Set - 3

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

A. Data Abstraction:

It is process of hiding certain details and showing only essential information to the user. Abstraction can be achieved with either abstract class or interfaces. The abstract keyword is a non-access modifier, used for classes and methods.

Abstract class:

It is a restricted class that cannot be used to create objects.

-Abstract Method:

It is only used in abstract method class to instantiate at a particular subclass that doesn't contain a class body. The body is provided by subclass.

* Abstraction is a process of hiding the implementation details and only showing functionality to user.

* We use abstract methods and classes to achieve encapsulation i.e to hide certain details and only show the important details of an object.

* An abstract class can have a data members, abstract method, method body (non abstract method), constructor and much method.

Difference between procedural abstraction and data abstraction.

DATA ABSTRACTION	PROCEDURAL ABSTRACTION
① Instead of focusing on operations, we focus on data first and then operations that manipulate the data.	① They are characterized by a programming language as function/sub-function.
② The separation of logical properties of data from the details of how the data is represented.	② The separation of logical properties of an action from the details of how action is implemented.
③ The product of data abstraction is an abstract data type in classes	③ The product of procedural abstraction is a procedure.

Programs:

```
import java.lang.*;
import java.io.*;
import java.util.Scanner;

abstract class Quadrilateral {
        protected int x1,x2,x3,x4,y1,y2,y3,y4;
        protected void setCoordinates(int a, int b,
                int c, int d, int e, int f, int g, int h)
        {
                x1 = a;
                x2 = c;
                x3 = e;
                x4 = g;
                y1 = b;
                y2 = d;
                y3 = f;
                y4 = h;
        }
        public abstract int getArea();
}
```

```
class Square extends Quadrilateral {
    public Square(int a, int b, int c, int d, int e, int f,
                  int g, int h) {
        Set Coordinate(a,b,c,d,e,f,g,h);
    }
    public int getArea() {
        int d = (int) Math.sqrt((x1-x2)*(x1-x2) +
                                (y1-y2)*(y1-y2));
        return d*d;
    }
}

class Rectangle extends Quadrilateral {
    public Rectangle(int a, int b, int c, int d, int e,
                     int f, int g, int h) {
        Set Coordinate(a,b,c,d,e,f,g,h);
    }
    public int getArea() {
        int d1 = (int) Math.sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
        int d2 = (int) Math.sqrt((x1-x4)*(x1-x4)+(y1-y4)*(y1-y4));
        return d1*d2;
    }
}
```

```
class Parallelogram extends Quadrilateral {
    private int height;
    public Parallelogram (int a, int b, int c, int d, int e,
        int f, int g, int h, int height) {
            setCordinates (a, b, c, d, e, f, g, h);
            this.height = height;
        }
    public int getArea() {
        int d = (int) Math.sqrt((x1-x2)*(x1-x2) +
                (y1-y2)*(y1-y2));
        return d * height;
    }
}
public class TestQuadrilateral {
    public static void main(String args[]) {
        Square sq = new Square (10,10,20,10,20,20,10,20);
        Rectangle rec = new Rectangle (10,10,30,10,30,20,10,20);
        Parallelogram pgram = new Parallelogram (10,10,30,10,20,20,
                                20,8);
        System.out.println("Area of Square:" +
                sq.getArea());
```

```
        System.out.println("Area of Rectangle : " +
                rec.getArea());
        System.out.println("Area of parallelogram :"
                + pgm.getArea());
    }
}
```

Output:

```
Area of Square : 100
Area of Rectangle : 200
Area of parallelogram : 160
```

Q: What is importance of constructors? Write a java program to perform constructor Overloading. Describe the usage of static members and nesting members with a suitable example programs in java.

A: Constructor:

It is a block of code that initializes the newly created object. The constructor resembles instance method in java but it is not a method as it doesn't have any return type.

* Constructor has same name as class.

There are three types of constructors:

① Default constructor.

② No-arg constructor.

③ Parameterized constructor.

* Every class has a constructor whether it is normal class or abstract.

* Importance of constructors:

① The purpose of constructor is to initialize the object of class while the purpose of a method is to perform a task executing java code.

② Constructors cannot be abstract, final, static.

(5) Constructors do not has return type.

(6) At the time of calling constructor, memory for the object is allocated memory.

(7) The purpose of constructor is to assign the values to a private data.

(8) It is a member function that initialises an object or a constructor is a member or method that gets invoked without making an explicit call to it.

* Constructor overloading:

The interesting feature of a constructor is that a class can have multiple constructors. This is called constructor overloading. All constructors have same names as corresponding class. They only differ in terms of arguments. As a constructor is also a method of class it also can be overloaded.

Program:

```
import java.io.*;

class Student {
    int id;
    String name;
    int age;

    Student(int i, String n, int a) {
        id = i;
        name = n;
        age = a;
    }

void display() { System.out.println(id+" "+name+" "+age); }

public static void main(String args[]) {

        Student s1 = new Student(111, "Manish");
        Student s2 = new Student(202, "Mathews", 28);
        s1.display();
        s2.display();
    }

}
```

Output:

111 Marks 0.

222 Marks 28.

* Static Members:

→ These belong to the class and we can access these members without instantiating class.

→ Static methods can access only static fields and can be called by assigned using static.

Ex:

```
import java.io.*;

public class MyClass {

    public static int data = 20;

    public static void sample() { System.out.println("Hello"); }

}

public static void main (String args[]) {

        System.out.println(MyClass.data);

        MyClass.sample();

    }

}
```

Output:

20
Hello

* Nested Members:

It is a nested class which is declared inside the class. It can access all the members of outer class including private data members and methods.

→ Nested classes are used to develop more readable and maintainable classical code.

→ Code Specification

File:

```
import java.io.*;
class OuterClass {
    int x = 10;
    class InnerClass {
        int y = 5;
    }
}
public class MyMainClass {
public static void main (String args[]) {
    OuterClass MyOuter = new OuterClass();
    OuterClass.InnerClass myInner = myOuter.new InnerClass();
    System.out.println(myInner.y + myOuter.x);
}
}
```

Output : 15

3. Define a class named Book with following description:

Instance Variables / Data members:

(i) Bname - Name - Stores the name of book.

(ii) Bprice - Price - Stores the price of book.

Member Methods:

(i) Book() - Default constructor to initialize data members.

(ii) Void input() - To input & store the name & the price of book.

(iii) Void calculate() - To calculate price after discount.

Discount is Calculated based on the following criteria:

Price	Discount
Less than or equal to ₹1000/-	0% of price
More than ₹1000/- and less than or equal to ₹3000/-	10% of price
More than ₹3000/-	15% of price.

(iv) Void display() - To display the name and price

Write a main method to create an object of class Book and call above mentioned methods.

A.

```
import java.io.*;
import java.util.Scanner;

public class discount {
    public static void main (String [] args) {
        Bookfair obj = new Bookfair();
        obj.input ();
        obj.calculate ();
        obj.display ();
    }
}

class Bookfair {
    String name;
    double price;
    public Bookfair() {}
    public void input() {
        Scanner input = new Scanner (System.in);
        Name = input.next Line();
        price = input.next Int ();
    }

public void calculati () { if (price > 1000 && price <= 3000)
    { double = 100-10; price = (5* price) /100; }
if (price > 3000) { double = 100-15; price = (6*price)/100; }
if (price <= 1000) { double = 100-5; price = (4*price)/100; }
}
```

```
void display () {
    System.out.println ("The price after discount is :"+
                                        price);
  }
}
```

Output:

If Room - Manikanta.
If Price - 1800.

The price after discount is : 1620.0.

Special words are those words which starts and ends with same letters.

Ex: * EXISTENCE * COMIC * WINDOW

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

Ex: * 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 Special word.

A.

```
public class SpecialI
{
    public static void main(String[] args)
    {
        System.out.println(SpecialChar("WOW"));
        System.out.println(Palindrome("HI"));
    }
    public static boolean SpecialChar(String s)
    {
        int len = s.length()-1;
        if(s.charAt(0) == s.charAt(len))
            return true;
        else
            return false;
    }
    public static boolean Palindrome(String s)
    {
        String n = "";
        for(int i = s.length()-1; i>=0; i--)
        {
            n = n + s.charAt(i);
        }
        if(s.equals(n))
            return true;
        else
            return false;
    }
}
```

Output: true false