Name: P.Chamanthi
Roll-no: 19BQ1A05 I8

JAVA Assignment-1

Set-3

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

Ans: Data Abstraction:

It 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.

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 can only be used in an abstract class, it does not have a body. The body is provided by subclass.

* Abstraction is a process of hiding the implementation details and only showing functionality to user.
* An abstract class can have a data member, abstract method, method body for non abstract method, constructor and main method.
* We use abstract methods and classes to achieve security, i.e., to hide certain details and only show the important details of an object.

Differences between procedural abstraction and data abstraction:

Data Abstraction	Procedural Abstraction
1) Instead of focusing on operations, we focus on data first and then the operations that manipulate the data.	1) They are characterised in a programming language as function / sub-function
2) The seperation of the logical properties of data from the details of how the data are represented.	2) The seperation of the logical properties of an action from the details of how action is implemented
3) The product of data abstraction is an abstract data type (ADT) ie., classes.	3) The product of procedural abstraction is a procedure

* Program:

```
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 setCoordinate(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 getAreac();
}
class Square extends Quadrileteral{
    public Square(int a, int b,int c,int d,int e,int f,int g,int h){
        setCoordinate(a,b,c,d,e,f,g,h);
    }
    public int getAreac(){
        int d=(int) Math.sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
        return d*d;
    }
}
class Rectangle extends Quadrileteral{
    public Rectangle(int a,int b,int c,int d,int e,int f,int g,int h)
    {
        setCoordinate(a,b,c,d,e,f,g,h);
    }
    public int getAreac()
    {
        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 Quadrileteral {
    private int height;
    public Parallelogram(int a,int b,int c,int d,int e,int f,int g,int h,int height)
    {
        setCoordinate(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 TestQuadrileteral{
    public static void main(String args[]) {
        Square sq = new Square(10,10,20,10,20,20,10,20);
        System.out.println("Area of square: "+ sq.getArea());
        Rectangle rec = new Rectangle(10,10,30,10,30,20,10,20);
        System.out.println("Area of rectangle: "+rec.getArea());
        Parallelogram pgrm=new Parallelogram(10,10,30,10,20,20,0,20,8);
        System.out.println("Area of Parallelogram: "+pgrm.getArea());
```

```
    }
}
```

OP:

```
Area of square: 100
Area of Rectangle: 200
Area of Parallelogram: 160
```

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 a suitable example programs in java.

A. Constructor:

It is a block of code that initialises the newly created object. The constructor resembles instance method in java but its not a method as it doesnot have any return type.

* Constructor has same name as class.

There are three types of constructors

1) Default constructor
2) No-arg constructor
3) Parameterised constructor

* Every class has a constructor whether its normal class or abstract

* Importance of constructor:

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

2) Constructors cannot be abstract, final, static

3) Constructors do not have return type.

4) At the time of calling constructor, memory for the object is allocated in the memory.

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

6) 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 overloadring. All constructors have same name as corresponding class, they only differ in terms of arguments. As a constructor is also a method of a class it also can be overloaded.

* program on constructor overloading:

```
import java.io.*;
class StudentData
{
   private int stuID;
   private String stuName;
   private int stuAge;
   public StudentData()
   {
```

```
    //Default Constructor
    stuID = 100;
    stuName = "Robert Pattinson";
    stuAge = 20;
}
public StudentData (int n, String str, int num)
{
    stuID = n;
    stuName = str;
    stuAge = num;
}
public int getStuID(){
    return stuID;
}
public void setStuID (int stuID)
{
    this.stuID = stuID;
}
public void setStuName (String stuName)
{
    this.stuName = stuName;
}
public String getStuName()
{
```

```
        return stuName;
    }
    public void setStuAge(int stuAge)
    {
        this.stuAge = stuAge;
    }
    public int getStuAge()
    {
        return stuAge;
    }
}
public class OverloadingTest {
    public static void main(String args[]) {
        StudentData stu1 = new StudentData();
        StudentData stu2 = new StudentData(555, "Chaitanya", 25);
        System.out.println("Student Name is : "+stu1.getStuName());
        System.out.println("Student Age is : "+ stu1.getStuAge());
        System.out.println("Student ID is: "+ stu1.getStuID());
        System.out.println("Student Name is: "+ stu2.getStuName());
        System.out.println("Student Age is: "+ stu2.getStuAge());
        System.out.println("Student ID is: "+ stu2.getStuID());
    }
}
```

o/p:

student Name is : Robert Pattinson

Student Age is : 20

Student ID is : 100

student Name is: Chaitanya

Student Age is : 25

Student ID is : 555

* Static members: are those which belongs to the class and you can access these members without instantiating class.

→ You can create a static method using the keyword static. These static methods can access only static fields, methods.

→ When you create a static field using keyword static. These have same value in all the instances of class.

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();
    }
}
```

o/p:

20

Hello

*Nested member: 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 code

* Code optimisation.

Ex:

```
import java.io.*;
class Outerclass{
    int x=10;
    private 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);
    }
}
```

o/p

15

* static nested class:

A static class created inside a class is called static nested class. static nested class cannot access non static members.

Ex:

```
import java.io.*;
class TestOuter{
    static int data=30;
    static class Inner{
        public void msg(){
            System.out.println("data is"+data);
        }
    }
    public static void main(String args[]){
        TestOuter1.Inner obj= new TestOuter1.Inner();
        obj.msg();
    }
}
```

o/p:

data is 30

3) Define a class named BookFair with description:

Instance variables: string Bname, double price.

Member methods:

i) BookFair() → Default constructor

ii) Void Input() → Input and store data.

iii) void calculate() → To calculate price after discount

iv) void display() → To display name & price of book.

A:

```
import java.io.*;
class BookFair{
    private String Bname;
    private double price;
    public BookFair(){
        this.Bname= "Twilight";
        this.Price = 1800.00;
    }
    public void Input(String Bname,double price){
        this.Bname=Bname;
        this.price = price;
    }
    public void calculate(){
        double discount;
        if(this.price<=1000)
            discount=0.02*price;
        else if(this.price>1000 && this.price<=3000)
            discount=0.1*price;
        else
            discount=0.15*price;
        this.price=price-discount;
    }
    public void display(){
        System.out.println("Book name: "+this.Bname);
```

```
        System.out.println("Price:  "+this.price);
    }
}
import java.io.*;
public class Book{
    public static void main(String args[]){
        BookFair b1=new BookFair();
        BookFair b2=new BookFair();
         b1.calculate();
         b1.display();
         b2.Input("Java",3000.00);
         b2.calculate();
         b2.display();
    }
}
```

O/P

Book name: Twilight

Price : 1620.0

Book name: Java

Price: 2700.0

4) special words are those words which start and end with same letter

* EXISTENCE
* COMIC

Palindrome words are those words which read same from left to right

MALAYALAM
MADAM
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 palindrome or only special word.

A)

```
import java.io.*;
import java.util.Scanner;
public class PalindromeOrSpecial{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        String str=sc.next();
        int i,j,count=0;
        String []lst=str.split("");
        j=(lst.length-1);
        for(i=0;i<(lst.length/2);i++)
        {
            if(j>0){
                if(lst[i].equals(lst[j]))
                    count++;
            }
```

```
            j--;
        }
        if(count==1)
            System.out.println("Only Special Word");
        else if(count==(1st.length/2))
            System.out.println("Palindrome");
        else
            System.out.println("Not a Spl word or Palindrome");
    }
}
```

O/P

Input-1:

MALAYALAM

Output-1:

Palindrome

Input-2:

EXISTENCE

Output-2:

Only Special Word

Input-3:

JAVA

Output-3:

Not a spl word or Palindrome.

Resources:

1st Q: w3schools.com
javatpoint.com
brainly.com

2nd Q: beginnersbook.com
tutorialspoint.com

3rd Q: own

4th Q: own