1 19BQ1A05G8 Set-3

1, What is data abstraction? Differentiate data And procedural abstraction. Write inheritance Hierarchy for the Superclass Quadrilateral, Parallelogram, Square and rectangle. Calculate Area of Square, rect, and parallelogram.

A: 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.

Another way, it shows only essential things to the user and hides the internal details, for example, Sending SMS where you type the Text and Send the message..

Difference between data and procedural abstractions:

procedural abstractions are normally characterized In a programming Language as "function / Sub-function" or "procedure". It is procedural abstraction is making use of methods That accepts formal parameters and hiding The implementation from the user. This

In Procedural Abstraction, methods are used to Capture the procedural patterns, abstracting over Behaviour. We know what a method does, But we don't know how it does it.

Eg:

```
String Str = "Hello World";
String Str1 = Str.Substring(0,6);
```

It returns the part from the String start to 6th character. But we have no idea how it does the function/ method.

Data Abstraction:- In this form of abstraction, instead of just focussing on operations, we focus on data first and then the operations that manipulate the data classes are used to abstract the related stateful values and their associated behaviours - also called as (ADT). It also be defined as the process of identifying only the required characteristics of an object ignoring the irrelevent details. The properties and behaviours of an object differentiate it from other objects of similar type and also help in classifying/ grouping the objects.

Write inheritance hierarchy for the super class Quadrilateral, parallelogram, Square, and rectangle 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 s = new Square ( );
      System. out.println (s. area (5, 5) ) ;
       Parallelogram p = new parallelogram ( ) ;
      System. out. println (p. area (5,6) ) ;
        Rectangle r = new Rectangle ( ) ;
      System. out. println (r. area (10,5)) ;
}
```

(2) What is importance of Constructor ?

A:- Constructor is block of code that initialises. The newly created object. A Constructor Resembles an instance method in java but it's not a method as it doesn't have a Return type. In short Constructor and Method are different. Constructor is Special Type of method in java.

Constructor has same name as the class, And looks like this in a java Code

```
public class myclass {
    // This is Constructor
    My class ( ) {
    }
    . .
}
```

Note; the Constructor name matches with the class name and it doesn't have a return type

Eg;

```
class N {
    int private int a;
        private int b;
        N( ) { }
        N(int c) {
            a = c ;
            b = D ; }
        N (int C. int d) {
            a = c;
            b = d;
    }
}.
```

Static members: In java, static members are those which belongs to the class. and you can Access these members without instanting the class.

The static Keyword can be used with methods, Fields, classes (inner/nested), blocks.

you can Create a static method by using the Keyword static. Static methods can access only static fields, methods. To access static Methods there is no need to instantiate the class.

Eg:

```
Class A {
    static private int a=0;
           private int b;
    A() { }    c=b;
    A(int c) { a=a+1; }
        a=c;
        b=      }
}
```

```
Public Class Main {
Public static void main
        (String [] args)
{ A a = new A(4);
System.out.println(A.a); #1
A b = new A(5);
System.out.println(A.a); #2
}
}
```

Nesting Members;

A nested class is a member of its enclosing class. Non-static nested classes (inner classes) have access to other members of the enclosing class, even if they are declared private. Static nested classes do not have access to other members of the enclosing class. As a member of the outerclass, a nested class can be

Declared private, public, protected.

Ex: static nested class:

```
class A {
    .
    .
    .
    static class B {
    }
}
public class main {
    psvm (String [] args) {
    A.B a = new A.B ();
    }
}
```

inner class:

```
class A {
    .
    .
    .
    class B {
    }
}
public class main {
public static void main (String [] args)
{
    A a = new A ();
    A.B a = a.new B();
}
}
```

(3) a:

```
public class BookFair {
    String Bname;
    double price;
    BookFair (String Bname, double price) {
        This. Bname = Bname;
        This price = Price;
    }
    public class BookFair {
        String Bname;
        double price;
```

```
Book fair() string { }
    Public void Input (·) {
        Scanner input = new Scanner (System.in);
            Bname = input.next ( );
            price = input.double ( );
    }
    Public void Calculate ( ) { double discount ;
      if (price <= 1000)
        discount = (2* price)/100;
      else if (price > 1000 && price <= 3000)
        discount = (10* price)/100;
      else
        discount = (15* price)/100;
        price = price - discount;
    }
      public void display ( ) {
    System.out.print (this.Bname + "of price: "
                                    + this price);
    }
    public static void main (String [ ] args) {
        Book fair b = new Book fair ( );
        b.input ( );
        b.Calculate ( );
        b.display ( );
    }
}
```

4) Write a program to accept word check. Print whether the word is a palindrome or only special word.

A:

```
import java.util.Scanner;
class test
{
    public static void main()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("enter a word");
        String w = s.next();
        int i = w.length();
        String w1 = "";
        char ch1, ch2;
    for (int k=0; k<i; k++)
    {   ch1 = w.charAt(k);
        w1 = ch1 + w1;
    }   if (w1.equals(w) == true)
    System.out.println("it is palindrome word");
      else if (w.charAt(0) == w.charAt(i-1))
    System.out.println("it is only a special word");
    else
    System.out.println("it is not a special word");
    }
}
```