III

19BQIA05D6 JAVA 1

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.

A. Data abstraction is the process of hiding certain details and shows only the required info to the users. Instead of focusing on operations that manipulate the data. The product is abstract data-type. ADT's are implemented as classes.

Difference:

*Procedural abstraction provides an operation as entity which provides mechanisms for well defined operations. They are normally showed in a language as function (or) procedure abstraction. Each method performs a function.

Eg:- String str = "Hello World";
String str1 = str.substring(0,6);

*In data abstraction, we focus on data first and operations that manipulate the data. Classes are used to abstract the related stateful values.

Here while defining the class itself, you need to identify only those attributes of class which are of those domain.

19BQ1A05D6

* Calculate the area :

```
import java.util.Scanner;
abstract class Quadrilateral {
    public abstract double area(int l, int b);
}
class Parallelogram extends Quadrilateral {
    public double area(int l, int b) {
        return l*b; }
class Rectangle extends Quadrilateral {
    public double area(int l, int b) {
        return l*b; }
class Square extends Quadrilateral {
    public double area(int l, int b) {
        return l*b; }
public class Area {
    public static void main(String[] args) {
        Square sq = new Square();
        System.out.println(sq.area(2,2));
        Parallelogram pl = new Parallelogram();
        System.out.println(pl.area(2,3));
        Rectangle rg = new Rectangle();
        System.out.println(rg.area(4,2));
    }
}
```

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 example

A * Constructor helps in initializing the private variables and default values. It is used to initialize newly created objects. If no user defined constructor is provided for a class, compiler initializes member variables to default values.

Eg:-

```
class Const{
  string str;
  Private String str1;
  Const() {}
  const (String str){
    this.str=str;
  }
  Const(string str, string str1){
    this.str1=str1;
    this.str=str;
  }
}
```

* Static member belongs to class, not object. Its value is shared between all the objects. These can be accessed without creating objects.

If value is changed by one object then all the objects having static members will have the changed values.

Example :

```
class stat {
    static int def =0;
    public void count () {
        def++;
    }
}
Public class Main {
    Public static void main(String[] args){
        stat obj 1 = new stat ();
        System.out.println(obj1.def);
        stat obj 2 = new stat ();
        System.out.println(obj 2.def);
```

* Nesting member :

It is a member of its enclosing class. They can access to that member of enclosing class. The reverse is not true, as a member, nested class can be declared private, public.

Example :

```
public class Main {
    int a=0;
    class one {
        int a=0;
    public void one_method(int x) {
```

```
            System.out.println(x);
            System.out.print(this.x);
        }
    }
    public static void main(String[]args){
    Main obj = new Main();
    obj.one obj1 = new obj.one;
    obj1.one - method(5);
    }
}
```

3.
```
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 Bname;
        double price;
        Bookfair(){}
        Public Void input() {
```

```
        Scanner inp = new Scanner(System.in);
        Bname = inp.nextLine();
        Price = inp.nextInt();
    }
    Public void calculate(){
        If (Price >1000 && Price <300) { double S=100-10 ;Price=((s*price)/100);}
        If (Price>3000){double S=100-15; Price=((s*price)/100) ;}
        If (Price<=1000){double S=100-2; price=((s*price)/100);}
    }
    Public void display(){
        System.out.println("The final price of product:" price);
    }
}
```

4.
```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    String word=input.next();
    int n= wordcheck(word, word.length());
    if (n==0)
        System.out.println("Not Palindrome, not special world");
    else if (n==1)
        System.out.println("Palindrome");
    else
        System.out.println("only special word");
    }
```

```
public static int wordcheck(String word, int n){
    if (word.charAt(0) != word.charAt(n-1))
        return 0;
    else {
        int palindrome = 1;
        for (int i=0, j=n-1; i<j; i++, j--){
            if(word.charAt(i)==word.charAt(j))
                continue;
            palindrome = 2; break;
        }
        return palindrome;
    }
}
```