JAVA PROGAMMING ASSIGNMENT - I

Set 3

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

⇒ Data Abstraction is the process of hiding certain details and showing only essential information to the user. Abstraction can be achieved with either abstract classes or interfaces.

⇒ procedural abstraction and Data abstraction:

Procedural abstractions are normally characterized in a Programming language as "function/sub-function" or "procedure" abstraction. And used extensively by requirements analysts, as well as designers and programmers. The implementation of the procedure requires a no. of steps to be performed. A sample example is a debit operation.

Data abstraction focus on data first and then the operations that manipulate the data. A sample example is queue data and the associated operations add( ) and delete( ).

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.

```
=> // programe
   import java. io.* ;
   import java. util. Scanner;
   Public class Quadrilateral{
          public int x1, x2, x3, x4, y1, y2, y3, y4 ;
          public void setCoordinate (inta, intb, intc, intd, int e, int f, int g, int h) {
                 x1 = a;
                 y1 = b;
                 x2 = c;
                 y2 = d;
                 x3 = e;
                 y3 = f;
                 x4 = g;
                 y4 = h;
          }
   }
    public class Parallelogram extends Quadrilateral{
           private int height;
           Parallelogram (int a, int b, int c, int d, int e, int f, int g, int h, int height){
                  set Coordinate (a,b,c,d,e,f, g,h);
                  this.height = height;
           }
           int area( ){
                  int d1 = (int) Math. Sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
                  return d1* height;
           }
    }
    public class Square extends Quadrilateral {
           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);
           }
           int area( ){
                  int d2 = (int) Math. Sqrt((x1-x2)* (x1-x2) +(y1-y2)*(y1-y2));
                  return d2*d2;
    }      }
```

```
public class Rectangle extends Quadrilateral {
        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);
        }
        int area( ) {
                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;
        }
}
public class TestQuadrilateral {
      Public static void main (String [] args) {
              Square   sq = new Square (10,10,20,10,20,20,10,20);
              System.out.println("Area of Square is "+ sq.area());
              Parallelogram p = new Parallelogram(10,10,30,10,20,20,0,20,8);
              System.out.println ("Area of Parallelogram is"+ p.area());
              Rectangle  rec = new Rectangle (10,10,30,10,30,20,10,20);
              System.out.println ("Area of Rectangle is"+ rec. area());
      }
}
```

Output:

```
Area of Square is 100
Area of Parallelogram is 160
Area of Rectangle is 200
```

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.

⇒ Importance of Constructor:

Constructor is a method like a block of code which is called by java runtime during object creation using new( ) operator. Constructor are special in the sense that they have the same name as the class they are part of. They are also special in a sense that they are called by JVM automatically when you create an object.

It adds lot on readability and usability of class. when you create an object of orderProcessor class. e.g. new OrderProcessor (myQueue, myDatabase), JVM will call this constructor. If you don't add any constructor Java by default add a default no argument constructor in your class. Constructors also make it easy to test a class because fundamentally they follow Dependency injection.

⇒ Usage of static members:

Variables and methods declared using keyword static are called static members of a class. The static members are used to store data independent of any instance of an object. When a member is declared static, it can be accessed before any objects of its class are created, and without reference to any object.

⇒ Example program

```
  import java.io.*;
  Public class Test {
           // static method
           static void m1( ){
                   System.out.println ("from m1");
           }
           Public  static void main (string[ ] args){
                   // calling m1 without Creating any object of class Test
                   m1();
           }
 }
```

Output:

```
 from m1
```

⇒ Usage of Nesting members :

The Java programming language allows you to define a class within another class. Such a class is called a nested class. A nested class is a member of its enclosing class. It is a way of logically grouping classes that are only used in One place. It increases encapsulation. It can lead to more readable and maintainable code.

⇒ Example program

```
import java.io.*;
Public class TestOuter{
        static int data = 30;
        static class Inner{
              static void msg(){
                     system.out.println("data is" + data);
              }
        }
        Public static Void main (string args[ ]){
              TestOuter. Inner. msg();
        }
}
```

Output:

```
data is 30
```

⇒ //Constructor Overloading Program :

```
import java.io.*;
Public class Main{
       Main(){
            System.out.Println("Constructor with no arguments. Default Constructor.");
       }
       Main (int i){
            System.out.println("Constructor with one argument. Value of I is" +i);
       }
       Main (int i, int j){
            System.out.println(" constructor with two arguments. Value of I is"
+i+"value of J is " +j);
       }
```

```
        Public static void main (String[] args) {
                Main obj 1 = new Main( );
                Main obj 2 = new Main (5);
                Main obj 3 = new Main (10, 20);
        }
}
```

Output:-

```
Constructor with no arguments. Default Constructor.
Constructor with one argument. Value of I is 5
Constructor with two arguments. Value of I is 10 value of J is 20
```

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 :

(i) BookFair( ) – 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 or equal to Rs.1000	– 2% of price
More than Rs 1000 & Less than or equal to Rs. 3000	– 10% of price
More than Rs. 3000	– 15% of price

(iv) Void display( ) – To display the name & price of book after discount.

Write a main method to create an object of the class and call the above member methods.

⇒ // program

```
Import java.io.*;
import java.util.Scanner;
class BookFair{
        String Bname;
        double price;
        BookFair(){
                Bname = " ";
                Price = 0;
        }
        void Input(){
                Scanner S= new Scanner (System.in);
                System.out.println("Enter Book name");
                Bname = S.nextLine();
                System.out.println("Enter Price");
                Price = S.nextDouble();
        }
        Void calculate(){
                double d;
                if (price <= 1000)
                        d = 2.0/100* Price;
                else if (Price >1000 || Price <= 3000)
                        d = 10.0/100* Price;
                else if (Price > 3000)
                        d = 15.0/100* Price;
                Price = Price - d;
        }
        Void display( ){
                System.out.println ("Book name" + Bname);
                System. out. println ("price" + price);
        }
      Public static Void main( ){
                BookFair b = new BookFair( );
                b.Input( );
                b.calculate( );
                b. display( );
      }
}
```

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

Examples :-

EXISTENCE

COMIC

WINDOW

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

Examples:

MALAYALAM

MADAM

LEVEL

ROTATOR

CIVIC

All palindromes are Special words, but all special words are not palindrome

Write a program to accept a word check and print whether the word is Palindrome or only special word.

⇒ //program

```
import java.io.*;
import java.util.Scanner;
class Main {
        public static void main (String[]args){
                String str, rev = "";
                Scanner sc = new Scanner (System.in);
                System.out.println ("Enter a string:");
                str = sc.nextLine( );
                int length = str.Length( );
                for (int i =0; i<length ; i++){
                        rev = str.charAt(i) + rev;
                }
                if (str.equals (rev))
                        System.out.println (str+"is a palindrome and Special word");
                else if (str.charAt(0) = = str.charAt(length-1))
                        System.out.println (str+" is only a Special word");

                else
                        System.out.println (str+" is not a palindrome and not
   a special word");

        }
}
```

Resources :-

Data abstraction and procedural abstraction - dos.iitm.ac.in

constructor - www.java67.com

static member - docs.oracle.com , Nested member - Tutorialspoint.com