

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

Abstraction is the mechanism by which you hide data that is not required by the user. Abstraction allows the user to work with only the needed data and is not required to view unwanted data.

**Data abstractions** refers to providing only essential information about data and hiding the background details or implementation whereas **Procedural abstraction** provides mechanisms for abstracting well defined procedures or operations as entities. In other words it is when we know what the method or procedure does but not how it does it.

//inheritance heirarchy program.

class Quadrilateral //super class.

{

```
protected double [][] points;
public Quadrilateral(double x1, double y1, double x2, double y2,
double x3, double y3, double x4, double y4)
{
```

```
    points = new double [4][2];
```

```
    points[0][0] = x1;
```

```
    points[0][1] = y1;
```

```
    points[1][0] = x2;
```

```
    points[1][1] = y2;
```

```
    points[2][0] = x3;
```

```
    points[2][1] = y3;
```

```
    points[3][0] = x4;
```

```
    points[3][1] = y4;
```

}

}

```

// parallelogram subclass.
import java.io.*;
import java.lang.Math;
class Parallelogram extends Quadrilateral
{
    protected double base;
    protected double height;
    public Parallelogram (double x1, double y1, double x2, double y2,
    double x3x3, double y3, double x4, double y4)
    {
        Super(x1, y1, x2, y2, x3, y3, x4, y4);
        double midx = (points[0][0] + points[1][0]) / 2;
        base = Math.sqrt(Math.pow((points[0][0] - points[1][0]), 2)
        + Math.pow((points[0][1] - points[1][1]), 2));
        height = Math.sqrt(Math.pow(midx - points[3][0], 2)
        + Math.pow((points[0][1] - points[3][1]), 2));
    }
    public double area()
    {
        return base * height;
    }
}

```

```

// Square subclass.
import java.lang.Math;
class Square extends Quadrilateral {
    protected double side;
    public Square (double x1, double y1, double x2, double y2,
    double y3y3, double y3, double x4, double y4) {
        Super(x1, y1, x2, y2, x3, y3, x4, y4);
        side = Math.sqrt(Math.pow((point[0][0] - point[1][0]), 2)
        + (Math.pow((point[0][1] - point[1][1]), 2)));
    }
    public double area()
    {
        return side * side;
    }
}

```

```

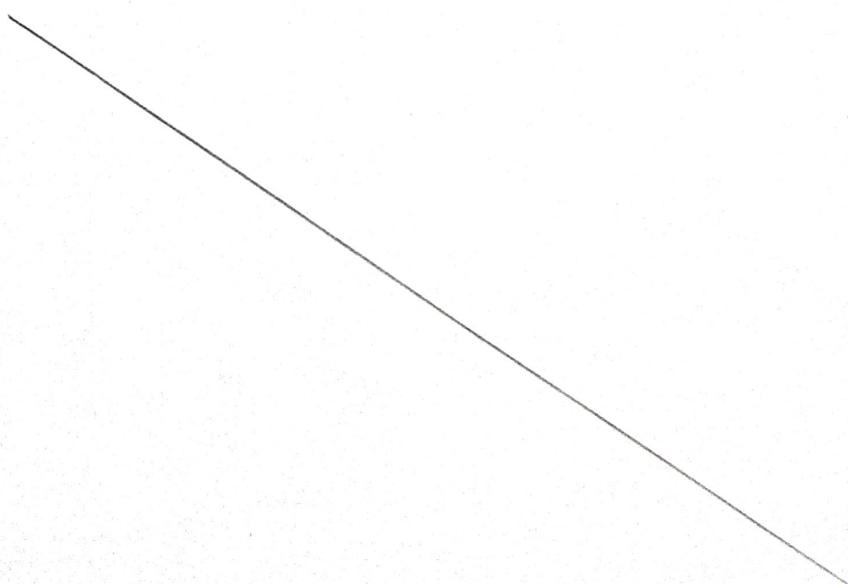
// Rectangle Subclass
import java.lang.Math;
class Rectangle extends Quadrilateral
{
    protected double length;
    protected double breadth;

    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);

        length = Math.sqrt(Math.pow((point[0][0] - point[1][0]), 2)
                           + (Math.pow((point[0][1] - point[1][1]), 2)));
        breadth = Math.sqrt(Math.pow((point[0][0] - point[3][0]), 2)
                            + (Math.pow((point[0][1] - point[3][1]), 2)));
    }

    public double area()
    {
        return length * breadth
    }
}

```



```
import java.io.*;
public class Test{
    public static void main(String args[])
    {
        double area[] = new double [3];
        Rectangle obj1 = new Rectangle(2, 2, 8, 2, 8, 8, 2, 8);
        Parallelogram obj2 = new Parallelogram(2, 2, 6, 2, 8, 8, 4, 8);
        Square obj3 = new Square(2, 2, 4, 2, 4, 4, 2, 4);
        area[0] = obj1.area();
        area[1] = obj2.area();
        area[2] = obj3.area();
        System.out.println("Area of rectangle: " + area[0]);
        System.out.println("Area of parallelogram: " + area[1]);
        System.out.println("Area of square: " + area[2]);
    }
}
```

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

Constructors are the methods having the same name as the class name and no return type. Constructors are called by JVM automatically when an object is created. Constructors are important because they initialize instance variables to default values, and they informs about the dependencies a class needs to do its job.

//constructor overloading.

```
class Multiplication
{
    int x, y;
    Multiplication() { //default constructor
        this(12, 8);
        this.multiply();
    }
    Multiplication(int x, int y) { //parameterized constructor
        this.x = x;
        this.y = y;
    }
    public void multiply() {
        int m = x * y;
        System.out.println("Product of the numbers is " + m);
    }
    public static void main(String args[]) {
        this
        Multiplication demo = new Multiplication();
    }
}
```

Static members are used to store data independent of any instance of an object i.e., you can access these members without initializing the class. It makes the program memory efficient (i.e., it saves memory).

```

// Example
import java.util.*;
public class Example {
    // static variable
    static int j=0;
    static int n;

    // static block
    static {
        System.out.println ("Static block initialized");
        n = j*8;
    }

    public static void main(String args[]) {
        System.out.println ("Value of j: " +j);
        System.out.println ("Value of n: " +n);
    }
}

```

The Java programming language allows you to define a class within another class. Such a class is called nested class.

Nested classes are divided into two categories: static and non-static. Nested classes that are declared static are called static nested ~~variables~~ classes. Non-static nested classes are called inner classes.

- A nested class is a member of its enclosing class.
- Inner classes (non-static nested classes) have access to other members of the enclosing class if even if they are declared private. Static nested classes do not have access to other members of the enclosing class.
- It increases encapsulation.

```

//Example
class Outer{
    //simple nested inner class
    class Inner{
        public void show(){
            System.out.println("This is inside a nested
                               class method");
        }
    }
}

class Main {
    public static void main(String arg[]){
        Outer.Inner in = new Outer().new Inner();
        in.show();
    }
}

```

3. Define a class named BookFair 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:

- BookFair() - Default constructor to initialize data members.
- void Input() - to input and store the name and price of the book.
- void calculate() - To calculate the price after discount. Discount is calculated based on the following criteria.

Price	Discount
Less than or equal to Rs 1000	2% of price
More than Rs 1000 and less than or equal to Rs 3000	10% of price
More than Rs 3000	15% of price

- void display() - To display the name and price of the book after discount.

```

import java.io.*;
import java.util.Scanner;
public class BookFair{
    String Bname;
    double price;

    BookFair(){
        Bname = "";
        price = 0;
    }

    public void Input(){
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter book name:");
        Bname = obj.nextLine();
        System.out.println("Enter book price:");
        price = obj.nextDouble();
    }

    void calculate(){
        double discount;
        if(price <=1000)
            discount = 2.0/100 * price;
        else if(price <=3000)
            discount = 10.0/100 * price;
        else
            discount = 15.0/100 * price;
        price = price - discount;
    }

    public static void main(String args[]){
        BookFair book = new BookFair();
        book.Input();
        book.calculate();
        book.display();
    }
}

```

4 Special words are those words which begin and end with the same letter.

EXISTENCE COMIC WINDOW

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

MALAYALAM MADAM LEVEL ROTATOR CIVIC.

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 or only a special word.

```

import java.io.*;
import java.util.Scanner;
import java.util.Arrays;
public class Words {
    public static void main(String args[]){
        String list[];
        int l, k;
        Scanner obj = new Scanner (System.in);
        System.out.println ("Enter a word");
        String word = obj.nextLine ();
        list = word.split ("");
        l = list.length;
        String temp [] = new String [l];
        int j=0;
        for (int i=l-1; i>=0; i--){
            temp[j] = list[i];
            j++;
        }
        if (list[0].equals (list[l-1])){
            if (Arrays.equals (list, temp))
                System.out.println ("The word is palindrome");
            else
                System.out.println ("The word is special word");
        }
        else
            System.out.println ("The word is neither a
                                palindrome nor a special word");
    }
}

```

## Sources:

- Core and Advanced Java - Black Book.
- Oracle - Java tutorials.  
[docs.oracle.com/javase/tutorial/java/javaOO](http://docs.oracle.com/javase/tutorial/java/javaOO).
- geeksforgeeks.org.