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:- Data Abstraction is the process of hiding certain details and showing only essential information to the user. In this form of abstraction, instead of just focussing on operations, we focus on data first and then the operations that manipulate the data. The product of data abstraction is an abstract data type (ADT). In object-oriented languages like Java, ADT's are implemented as classes

Difference between data and procedural abstractions:

Procedural abstraction provides mechanisms for abstracting well defined produces an operation as entity. Procedural abstractions are normally characterized in a programming language as "function/sub-function" or "procedure" abstraction. It is tied to the idea that each particular method performs a well-specified function. We know what a method does, but we donot 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 charact. 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).

In Data abstraction it means while designing/defining the classes itself, you need to identify only those attributes of class which are relevant to that domain.

As for the procedural abstraction, the necessary part is "what the procedure does and ignoring how it does it".

Write inheritance hierarchy for the super class Quadrilateral, Parallelogram, Square, and Rectangle. Calculate the area

```
import java.util.Scanner;
class Quadrilateral 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 {
    puble 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();
        s.area(5,5) 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 :-

Constructor is a special method that is used to initialize newly created object, is called just after the memory is allocated. It can be used to initialize the objects to desired values or default values. If no user-defined constructor is provided for a class, compiler initializes members variables to default values For eg:- 0 for int, null for characters and objects. It has same name as the class, it will not return a value

Eg:-

```
class A {
    int private int a ;
    private int b ;
    A ( ) { }
    A (int c) {
        a= c;
        b=0 ; }
    A(int c, int d){
        a= c;
        b= d;
    }
}
```

Static members:- Static members are those which belongs to the class, not to the object. These members can be accessed without creating an object. Static members are not part of object. The value of static members is shared same btw all the objects. If the value is changed by one object, then all the objects having static members will have the changed value.

Eg:-
```
class A {
    static int a=0;
    int c;
    A (int b){
        c=b;
        a=a+1;
    }
}
public class Main {
    public static void main(String[] args)
    {
        A a= new A(5);
        System.out.println(A.a); #1
        A b= new A(6);
        System.out.print(A.a); #2
    }
}
```

Memory allocated for static member is only once.

Nesting Members:-

Java allows you to define a class within another class. A nested class is a member of its enclosing class. The scope of nested class is bounded by scope of its enclosing class. A nested class has access to members of class in which it is nested. However, the reverse is not true. As a member of enclosing class, nested class can be declared private, public. It is of 2 types:- static nested class, inner class.

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

```
import java.util.Scanner;
```

3,
```
public class BookFair{
    string Bname;
    double price;
    BookFair(){ }
    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
        return = price - discount;
    }
```

```
    public void display(){
        System.out.print(this.Bname + "of price : "+this.price);
    }
    public static void main(String[] args){
        BookFair b= new BookFair();
        b.input();
        b.calculate1();
        b.display();
    }
}
```

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

```
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 word");
    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;}}
}
```