1.) What is data abstraction? Differentiate data and procedure procedural abstractions. Write inheritance hierarchy for the super class Quadrilateral, Parallelogram, Square and Rectangle. Calculate area of square, rectangle and parallelogram.

A) Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation. Consider a real life example of a man driving a car; The man only knows that pressing the accelerator will increase the speed of car and applying brakes will stop the car but he does not know about how on pressing the accelerator the speed is actually increasing, he doesn't not know about the inner mechanism of the car or the implementation of ~~accelear~~ acceleration breaks etc;

Procedural abstraction :-

procedural abstraction provides mechanisms for abstracting well defined procedures or operations as entities. The implementation of procedure requires a number of steps to be performed. A simple example is a debit operation which performs various steps to debit certain artain amount from the bank account. Hence at the banking level, credit and debit become well-defined procedural abstractions. These are extensively used by requirements analysts, as well as designers and programmers.

Procedural abstractions are normally characterized in a programming language as "function/sub-function" or "procedure".

Data ~~Ab~~ Abstraction :-

This principle is at the core of object orientation. In this form of abstraction, ~~instead~~ instead of just focusing on operations, we focus on data first and then the operations that manipulate the data. A simple example is queue data and the associated operations add() and delete(). Both add() and delete() operations manipulates queue data. In a simple procedural abstraction, there would be only add and delete operations seperately but their association with the queue data will not be captured. The advantage of data abstraction over procedural abstraction is that the data and the associated operations get specified together and hence it is easy to modify the code, when data changes.

Program

```
class Point {
        private double x;
        private double y;
        public point(double x, double y) {
                this.x=x;
                this.y=y;
        }
        public Point() { }
        public double getX() {
                return x;
        }
        public void setX(double x) {
                this.x =x;
        }
        public double getY() {
                return y;
        }
```

19BQ1A05M6

```
        Public  void setY(double Y) {
                    this.Y = Y;
        }

    }

    Public  class  Quadrilateral  extends  Point{
              private  Point P1;
              private  point P2;
              private  point P3;
              private  point P4;
              Public  Quadrelatoral(double x1, double Y1, double x2, double Y2,
                                  double x3, double Y3, double x4, double Y4){
                         this.P1 = new  Point(x1,Y1);
                         this.P2 = new  Point (x2,Y2);
                         this.P3 = new  Point(x3,Y3);
                         this.P4 = new  point (x4,Y4);
              }
              Public  Quadrilateral(double x, double Y) {
                         super(x,Y);
              }
              Public Point getP1() {
                         return P1;
              }
              Public  void  setP1(Point p1) {
                         this.P1 = P1;
              }
              public  Point  get P2(){
                         return P2;
              }
              Public  void  setP2 (Point P2) {
                         this.P2 = P2;
              }
              public Point  getP3(){
                         return P3;
              }
              public  point  set P3 (){
                         this.P3 = P3;
              }
```

19BQ1A05M6

```
                Public Point getP4() {
                        return P4;
                }
                Public void setP4(Point P4) {
                        this.P4 = P4;
                }
}
import java.text.DecimalFormat;
public class Parllalogram extends Quadrilateral {
                private double width;
                private double height;
                public Parallelogram (double x1, double y1, double x2,
                 double y2, double x3, double y3, double x4, double y4) {
                        super(x1, y1, x2, y2, x3, y3, x4, y4);
                width = Math.sqrt (Math.pow ((getP2().getX()- getP1().getX()), 2)
                        + Math.pow((getP2.getY() - getP1().get Y(), 2));
                height = Math.Sqrt (Math.pow ((getP4() .getX()- getP1().getX(), 2)
                        + Math.pow ((getP4().get Y()-getP1().get Y(), 2));
                }
                Public double area() {
                        return width * height;
                }
                Public String toString () {
                        DecimalFormat df = new DecimalFormat(".0");
                }       return "\n Area of parallelogram is :"+df.Format
                                                                (area());
}
import java.text.Decimal Format;
Public class Square extends Quadrilateral {
                Private double side;
                public Square (double x1, double y1, double x2, double y2,
                              double x3, double y3, double x4, double y4) {
                        Super (x1, y1, x2, y2, x3, y3, x4, y4);
```

19BQ1A05M6

```
                Side = Math. sqrt (Math. pow ((get P2 ().getX()- get P1().getX()),2)
                       + Math.Sqrt (Math.pow ((getP2() .getY () -getP1().getY()),2));

         } }

         public double area () {
                return side * side;
         }
         Public String toString () {
                Decimal Format df = new DecimalFormat (".0");
                return "\n Area of square is : "+df.format(area());
         }
}

import java . text . DecimalFormat;
Public class Rectangle extends Quadrilateral {
                ~~private~~
         private double width ;
         private double height;
         Public Rectangle (double x1, double y1, double x2, double y2,
         double x3, double y3, double x4, double y4){
                Super (x1,y1,x2,y2,x3,y3,x4,y4);
                width = Math.Sqrt(Math.pow((getP2().getX() - getP1().getX()),2)
                       + Math.Sqrt(Math.pow ((getP2().getY()-getP1 ().getY()),2)));
                height = Math.Sqrt (Math.pow((getP4().getX()-get P1().getX()),2)
                       + Math.Sqrt( Math.pow ((getP4().getY()-getP1().getY()),2));
         }
         Public double area () {
                return width * height;
         }
         Public String toString () {
                DecimalFormat df = new DecimalFormat(".0");
                return "\n Area of rectangle is :"+df.format (area());
         }
```

```
// Main
class QuadrilateralTest {
        public static void main ( String[] args) {
              Parllalogram parallelogram = new Parllalogram(5.0,5.0,11.0,5.0,
              12.0, 20.0, 6.0, 20.0);
              Rectangle rectangle = new Rectangle (17.0,14.0,30.0,14.0,
              30.0, 28.0, 17.0, 28.0);
              Square square = new Square(9.0,0.0,8.0, 0.0,
              8.0,9.0, 9.0,9.0);
              System.out.printf("%S %S %S\n", parallelogram, rectangle,
                                                    square);
        }
} // class ends
```

output

```
Area of the parallelogram:-90.2
Area of the Rectangle : 192.0
Area of the square : 81.0
```

19BQ1A05M6

2.) What is the importance of constructor ? Write a java program to perform constructor overloading. Describe the usage of static members and nesting members with suitable example programs in java.

A) Constructors:

Constructor in java is a special type of method that is used to initialize the object. Java constructor is invoked at the time of object creation. It constructs the values, i.e. provides data for the object that is why it is known as Constructor.

Need of using a constructor :-

When you create various objects of a class, data members are automatically allocated under each object. If you are allowed to initialize data members at the time of declaration in class then the data members corresponding to all the objects will possess the same initial values. But in practise, you would like to have different initial values for the instance variables of each object. In case, you use any member method to initialize the data members, you may need to call it seperately every time after creating an object. Hence, you require such a member method that can automatically be called while creating an object to initialize its elements. To do so you need a constructor.

Constructor overloading

A process of using a number of constructors with the same name but different types of parameters is known as constructor overloading.

Example :-

```
class ConstructorDemo
{
        int num1, num2;
        ConstructorDemo()
          {
              num1=num2=0;
          }
        ConstructorDemo (int n1, int n2)
          {
                num1 = n1
                num2 = n2
          }
        void display()
        {
           System.out.println("Num1 value is"+num1+"Num2 value is"
                                                        +num2);
        }
        Public static void main (String args[])
        {
                ConstructorDemo cd = new ConstructorDemo();
                 cd.display();
                ConstructorDemo cd1 = new ConstructorDemo(2,5);
                 cd1.display();
          }
}
```

output:-

```
Num1 value is 0 Num2 value is 0
Num1 value is 2 num2 value is 5
```

19BO1A05M6

Static Data Members

A Data member declared within a class by using static keyword is said to be a static data member. It is also known as a class variable. A static data member is a single copy available for all the objects of a class. Any change made in the static data through an object will affect the common field available for all the objects. Like static data member a member method can also be static. A static member method is a method which uses only static data members.

example of static member method:

```
class Example
{
    int a;
    static int count;
    void get count()
    {
        a=++ count;
    }
    static void give count()
    {
        System.. out. println ("Counter value"+ count);
    }
}
```

Nested Members

A member method can be used within another method to fullfill some requirements. A method is invoked under another method, This is known as nesting of member methods.

Example of Nested member methods :-

```
class Nested
{
    int a,b,c;
    double s,ar;
    public void get data ()
    {
        a=2;
        b=3;
        c=4;
    }
   public double semiperimeter()
    {
        double t =(a+b+c)/2;
         return(t);
    }
    public void area()
    {
        s= Semiperimeter ();
        ar= Math.sqrt(s*(s-a)*(s-b)*(s-c));
    }
   public void display()
    {
        System.out.println("Area of the triangle ="+ar);
    }
   public static void main ()
    {
        Nested ob = new Nested();
             ob.get data();
             ob. area();
             ob. display();
    }
}
```

19BG1A05M6

3) Define a class named Book fair with the following description:
Instance variables / Data members:
string Bname – stores the name of the book
double price – stores the price of the book
Member Methods:
(i) Book Fair () – Default constructor to initialize data members.
(ii) void Input() – To input and store the name and the price of the book.
(iii) void calculate () – To calculate the price after discount. Discount is calculated based on the following criteria.

Price	Discount
Less than & equal to RS1000	2% of price
More than RS 1000 and less than & equal to RS3000	10% of price
More than RS 3000	15% of price

(iv) void display () – To the display the name fand and price of the book after discount.
write a main method to create an object of the class and call the above member methods.

A.)
```
import java.util.Scanner;
class Book Fair {
        string Bname;
        double price;
        Scanner sc = new Scanner (System.in);
        Public BookFair(){}
        Public BookFair (string Bname, double price) {
                    this.Bname = Bname;
                    this.price = price;
        }
        Public void Input(){
                    System.out.println("Enter Book name");
                    Bname = sc.nextLine().
                    System.out.println("Enter Book price");
        }           price = sc.nextDouble();
```

```
    Public void calculate(){
            if (price <= 1000.00) {
                    price = price -(price*0.2);
            }
            else if (price >1000.00 && price <=3000.00) {
                    price = price - (price*0.1);
            }
            else {
                    price = price - (price*0.15);
            }
    }

    Public static void main (String args[]) {
            BookFair book1 = new BookFair();
            book1. Input();
            book1. Calculate ();
            book1. display ();
    }
} // class ends
```

output -1

```
Enter Book Name :
royal
Enter Book price :
2322
Name of the book = royal
price of the book after discount = 2089.8
```

output-2

```
Enter Book Name :
real men
Enter Book price:
678
Name of the book = real men
price of the book after discount = 664.44
```

output-3

```
Enter Book name :
jackpot
Enter Book price :
4554
Name of the book = jackpot
price of book after discount = 3870.9
```

4) Special words are those words which starts and ends with the same letter.

Examples:-

Existence
Comic
Window

palindrome words are those words which read the same from left to right and vice-versa.

Example:-

Malayalam
Madam
Level

All palindromes are special words, but all special words are not palindromes. Write a program to accept a word check and print whether the word is a palindrome & only special word.

A.)
```
import java.util.*;
Public class Words
{
  Public static void main (String[] args){
                    String original, reverse = " ";
                    Scanner in = new Scanner (System.in);
                    System. out. ~~prt~~println ("Enter a string to reverse:");
                    Original = in. nextLine();
                    int length = original. length();
                    for (int i = length-1; i>=0; i--)
                              reverse = reverse + original. charAt (i);
                    System. out. println(" Reverse of the string:"+reverse).
                    if (original.equals(reverse) && (original.Substring (0,1).equals(
                       original. substring (length-1))))
                              System. out. println ("palindrome");
```

```
        else if( original.substring(0,1).equals (original.substring(length-1)))
                System.out.println("Special ~~words~~");

        else
                System.out.println("None");

    }
}// class ends
```

output-1

```
Enter a string to reverse:
MADAM
Reverse of the string: MADAM
palindrome
```

output-2

```
Enter a string to reverse:
COMIC
Reverse of the string: CIMOC
special word
```

output-3:-

```
Enter a string to reverse:
ROTATOR
Reverse of the string: ROTATOR
Palindrome.
```