1.1) Write a Java program to accept IA marks obtained by five students in three subjects. The program should accept marks obtained by each student and display the total marks and the average marks. The average marks is computed as the average of best two marks obtained.

```
import java.io. *;
import java.lang.*;
import java.util.*;

class TW1_A {

static float averageOfBestTwo(int a, int b, int c){
   float avg=0;
   if(a>b && a>c) {
      if(b>c)
         return ((a+b)/2);
      else
         return ((a+c)/2);
   }
   else if(b>a && b>c){
      if(a>c)
         return ((a+b)/2);
      else
         return ((b+c)/2);
   }
   else{
      if(a>b)
         return ((c+a)/2);
      else
         return ((c+b)/2);
   }
}
   public static void main(String[]args){
      Scanner venki = new Scanner(System.in);

      int marks[][]=new int[5][3],total[]=new int[5],i,j,sum=0;
      float average=0;
      System.out.println("Enter the student data:");
      for(i=0;i<5;i++){
         sum=0;
         System.out.println("\nStudent : "+(i+1));
         for(j=0;j<3;j++){
            System.out.println("Subject "+(j+1)+" : ");
            marks[i][j]=venki.nextInt();
```

```
                sum+=marks[i][j];
            }
            total[i]=sum;
        }
        System.out.println("Results - ");
        for(i=0;i<5;i++){
            average = averageOfBestTwo(marks[i][0],marks[i][1],marks[i][2]);
            System.out.println("\nStudent "+(i+1));
            System.out.println("Total : "+total[i]);
            System.out.println("Average : "+average);
        }
    }
}
```

1.2) A company has 10 zonal sales offices in four zones namely, North, East, West and South. The company wants to organize the sales data of each of the office in each zone and find answers to queries such as,

a. Which office has performed the highest sales in each zone?
b. What is the average sales done by all the offices in each zone?
c. Which office among each zone is the poorly office?

You are required to answer the following:

i. How do you organize the above data?
ii. How do you provide answers to the above queries?

Design a Java application for the same and demonstrate the correctness of the solution.

```
import java.io.*;
import java.lang.*;
import java.util.*;
class TW1_B{
   static int maximum(int a[]){
      int m=0,pos=0;
      for(int i=0;i<10;i++){
         if(a[i]>m){
            m=a[i];
            pos=i+1;
         }
      }
      return pos;
   }
   static int minimum(int a[]){
      int m=99999,pos=0;
      for(int i=0;i<10;i++){
         if(a[i]<m){
```

```
            m=a[i];
            pos=i+1;
          }
       }
       return pos;
    }
    public static void main(String[]args){

       Scanner venki = new Scanner(System.in);

       int data[][]=new int[4][10];
       int i,j,max[]=new int[4];
       int min[]=new int[4];
       float avg[]=new float[4];
       String zones[]={"North","South","East","West"};

       System.out.println("Enter the data of the zones : ");
       for(i=0;i<4;i++){
          System.out.println("\n"+zones[i]+" : ");
          int sum=0;
          for(j=0;j<10;j++){
             System.out.println("Office "+(j+1)+" : ");
             data[i][j]=venki.nextInt();
             sum+=data[i][j];
          }
          avg[i]=sum/10;
          max[i] = maximum(data[i]);
          min[i] = minimum(data[i]);
       }

       System.out.println("\nOffice Number with highest sales zone wise : ");
       for(i=0;i<4;i++)
          System.out.println(zones[i]+" : "+max[i]);

       System.out.println("\nOffice with poorest sales zone wise : ");
       for(i=0;i<4;i++)
          System.out.println(zones
[i]+" : "+min[i]);

       System.out.println("\nAverage sales zone wise : ");
       for(i=0;i<4;i++)
          System.out.println(zones[i]+" : "+avg[i]);
```

```
    }
}
```

2.1) Design a class called Rectangle having two methods. First method named as setDim() takes length and breadth of rectangle as parameters and the second method named as getArea() returns the area of the rectangle. Length and breadth of rectangle are entered through keyboard.

```
package tw2;
import java.util.Scanner;
class Rectangle{
    int length,breadth;
    void setDim(int length,int breadth) {
            this.length=length;
            this.breadth=breadth;
    }
    int getArea() {
            return length * breadth;
    }
}
public class tw2 {

    public static void main(String[] args) {
            Rectangle r1= new Rectangle();
            Scanner in=new Scanner(System.in);
            int length,breadth;
            System.out.println("Enter the length and breadth: ");
            length=in.nextInt();
            breadth=in.nextInt();
            r1.setDim(length,breadth);
            System.out.println("Area of the rectangle is "+r1.getArea());
            in.close();
    }
}
```

2.2) Design a class called Triangle. Include methods:

i. getSides() to initialize the three sides of triangle.
ii. checkType() to determine the type of triangle represented by the three sides (Equilateral/ Isosceles/ Scalene triangle).
iii. computeArea() to compute and return the area of the triangle

```
import java.util.Scanner;
class Triangle{
    int a,b,c;
```

```
	void set(int a, int b, int c) {
		this.a=a;
		this.b=b;
		this.c=c;
	}
	double computeArea() {
		double p = (a+b+c)/2.0;
      return Math.sqrt(p*(p-a)*(p-b)*(p-c));
	}
	void checkType() {
		if (a==b && b==c) {
			System.out.println("It is an equilateral triangle");
		}
		else if(a==b || b==c || a==c){
			System.out.println("It is an isosceles triangle");
		}
		else{
			System.out.println("It is a scalene triangle");
		}
	}
}
public class Tw2pract1 {

	public static void main(String[] args) {
		int a,b,c;
		Triangle t1= new Triangle();
		Scanner in =new Scanner(System.in);
		System.out.println("Enter the three sides of triangle: ");
		a=in.nextInt();
		b=in.nextInt();
		c=in.nextInt();
      if(!(a+b>c && b+c>a && a+c>b)) {
			System.out.println("Invalid input... Program terminated");
			System.exit(0);
		}
		t1.set(a,b,c);
		System.out.println("Area of triangle is: "+t1.computeArea());
		t1.checkType();
	}

}
```

3.1) A certain small bank intends to automate few of its banking operations for its customers. Design a class by name mybankAccount to store the customer data having following details: 1.accountNumber 2. acctType 3. Name 4. Address 5. accountBalance.
The class must have both default and parameterized constructors. Write appropriate method to compute interest accrued on accountBalance based on accountType and time in years. Assume 5% for S/B account 6.5% for RD account and 7.65 for FD account. Further, add two methods withdrawAmount/depositAmount with amount as input to withdraw and deposit respectively. The withdrawAmount method must report in-sufficient balance if accountBalance falls below Rs. 1000.

TASK 1: Build the class with appropriate member variables, constructors and methods.

TASK 2: Instantiate three objects of above type and perform different operations on the same.

TASK 3: Write a function to display all the three customer details in a tabular form with appropriate column headings.

```
import static java.lang.System.exit;
import java.util.Scanner;

class Mybankaccount{
    int accno;
    String acctype,name,address;
    double balance;
    static int count=1;

    Mybankaccount(){
        Scanner s = new Scanner(System.in);
        accno = count++;
        System.out.println("Enter the account type :");
        acctype = s.nextLine();
        System.out.println("Enter the account holders name :");
        name = s.nextLine();
        System.out.println("Enter the account holders address :");
        address = s.nextLine();
        System.out.println("Enter the account holders balance :");
        balance = s.nextDouble();
    }
    Mybankaccount(String type,String name,String address,double balance){

        accno = count++;
```

```
    this.acctype = type;
    this.name = name;
    this.address = address;
    this.balance = balance;

}
void computeInterest()
{
    double interest = 0;
   switch(acctype)
   {

       case "SB" :
           interest = balance *0.05;
           balance+=interest;
           break;
       case "RD" :
           interest = balance * (6.5/100);
           balance+=interest;
           break;
       case "FD" :
           interest = balance * (7.65/100);
           balance+=interest;
           break;
   }
   System.out.println("the interest gained is :"+ interest);
}
void withdraw(double amt)
{
    if(acctype.equals("SB"))
    {
    if(balance - amt<1000)
    {
        System.out.println("Insuffient amount in balance ");
        exit(0);
```

```
        }
        else
          balance = balance - amt;
        }
        else
         System.out.println("amount Cant be withdrawn from this type of account");

    }
    void deposit(double amt)
    {
        balance = balance + amt;
    }
    void dispDetails()
    {
        System.out.println(String.format("%-5s|%-5d|%-10s|%-15s|%-
6f|",acctype,accno,name,address,balance));
    }
}

public class Tw3Bank {

    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Mybankaccount a1 = new Mybankaccount("SB","Samuel","Bhagyanagar",5000);
        Mybankaccount a2 = new Mybankaccount("RD","Rahul","Ganesh nagar",6000);
        Mybankaccount a3 = new Mybankaccount("FD","Ajay","vilas nagar",8000);

        a1.dispDetails();
        a2.dispDetails();
        a3.dispDetails();
        Mybankaccount a4 = new Mybankaccount();
        a4.dispDetails();
        a4.computeInterest();
        System.out.println("Enter the amount to be deposited :");
        double amt = s.nextDouble();
        a4.deposit(amt);
```

```
        System.out.println("Enter the amount to be withdrawn :");
        double amt1 = s.nextDouble();
        a4.withdraw(amt1);
        a4.dispDetails();
    }
}
```

3.2) Define a class to represent a rectangle in which constructors and parameterized methods are used. It also has a method to compute area of rectangle.

i. First make a class rectangle in which we declare the parameterized constructor.
ii. Then demonstrate the use of parameterized method.
iii. Use a method to compute area of rectangle.
iv. Create a class to demonstrate the call of the methods in previously created class rectangle holding constructors, parameterized methods and method to compute area of rectangle.

```
package tw3;
import java.util.Scanner;

class Rectangle{
        int length,breadth;
        Rectangle(int l,int b){
                length=l;
                breadth=b;
        }
        Rectangle(){
                //initialize an object
        }
        void changeDim(int length,int breadth) {
                this.length=length;
                this.breadth=breadth;
        }
        int area() {
                return length*breadth;
        }
}
public class tw3 {

        public static void main(String[] args) {
                Rectangle r1=new Rectangle(20,40);
                System.out.println("Area of rectangle for object r1 :"+r1.area());
                Scanner s=new Scanner(System.in);
                Rectangle r2=new Rectangle();
                System.out.println("Enter the length and breadth for rectangle: ");
                int length=s.nextInt();
                int breadth=s.nextInt();
                r2.changeDim(length, breadth);
                System.out.println("Area of rectangle for object r2 :"+r2.area());
        }
}
```

4.1) A company has two types of employees – FullTime and Partime. The company records for each employee his/her name, age, address, salary and gender. Given the basic salary of the FullTime employee the components of his/her gross salary are: Dearness allowance – 75% of basic salary, HRA – 7.5% of basic salary, IT – 10% of basic. The salary of a Partime employee is dependent on the qualification, experience, number of working hours and the rate per hour, as below:

	Qualification		
Experience	B.E.	M.Tech	PhD
1-5 Years	300 Rs.	500 Rs.	800 Rs.
6-10 Years	400 Rs.	700 Rs.	1200 Rs.
>10 Years	500 Rs.	1000 Rs.	1500 Rs.

Model this as a problem of hierarchical inheritance by:

i. Identifying the super class with its data members and member functions.
ii. Identify the sub-class/sub-classes and their associated data members and member functions.

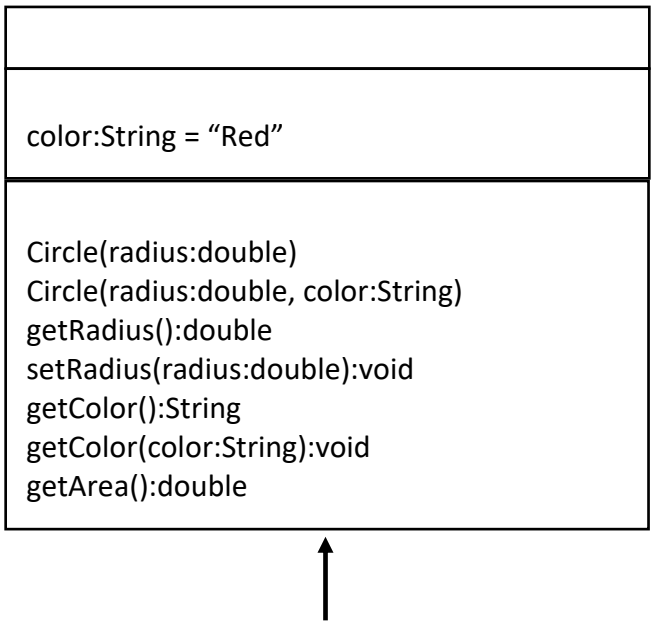
Test the program by creating objects of the classes that are so identified.

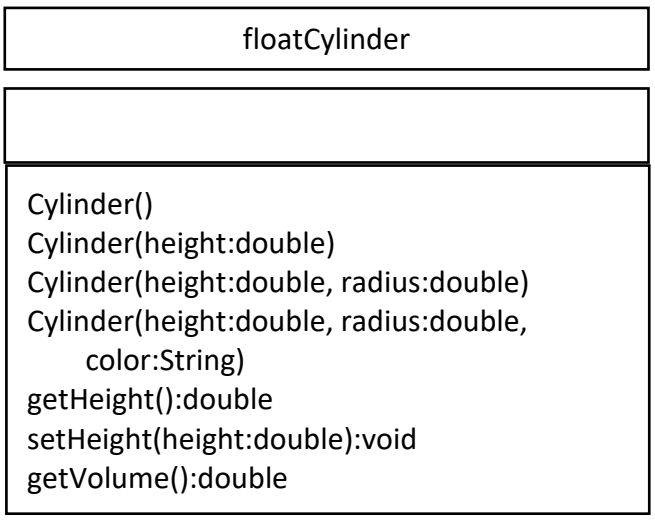
```
abstract class Employee
{
 String name;
 int age;
 String address;
 char gender;
 double salary;
 Employee(String name,int age,String address, char gender)
 {
 this.name = name;
 this.age = age;
 this.address = address;
 this.gender = gender;
 }
 void showDetails()
 {
 System.out.println("Name : "+name);
 System.out.println("Age :"+age);
 System.out.println("Address :"+address);
 System.out.println("Gender : "+gender);
 System.out.println("Salary :"+salary);
 }
 abstract void computeSalary();
}
class FTEmployee extends Employee {
 double basic;
 FTEmployee(String name,int age,String address, char gender,double basic)
 {
 super(name,age,address,gender);
 this.basic = basic;
 }
```

```
 @Override
 void computeSalary()
 {
 double da,hra,tax;
 da = basic * 0.75;
 hra = basic * 0.075;
 tax = basic * 0.1;
 salary = basic + da + hra - tax;
 }
}
class PTEmployee extends Employee
{
 String qual;
 int exp,hrsWorked;
 PTEmployee(String name,int age,String address, char gender,String qual,int exp,int hrsWorked)
 {
 super(name,age,address,gender);
 this.qual = qual;  this.exp = exp;
 this.hrsWorked = hrsWorked;
 }
 @Override
 void computeSalary()
 {
 switch(qual)
 {
 case "BE":
 if(exp>=1 && exp<=5)
 salary = hrsWorked * 300;
 else if(exp<=10)
 salary = hrsWorked * 400;
 else
 salary = hrsWorked * 500;
 break;
 case "Mtech":
 if(exp>=1 && exp<=5)
 salary = hrsWorked * 500;
 else if(exp<=10)
 salary = hrsWorked * 700;
 else
 salary = hrsWorked * 1000;
 break;
 case "phD":  if(exp>=1 && exp<=5)
 salary = hrsWorked * 800;
 else if(exp<=10)
 salary = hrsWorked * 1200;
 else
 salary = hrsWorked * 1500;
 break;
 }
 }
}
public class companyEmployee {
public static void main(String[] args) {
 FTEmployee f1 = new FTEmployee("Kiran",50,"Bhagya Nagar, Belagavi",'M',56000);
 PTEmployee p1 = new PTEmployee("Shrinivas",46,"Tilak Nagar",'M',"BE",10,100);
```

```
f1.computeSalary();
f1.showDetails();
p1.computeSalary();
p1.showDetails();
}

}
```

4.2) The class Cylinder inherits all the instance variables (radius and color) and methods (getRadius(), getArea(), among others) from its superclass Circle. It further defines a variable called height, three methods getHeight(), setHeight() and getVolume() and its own constructors. Implement the hierarchy as shown below.





```
package tw4;

//superclass
class Circle{
	double radius;
	String color;

	Circle(){
		radius = 1.0;
		color="Red";
	}
	Circle(double radius){
		this.radius=radius;
		color="Red";
	}
	Circle(double radius, String color){
		this.radius=radius;
		this.color=color;
	}
	double getRadius() {
		return radius;
	}
	void setRadius(double radius) {
		this.radius=radius;
	}
	String getColor() {
		return color;
	}
	void setColor(String color) {
		this.color=color;
	}
	double getArea() {
		return (Math.PI*radius*radius);
```

```
	}
}
//subclass
class Cylinder extends Circle{
	double height;

	Cylinder(){
		super();
		height=1.0;
	}
	Cylinder(double height){
		super();
		this.height=height;
	}
	Cylinder(double height, double radius){
		super(radius);
		this.height=height;
	}
	Cylinder(double height, double radius, String color){
		super(radius,color);
		this.height=height;
	}
	double getHeight() {
		return height;
	}
	void setHeight(double height) {
		this.height=height;
	}
	double getVolume() {
		return (Math.PI*radius*radius*height);
	}
}

public class Tw4 {
```

```
        public static void main(String[] args) {


                Circle c=new Circle(3.0,"Violet");

                System.out.println("Radius of circle = "+c.getRadius()+"\nColor of Circle = "+c.getColor());

                c.setColor("Blue");

                System.out.println("Changed Color of the cirlce : "+c.getColor()+(String.format("\nThe Area of the circle : %.2f",c.getArea())));


                Cylinder c1=new Cylinder(3.0,4.0,"Green");

                System.out.println("Radius of the Cylinder : "+c1.getRadius()+"\nHeight of the Cylinder : "+c1.getHeight()+"\nColor of the Cylinder : "+c1.getColor()+String.format("\nVolume of the Cylinder : %.2f",c1.getVolume()));

        }

    }
```

5a.1) Create a Stack class having an integer array say elem and top_of_stack as instance variables. Define three overloaded methods having the following signatures:

i. initStack(int size) to create an array of specified size and initialize the top_of_stack

ii. initStack(Stack another) to intialize the Stack object with state of the Stack object "another"

iii. initStack(int [] a) to initialize contents of a[] to the instance variable elem.

Write following methods:

i. push(): Pushes the element onto the stack,

ii. pop(): Returns the element on the top of the stack, removing it in the process, and

iii. peek(): Returns the element on the top of the stack, but does not remove it.

Also write methods that check whether stack is full and stack is empty and return boolean value true or false appropriately.

```
class MyStack{

        int []elem;

        int top;

        void initStack(int size) {

                elem=new int[size];

                top=-1;

        }

        void initStack(MyStack another) {

                elem=new int[another.elem.length];

                top=-1;
```

```
        for(int e:another.elem) {
                push(e);
        }
}
void initStack(int []a) {
        elem=new int[a.length];
        top=-1;
        for(int e:a) {
                push(e);
        }
}
 void push(int e) {
        if(isfull()) {
                System.out.println("stack overflow");
        }
        else
        {       elem[++top]=e;
        System.out.println("pushed into the stack:"+e);
        }
}
 void pop() {
         if(isEmpty())
                 System.out.println("Stack underFlow ");


         else
                 System.out.println("element is poped is :"+ elem[top--]);
 }
 void peek()
 { if(isEmpty())
         System.out.println("nothing is on the top ");
 else
         System.out.println("element on top is:"+elem[top]);
 }
 boolean isfull()
```

```
        {
                return top==elem.length-1 ? true:false;
        }
        boolean isEmpty() {
                return top==-1 ?true:false;
        }
}
public class MyClass {

        public static void main(String[] args) {
                MyStack s1= new MyStack();
                s1.initStack(5);
s1.push(10);
s1.push(20);
s1.push(30);
s1.push(40);
s1.push(50);
s1.push(60);

MyStack s2=new MyStack();
s2.initStack(s1);
s2.pop();
s2.pop();
s2.pop();
s2.pop();
s2.pop();
s2.pop();
s2.peek();
int[]a = {1,2,3,4,5,6};
MyStack s3=new MyStack();
s3.initStack(a);
s3.peek();
s3.pop();
s3.pop();
```

```
s3.pop();
}
}
```

5a.2) Implement a linear search function by using method overloading concept for an array of integers, double and character elements.

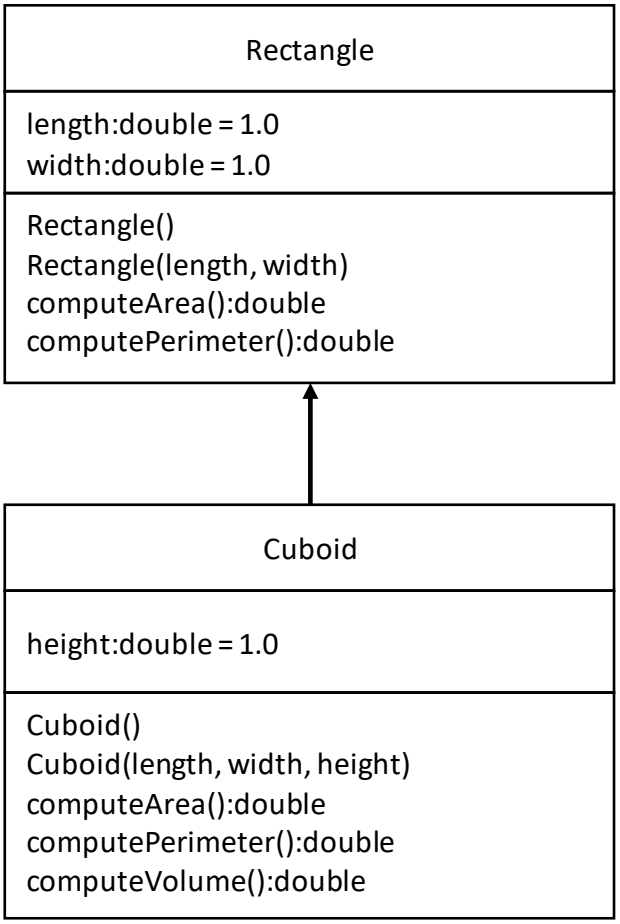
```
import java.util.Scanner;
class MyLinearSearch{

    int linearSearch(int[] c,int key2) {
        for(int i=0;i<c.length;i++) {
    if(c[i]==key2)
    {
    return i;
    }
        }
            return -1;
    }
    double linearSearch(double []a,double key) {
        for(int i=0;i<a.length;i++)
            if(a[i]==key)
                return i;

            return -1;

    }
    int linearSearch(char []a,char key) {

        for(int i=0;i<a.length;i++)
            if(a[i]==key)
                return i;

                return -1;
```

```
	}


}

public class tw5overloadinga {

	public static void main(String[] args) {

	MyLinearSearch m=new MyLinearSearch();
		Scanner in=new Scanner(System.in);
		System.out.println("what you want to search");
	System.out.println("1:integer \t 2:double  \t    3:char");
		int choice=in.nextInt();
		System.out.println("how many elements you want to enter");
		int n=in.nextInt();
		System.out.println("enter "+ n+" elements ");
		switch(choice) {

			case 1:  int[]a=new int[n];
				for(int i=0;i<n;i++)
					a[i]=in.nextInt();
		System.out.println("entered elements to be searched");
		int key=in.nextInt();
		int position=m.linearSearch(a, key);
		check(position);
		break;
			case 2:
				double[]a1=new double[n];
				for(int i=0;i<n;i++)
					a1[i]=in.nextDouble();
		System.out.println("entered elements to be searched");
		double key1=in.nextInt();
		int position1=(int)m.linearSearch(a1, key1);
```

```
                check(position1);
                break;
                        case 3:
                                char[]c=new char[n];
                                for(int i=0;i<n;i++)
                                        c[i]=in.next().charAt(0);
                System.out.println("entered elements to be searched");
                char key2= in.next().charAt(0);
                int position2=(int)m.linearSearch(c, key2);
                check(position2);
                break;
                }
        }

        private static void check(int position1) {

                if(position1>=0)
                        System.out.println("element is found at "+(position1+1)+" position");
                else
                        System.out.println("element is not found\n");
        }
}
```

Implement the following class hierarchy. In the Cuboid class, override the method computeArea() and computePerimeter() of Rectangle class to compute the surface area and perimeter of a rectangle cuboid. Add a method computeVolume() in Cuboid class to compute volume of the cuboid. Assuming length, width and height as l, w and h respectively,

- formula to find the surface area = 2(lw) + 2(hl) + 2(hw)
- formula to find the perimeter = 2l + 2w
- formula to find the volume = l x w x h



5b.1)

```
class Rectangle{
   protected double width,length;


   public Rectangle(double width, double length) {
                  this.width = width;
                  this.length = length;
          }
   public double computeArea(){
      return (length*width);
   }
   public double computePeri(){
```

```
        return 2*(length+width);
    }
}

class Cuboid extends Rectangle{
    double height;

    public Cuboid(double width, double length, double height) {
        super(width, length);
        this.height = height;
    }
    public double computeArea(){
        return (2*(length*width+height*length+height*width));
    }
    public double computePeri(){
        return (2*(length+height+width));
    }
    public double computeVolume(){
        return (length*width*height);
    }
}


public class tw5overriding {
    public static void main(String[] args) {
        Rectangle r1=new Rectangle(5, 6);
        System.out.println("Area of rectangle="+r1.computeArea());
        System.out.println("Perimeter of rectangle="+r1.computePeri());

        Cuboid c1=new Cuboid(5, 6, 8);
        System.out.println("Area of Cuboid="+c1.computeArea());
        System.out.println("Perimeter of Cuboid="+c1.computePeri());
        System.out.println("Volume of Cuboid="+c1.computeVolume());
```

```
    }
  }
```

5b.2) Design a base class ArrayStack that uses array to hold the elements and has 3 methods namely, push, pop and display. Derive a class LinkedStack that overrides these 3 methods and uses linked list to implement stack. Demonstrate the working of both the classes by performing push, pop and display operations on the objects of the above to classes.

```
import java.io.*;
import java.lang.*;
import java.util.*;
class ArrayStack{
  int top,size;
  int stack[];
  ArrayStack(int s){
    top=-1;
    size=s;
    stack=new int[size];
  }
  void push(int item){
     if(top==size-1){
       System.out.println("Stack Overflow!");
       return;
    }
    stack[++top]=item;
    System.out.println("Item Added : "+item);
  }
  void pop(){
    if(top==-1){
       System.out.println("Stack Underflow!");
       return;
    }
    System.out.println("Item popped : "+stack[top--]);
  }
  void display(){
    if(top==-1){
       System.out.println("Stack Empty!");
       return;
    }
    System.out.println("Stack items");
    for(int i=top;i>=0;i--){
       System.out.println("|"+stack[i]+"|");
```

```
        }
          System.out.println("|--|");
    }
}

class LinkedStack{

    private class Node{
        int data;       // holds data
        Node link;          // reference variable Node Type
    }
    Node top;      // Head Node
    LinkedStack(){
        this.top=null;
    }
    void push(int item){
        Node temp = new Node();
        temp.data = item;
        temp.link = top;
        top = temp;
        System.out.println("Item Pushed : "+item);
    }
    void pop(){
        if(top==null){
            System.out.printf("\nStack Underflow");
            return;
        }
        System.out.println("Item popped : "+top.data);
        top = (top).link;
    }
    void display(){
        if(top==null){
            System.out.printf("\nStack Underflow");
            return;
        }
        Node temp=top;
        while(temp!=null){
            System.out.println("|"+temp.data+"|");
            temp = temp.link;
        }
         System.out.println("|--|");
    }
```

```
}

class TW5_B{
   public static void main(String[]args){
      ArrayStack s = new ArrayStack(10);
      System.out.println(" -- Array Stack --");
      s.push(10);
      s.push(14);
      s.push(13);
      s.push(74);
      s.push(51);
      s.display();
      s.pop();
      s.pop();
      s.display();
      LinkedStack c = new LinkedStack();
      System.out.println(" -- Linked Stack --");
      c.push(11);
      c.push(21);
      c.push(11);
      c.push(74);
      c.push(15);
      c.display();
      c.pop();
      c.pop();
      c.display();
   }
}
```

6.1) Design an abstract class Car to have carName, chassiNum, modelName as member variables and add two abstract methods, startCar and operateSteering . Inherit MarutiCar and BmwCar from Car class and override the two abstract methods in their own unique way.
Design a driver class to have driver name, gender and age as data members and add a method driveCar with abstract class reference variable as argument and invoke the two basic operations namely, startCar and operateStearing and demonstrate run-time polymorphism.

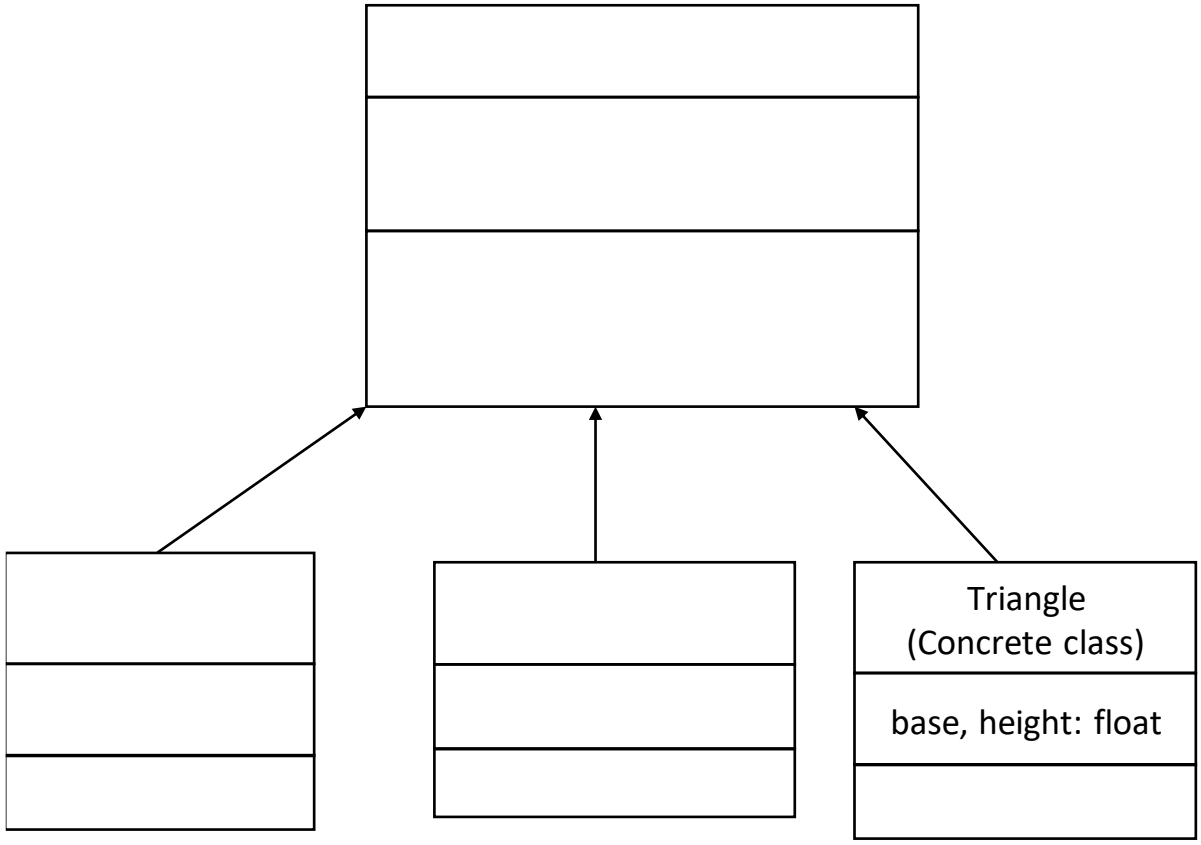
```
abstract class Car {
String carName, model;
long chassNumber;
public Car(String carName, String model, long chassNumber) {
this.carName = carName;
   this.model = model;
```

```
this.chassNumber = chassNumber;
}
public abstract void startCar();
public abstract void operateStearing();
}
class MarutiCar extends Car {
float mileage;
public MarutiCar(float mileage, String carName, String model, long chassNumber) {
super(carName, model, chassNumber);
this.mileage = mileage;
}
public void startCar(){ System.out.println("Maruti Car Started -Key Start......"); }
public void operateStearing() { System.out.println("Its manualstearing.... Its very Hard..."); }
}
class BMWCar extends Car{
float horsePower;
public BMWCar(float horsePower, String carName, String model, long
chassNumber) {
super(carName, model, chassNumber);
this.horsePower = horsePower;
}
public void startCar(){ System.out.println("BMW Car started -Touch start...."); }
public void operateStearing(){ System.out.println("Its powerstearing and is very Smooth..."); }
}
class Driver {
String name, gender;
int age;
public Driver(String name, String gender, int age) {
this.name = name;
this.gender = gender;
this.age = age;
}
public void driveCar(Car c){
System.out.println(name + " is driving "+c.carName+" car ");
c.startCar();
c.operateStearing();
}
}
public class tw6 {
   public static void main(String[] args) {
   MarutiCar c=new MarutiCar(22,"Maruti","Maruti 800",10001023);
   BMWCar c1=new BMWCar(125,"BMW-AS 6","BMW",1233422);
```

```
Driver d=new Driver("Mahendra","Male",21);
d.driveCar(c);
d.driveCar(c1);
}
}
```

a. Implement the following inheritance hierarchy.

6.2)




```
import java.util.Scanner;

abstract class Shape{
        double area;
        double perimeter;
        String type;

        abstract void computeArea();
        abstract void computePerimeter();
}

class Rectangle extends Shape{
        double length, width;
```

```
	Rectangle(){
		Scanner in = new Scanner(System.in);
		System.out.println("Enter the length and width: ");
		length = in.nextDouble();
		width = in.nextDouble();
	}

	void computeArea() {
		area = length * width;
		System.out.println("Area of the rectangle is: " + area);
	}

	void computePerimeter() {
		perimeter = 2 * (length + width);
		System.out.println("Perimeter of the rectangle is: " + perimeter);
	}
}

class Circle extends Shape{
	double radius;

	Circle(){
		Scanner in = new Scanner(System.in);
		System.out.println("Enter the radius: ");
		radius = in.nextDouble();
	}

	void computeArea() {
		area = Math.PI * radius * radius;
		System.out.println("Area of the circle is: " + area);
	}

	void computePerimeter() {
```

```
        perimeter = 2 * Math.PI * radius;
        System.out.println("Perimeter of the circle is: " + perimeter);
    }
}

class Triangle extends Shape{
    double a, b, c, s;

    Triangle(){
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the 3 sides: ");
        a = in.nextDouble();
        b = in.nextDouble();
        c = in.nextDouble();
    }

    void computeArea() {
        s = (a + b + c) / 2.0;
        area = Math.sqrt(s * (s-a) * (s-b) * (s-c));
        System.out.println("Area of the triangle is: " + area);
    }

    void computePerimeter() {
        perimeter = a + b + c;
        System.out.println("Perimeter of the triangle is: " + perimeter);
    }
}
public class tw6pract {

    public static void main(String[] args) {

        Rectangle r = new Rectangle();
        r.computeArea();
        r.computePerimeter();
```

```
                Circle c = new Circle();
                c.computeArea();
                c.computePerimeter();

                Triangle t = new Triangle();
                t.computeArea();
                t.computePerimeter();
        }

    }
```

7.1) Write a Java application to implement the following UML diagram.

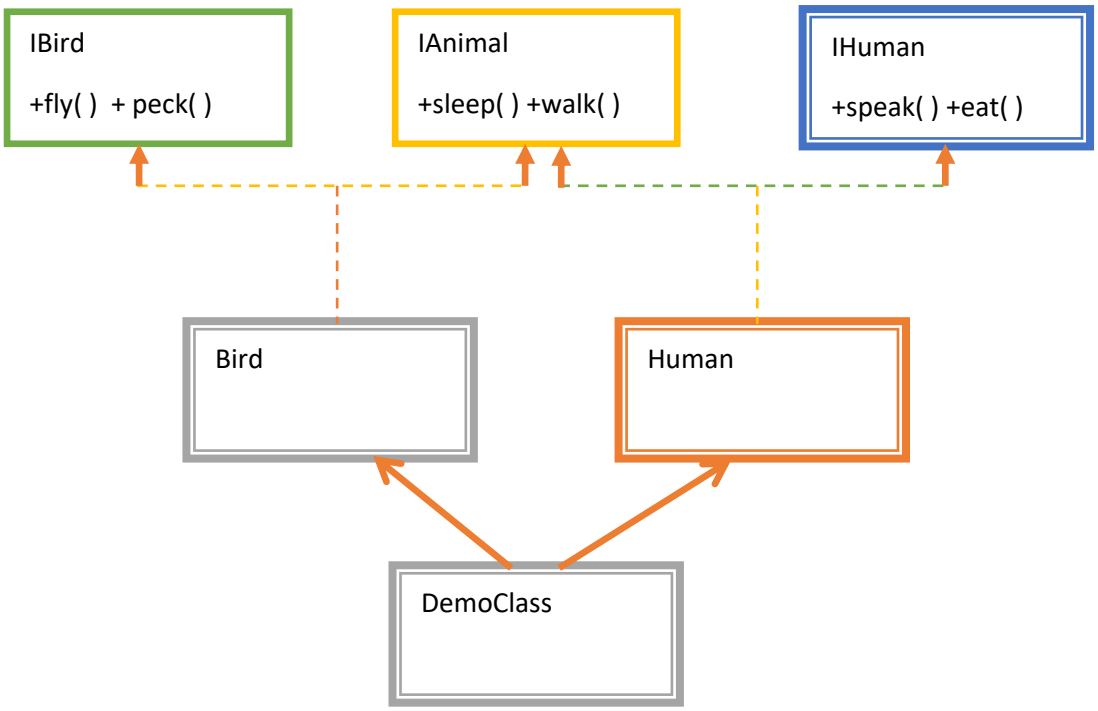
i. PrimeTester class implements isPrime() method by iterating from 2 to n-1 for a given number n
ii. ImprPrimeTester class implements isPrime() method by iterating from 2 to n/2
iii. FasterPrimeTester class implements isPrime() method by iterating from 2 to
iv. FastestPrimeTester class implements isPrime() method using Fermat's Little theorem. o Fermat's Little Theorem:
If n is a prime number, then for every a, $1 < a < n-1$, $a^n-1 \% n = 1$

```
interface IPrime{
    boolean isPrime(int n);
}


class PrimeTester implements IPrime{
    public boolean isPrime(int n) {
        boolean flag = true;
        for(int i=2; i<n; i++) {
            if(n % i == 0) {
                flag = false;
                break;
            }
        }
        return flag;
    }
}


class ImprPrimeTester implements IPrime{
    public boolean isPrime(int n) {
        boolean flag = true;
        for(int i=2; i<n/2; i++) {
```

```
                if(n % i == 0) {
                        flag = false;
                        break;
                }
            }
            return flag;
        }
}


class FasterPrimeTester implements IPrime{
        public boolean isPrime(int n){
            boolean flag = true;
            for(int i=2; i<Math.sqrt(n); i++) {
                if(n % i == 0) {
                        flag = false;
                        break;
                }
            }
            return flag;
        }

}

class FastestPrimeTester implements IPrime{
        public boolean isPrime(int n) {
            int a = 2;
            if(Math.pow(a, n-1) % n == 1) {
                return true;
            }
            else {
                return false;
            }
        }
}

public class tw7 {

        public static void main(String[] args) {
            PrimeTester p1 = new PrimeTester();
            ImprPrimeTester p2 = new ImprPrimeTester();
            FasterPrimeTester p3 = new FasterPrimeTester();
            FastestPrimeTester p4 = new FastestPrimeTester();
            System.out.println(p1.isPrime(12));
            System.out.println(p1.isPrime(13));
            System.out.println(p2.isPrime(12));
            System.out.println(p2.isPrime(13));
            System.out.println(p3.isPrime(12));
            System.out.println(p3.isPrime(13));
            System.out.println(p4.isPrime(12));
            System.out.println(p4.isPrime(13));
        }
```

}

7.2) Design an interface IAnimal that has walk, and sleep methods, an interface IBird that has fly, and peck methods, an interface IHuman that has eat and speak methods. Construct a Bird class that implements IAnimal and IBird interfaces and also construct Human class that implements IAnimal and IHuman interfaces. Demonstrate the working of these methods by invoking the methods using appropriate reference variables.




```
package tw7practice1;

interface IAnimal {

        abstract void sleep();

        abstract void walk();

}

interface IBird {

    abstract void fly();

    abstract void peek();

}
```

```
interface IHuman {

	abstract void eat();
	abstract void speak();

}

class Bird implements IAnimal, IBird {

	public void sleep()
	{
		System.out.println("Bird is sleeping");
	}
	public void walk()
	{
		System.out.println("Bird is walking");
	}

	public void fly()
	{
		System.out.println("Bird is flying");
	}
	public void peek()
	{
		System.out.println("Bird is peeking");
	}

}

class Human implements IAnimal,IHuman {

	public void sleep()
	{
```

```
                System.out.println("Human is sleeping");
        }
        public void walk()
        {
                System.out.println("Human is walking");
        }

        public void eat()
        {
                System.out.println("Human is eating");
        }

        public void speak()
        {
                System.out.println("Human is speakng");
        }
}

public class Tw7practice1 {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                Bird b=new Bird();
                b.fly();
                b.peek();
                b.sleep();
                b.walk();
                Human h= new Human();
                h.sleep();
                h.walk();
                h.speak();
                h.eat();
        }
}
```

8.1) Assume that you have received a request from the transport authority for automating the task of issuing the permanent license for two wheelers. The mandatory condition to issue the license are: 1) the applicant must over 18 years of age and 2) holder of a valid learner's license and 3) no accident cases in the last one year.
Write a Java program that reads user details as required (use the Scanner class). Create user defined exceptions to check for the three conditions imposed by the transport authority.
Based on the inputs entered by the user, decide and display whether or not a license has to be issued or an error message as defined by the user exception.

```
import java.util.Scanner;
class License{  // class named License is created
   String gender,name;
   int no_of_cases,age;
   char valid_LL;
   // necessary variables are declared
   License() // constructor is defined
   {
      Scanner s = new Scanner(System.in);
      System.out.println("Name of the Applicant        :");
      name = s.nextLine();
      System.out.println("Age                          :");
      age = Integer.parseInt(s.nextLine());
      System.out.println("Gender (M/F)                 :");
      gender = s.nextLine();
      System.out.println("Is valid LL issued (Y/N)?    :");
      valid_LL = s.nextLine().charAt(0);
      System.out.println("Num of accidents in past year :");
      no_of_cases =Integer.parseInt(s.nextLine());
      System.out.println();
// the variables will be initialised by the users input
   }
   void validateData() // method to validate the data provided by the user
   {
      try // try block to check for user defined exceptions
      {
         if(age<18)
            throw new UnderAgeException("Invalid Age");
         if(valid_LL!='Y')
            throw new InvalidLLException("Invalid LLR");
         if(no_of_cases>0)
            throw new AccidentException("Involved in Accidents ");
         System.out.println("The License can be issued ");
      }
      catch(UnderAgeException e) /* catch block to provide the message related to under age exception*/
      {
         System.out.println(e.getMessage());
         System.out.println(e);
      }

 catch(InvalidLLException e)  /* catch block to provide the message related to Invalid learning license exception*/

      {
         System.out.println(e.getMessage());
```

```
            System.out.println(e);
        }
         catch(AccidentException e)  // catch block to provide the message related to accident Exception

        {
            System.out.println(e.getMessage());
            System.out.println(e);
        }
    }
}
class UnderAgeException extends Exception /* subclass UnderAgeException is extending the superclass
Exception*/
{
    UnderAgeException(String msg)
    {
            super(msg);
    }
    public String toString(){ // overriding the toString method.
        return "The person is underage";
    }

}
class InvalidLLException  extends Exception /*subclass InvalidLLException is extending the superclass
Exception*/

{
     InvalidLLException(String msg)
    {
            super(msg);
    }
    public String toString(){ // overriding the toString method.
        return "The person hasn't issued for the LLR yet";
    }
}
class AccidentException extends Exception /*subclass AccidentException is extending the superclass
Exception.*/
{
     AccidentException(String msg)
    {
            super(msg);
    }
    public String toString(){ // overriding the toString method.
        return "The person has many accidents case on him";
    }
}
public class tw8 {
    public static void main(String[] args) {
        License l = new License(); // object of class license is instantiated.
        l.validateData(); // method validate data is called
    }

}
```

8.2) Write java program that takes the value of num variable and checks it is odd, then the throw keyword will raise the user defined exception and the catch block will get execute. OddNumberException class is derived from the Exception class. To implement user defined exception throw an exception object explicitly

```
class OddNumberException extends Exception {


public String toString()
{
        return ("number is odd");
}
}


public class tw8pract3 {


public static void main(String[] args) {
        // TODO Auto-generated method stub
                java.util.Scanner in=new java.util.Scanner(System.in);
                System.out.println("enter a number");
                int num=in.nextInt();
                try {
                   if(num%2!=0)
                           throw new OddNumberException();
    else
       System.out.println("It is an even number.");
   }
                catch(OddNumberException e)
                {
                        System.out.println(e);
                }
 }


}
```

9.1) Read a string containing 3 to 4 words using Scanner class object. Split it into words and for each

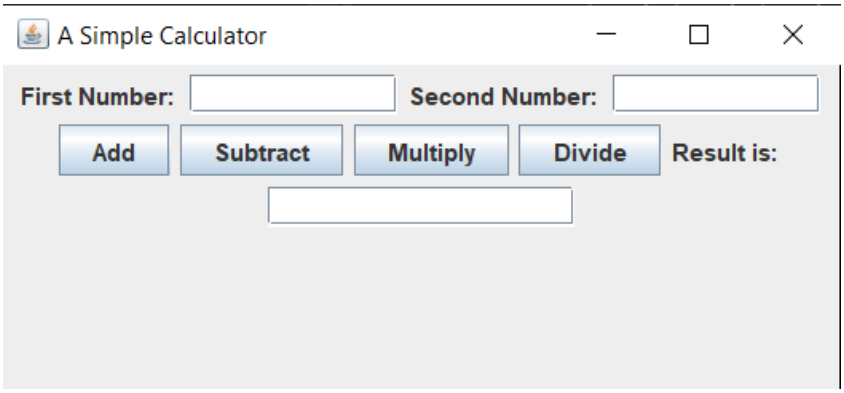
word check if it's palindrome by writing a function isPalindrome(String the myWord, int s, int e) which return true if its palindrome else return false. Where s is start index and e is end index of the input myWord. Print it in uppercase if it is palindrome else reverse the string and print it in lowercase. Use appropriate string functions to implement the above problem statement.

```
import java.util.*;

public class tw9 {
    public static void main(String[] args) {
            String inputLine; String[] allWords;
            Scanner sc=new Scanner(System.in);
            System.out.print("Enter a Sentence : ");
            inputLine=sc.nextLine();
            allWords=inputLine.split(" ");
            for(String s : allWords){
                    if(isPalindrome(s,0,s.length()-1)){
                            System.out.println(s.toUpperCase());
                            System.out.println("It is Palindrome");
                    }
                    else{
                            System.out.println(reverseString(s).toLowerCase());
                            System.out.println("It is not Palindrome");
                    }
            }
    }
    public static boolean isPalindrome(String myWord, int s, int t ){
            if(myWord.charAt(s)== myWord.charAt(t)) {
                    if(s<t)
                            return isPalindrome(myWord,s+1,t-1);
                    else
                            if(s==t)
                                    return true;
            }
            return false;
    }
    public static String reverseString(String s){
            String rS="";
            for(int i=s.length()-1;i>=0;i--)
                    rS=rS+s.charAt(i);
            return rS;
    }
}
```

9.2) Two strings will be anagram to each other if and only if they contain the same number of characters (order of the characters doesn't matter). That is, If the two strings are anagram to each other, then one string can be rearranged to form the other string. For Example: creative and reactive are anagrams. Write a Java program to test whether two strings are anagrams or not. (listen and silent, stressed and desserts, dusty and study)

```
import java.util.Arrays;

public class tw9pract {
    public static void main(String[] args) {
            String s1 = "dusty";
            String s2 = "study";
            checkAnagram(s1, s2);
    }
    static void checkAnagram(String s1, String s2) {
            char [] c1 = s1.toCharArray();
            char [] c2 = s2.toCharArray();
            if(c1.length != c2.length) {
                    System.out.println(s1 + " and " + s2 + " are " + "not Anagrams");
            }
            else {
                    Arrays.sort(c1);
                    Arrays.sort(c2);
                    if(Arrays.equals(c1, c2)) {
                            System.out.println(s1 + " and " + s2 + " are " + "Anagrams");
                    }
                    else {
                            System.out.println(s1 + " and " + s2 + " are " + "not Anagrams");
                    }
            }
    }
}
```

10.1) Design and develop a GUI application as shown below. Assume the two numbers to be integers. The application must check for invalid division condition and throw an appropriate exception.

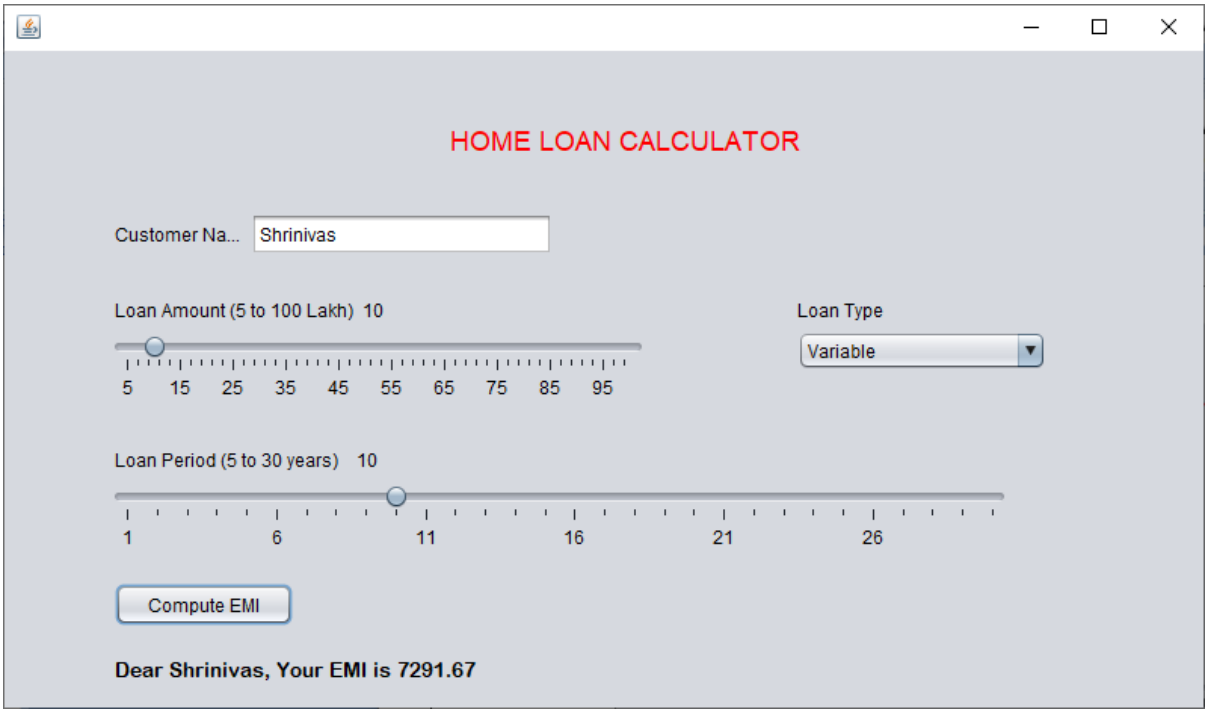



```
import javax.swing.JButton;
import javax.swing.JFrame;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;


class myCalculator implements ActionListener{
   JTextField jT1;
   JTextField jT2;
   JTextField jT3;
   public myCalculator(){
      JFrame jF=new JFrame("My Calculator App");
      jF.setSize(450,200);
      jF.setLayout(new FlowLayout());
      jF.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      JLabel jL1=new JLabel("First Number:");
      JLabel jL2=new JLabel("Second Number:");
      JLabel jL3=new JLabel("Result:");
      jT1=new JTextField(10);
      jT2=new JTextField(10);
      jT3=new JTextField(20);
```

```
        JButton jB1=new JButton("ADD");
        JButton jB2=new JButton("Subtract");
        JButton jB3=new JButton("Multiply");
        JButton jB4=new JButton("Divide");

        jF.add(jL1);jF.add(jT1);jF.add(jL2);jF.add(jT2);
        jF.add(jB1);jF.add(jB2);jF.add(jB3);jF.add(jB4);
        jF.add(jL3);jF.add(jT3);
        jB1.addActionListener(this);
        jB2.addActionListener(this);
        jB3.addActionListener(this);
        jB4.addActionListener(this);
        jF.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        jF.setVisible(true);
    }
    public void actionPerformed(ActionEvent ae){
        Float op1,op2,res;
        op1=Float.parseFloat(jT1.getText());
        op2=Float.parseFloat(jT2.getText());
        switch(ae.getActionCommand().toString()){
            case "ADD":  res=op1+op2;jT3.setText(res.toString());   ;break;
            case "Subtract":   res=op1-op2;jT3.setText(res.toString()); ;break;
            case "Multiply":   res=op1*op2;jT3.setText(res.toString()); ;break;
            case "Divide":   res=op1/op2;jT3.setText(res.toString());   ;break;
        }
    }
}

public class tw10 {
    public static void main(String[] args) {

        SwingUtilities.invokeLater(new Runnable(){
            public void run(){
```

```
            new myCalculator();
        }
    });
  }
}
```

10.2) Design and implement a Home loan Emi calculator using appropriate Swing components. The GUI should like as under:



The formula to compute Home loan EMI for a given Principal amount PA and interest rate IR for a period of T years is

$$EMI = (PA + (PA * IR * T))/12*T ;$$

Hint : Use SlideBars for Amount and Loan Period LoanType - ComboBox

```
public class tw10pract extends javax.swing.JFrame {

  public float interest = (float) 0.082  ;

  public tw10pract() {
    initComponents();
  }
```

```
@SuppressWarnings("unchecked")

private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    Title = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    CustomerNameText = new javax.swing.JTextField();
    YearsSlider = new javax.swing.JSlider();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    LoanTypeComboBox = new javax.swing.JComboBox<>();
    LoanAmountSlider = new javax.swing.JSlider();
    jLabel4 = new javax.swing.JLabel();
    EMIsentencelabel = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    TotalEmiText = new javax.swing.JTextField();
    ComputeEMIButton = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
    jPanel1.setPreferredSize(new java.awt.Dimension(838, 600));

    Title.setFont(new java.awt.Font("Microsoft YaHei UI Light", 1, 36)); // NOI18N
    Title.setForeground(new java.awt.Color(255, 0, 0));
    Title.setText("HOME LOAN CALCULATOR");

    jLabel1.setText("CUSTOMER NAME :");

    CustomerNameText.setToolTipText("Name");
    CustomerNameText.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            CustomerNameTextActionPerformed(evt);
```

```
    }
});

YearsSlider.setForeground(new java.awt.Color(0, 0, 0));
YearsSlider.setMajorTickSpacing(5);
YearsSlider.setMaximum(30);
YearsSlider.setMinimum(5);
YearsSlider.setMinorTickSpacing(1);
YearsSlider.setPaintLabels(true);
YearsSlider.setPaintTicks(true);
YearsSlider.setSnapToTicks(true);
YearsSlider.setToolTipText("");
YearsSlider.setValue(17);

jLabel2.setText("Loan Amount 5-100 lakh ");

jLabel3.setText("Loan Type");

LoanTypeComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Personal Loans.",
"Credit Cards.", "Home-Equity Loans.", "Home-Equity Lines of Credit." }));

LoanAmountSlider.setForeground(new java.awt.Color(0, 0, 0));
LoanAmountSlider.setMajorTickSpacing(10);
LoanAmountSlider.setMinimum(5);
LoanAmountSlider.setMinorTickSpacing(5);
LoanAmountSlider.setPaintLabels(true);
LoanAmountSlider.setPaintTicks(true);
LoanAmountSlider.setSnapToTicks(true);
LoanAmountSlider.setToolTipText("");

jLabel4.setText("Loan Period 5-30 years");

EMIsentencelabel.setFont(new java.awt.Font("Dialog", 1, 14)); // NOI18N
EMIsentencelabel.setPreferredSize(new java.awt.Dimension(41, 16));
```

```
jLabel5.setFont(new java.awt.Font("Dialog", 1, 14)); // NOI18N
jLabel5.setText("EMI");

TotalEmiText.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        TotalEmiTextActionPerformed(evt);
    }
});

ComputeEMIButton.setText("Compute EMI");
ComputeEMIButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ComputeEMIButtonActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(112, 112, 112)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(jLabel2)
                .addGap(89, 89, 89)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(238, 238, 238)
                        .addComponent(jLabel3))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(173, 173, 173)
```

```
                    .addComponent(LoanTypeComboBox, javax.swing.GroupLayout.PREFERRED_SIZE, 169,
javax.swing.GroupLayout.PREFERRED_SIZE))))

            .addGroup(jPanel1Layout.createSequentialGroup()

              .addComponent(jLabel1)

              .addGap(78, 78, 78)

              .addComponent(CustomerNameText, javax.swing.GroupLayout.PREFERRED_SIZE, 225,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGroup(jPanel1Layout.createSequentialGroup()

              .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addComponent(jLabel4)

                .addComponent(YearsSlider, javax.swing.GroupLayout.PREFERRED_SIZE, 191,
javax.swing.GroupLayout.PREFERRED_SIZE))

              .addGap(212, 212, 212)

              .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(TotalEmiText, javax.swing.GroupLayout.PREFERRED_SIZE, 167,
javax.swing.GroupLayout.PREFERRED_SIZE))))

          .addContainerGap(153, Short.MAX_VALUE))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

          .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

          .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

              .addComponent(Title, javax.swing.GroupLayout.PREFERRED_SIZE, 516,
javax.swing.GroupLayout.PREFERRED_SIZE)

              .addGap(139, 139, 139))

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

              .addComponent(ComputeEMIButton)

              .addGap(361, 361, 361))

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

              .addComponent(EMIsentencelabel, javax.swing.GroupLayout.PREFERRED_SIZE, 310,
javax.swing.GroupLayout.PREFERRED_SIZE)

              .addContainerGap())))

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

          .addGroup(jPanel1Layout.createSequentialGroup()

            .addGap(122, 122, 122)
```

```
            .addComponent(LoanAmountSlider, javax.swing.GroupLayout.PREFERRED_SIZE, 229,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addContainerGap(485, Short.MAX_VALUE)))

    );

    jPanel1Layout.setVerticalGroup(

      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

      .addGroup(jPanel1Layout.createSequentialGroup()

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(Title, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

        .addGap(47, 47, 47)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

          .addComponent(jLabel1)

          .addComponent(CustomerNameText, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

          .addGroup(jPanel1Layout.createSequentialGroup()

            .addGap(42, 42, 42)

            .addComponent(jLabel2))

          .addGroup(jPanel1Layout.createSequentialGroup()

            .addGap(54, 54, 54)

            .addComponent(jLabel3)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addComponent(LoanTypeComboBox, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGap(43, 43, 43)

        .addComponent(EMIsentencelabel, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

          .addComponent(jLabel4)

          .addComponent(jLabel5))

        .addGap(18, 18, 18)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

          .addComponent(YearsSlider, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addComponent(TotalEmiText, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(61, 61, 61)
            .addComponent(ComputeEMIButton)
            .addGap(138, 138, 138))
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(221, 221, 221)
                .addComponent(LoanAmountSlider, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(368, Short.MAX_VALUE)))
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap(63, Short.MAX_VALUE)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(62, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    pack();
  }// </editor-fold>

  private void CustomerNameTextActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
  }

  private void TotalEmiTextActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
  }

  private void ComputeEMIButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    int PrincipalAmount = LoanAmountSlider.getValue() * 100000 ;
    int Time = YearsSlider.getValue() ;

    EMIsentencelabel.setText(CustomerNameText.getText() + " your EMI with " +
LoanTypeComboBox.getSelectedItem()
     + " is ");

    float EMI = (PrincipalAmount + (PrincipalAmount * interest * Time ) ) / (12 * Time) ;
    TotalEmiText.setText(String.valueOf(EMI));
  }


  public static void main(String args[]) {

    try {
      for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
          javax.swing.UIManager.setLookAndFeel(info.getClassName());
          break;
        }
      }
    } catch (ClassNotFoundException ex) {
      java.util.logging.Logger.getLogger(tw10pract.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
```

```
            java.util.logging.Logger.getLogger(tw10pract.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(tw10pract.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(tw10pract.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new tw10pract().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton ComputeEMIButton;
    private javax.swing.JTextField CustomerNameText;
    private javax.swing.JLabel EMIsentencelabel;
    private javax.swing.JSlider LoanAmountSlider;
    private javax.swing.JComboBox<String> LoanTypeComboBox;
    private javax.swing.JLabel Title;
    private javax.swing.JTextField TotalEmiText;
    private javax.swing.JSlider YearsSlider;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JPanel jPanel1;
    // End of variables declaration
}
```