate();
        a.display();
        a.computer();
        a.output();
    }
}

```

OUTPUT:

Enter the Radius

10

Area = 314.0

Volume = 4186.6665