Practical 08

<u>Exercise 01:</u>

Create a class named "BankAccount" with private instance variables "accountNumber" and "balance." Implement encapsulation by providing public getter and setter methods for both variables. Additionally, create an abstract method called "calculateInterest" in the "BankAccount" class. Implement two subclasses, "SavingsAccount" and "CheckingAccount," that extend the "BankAccount" class and provide their own implementations of the "calculateInterest" method. Write the implementation code for the getter and setter methods in the "BankAccount" class, and the "calculateInterest" method in both the "SavingsAccount" and "CheckingAccount" classes. Assuming that the interest for saving is 12% and checking is 2% (both private variables), find out What will be the interest for a person with 1 million in his checking and 20 million in his saving account.

```
public class BankEx
{
   public static void main(String[] args)
   {
      //savings
      SavingsAccount s1=new SavingsAccount();
      s1.setBalance(20000000);
      s1.calculateInterest();

      //checking
      CheckingAccount c1=new CheckingAccount();
      c1.setBalance(1000000);
      c1.calculateInterest();
   }
}

public abstract class BankAccount
{
   //variables
   private int accountNumber;
   private double balance;

   //setters and getters
   public void setAccNo(int no)
   {
      accountNumber=no;
   }
   public int getAccNo()
   {
      return accountNumber;
   }
   public void setBalance(double b)
   {
      balance=b;
   }
```

```
    public double getBalance()
    {
        return balance;
    }

    //abstract method
    public abstract void calculateInterest();
}

public class SavingsAccount extends BankAccount
{
    private static final double SAVINGS_INTEREST_RATE=0.12;

    @Override
    public void calculateInterest()
    {
        double interest=getBalance()*SAVINGS_INTEREST_RATE;

        System.out.println("The interest for the savings account is "+interest);
    }

}

public class CheckingAccount extends BankAccount
{
    private static final double CHECKING_INTEREST_RATE=0.02;

    @Override
    public void calculateInterest()
    {
        double interest=getBalance()*CHECKING_INTEREST_RATE;

        System.out.println("The interest for the savings account is "+interest);
    }
}
```

Exercise 02:

Create an interface called "Shape" with two abstract methods: "double calculateArea()" and "double calculatePerimeter()". Implement the "Shape" interface in three classes: "Circle", "Rectangle", and "Triangle". Each class should have private instance variables relevant to its shape, and provide public getter and setter methods for these variables. Additionally, each class should define a constructor that initializes the instance variables. Write the implementation code for the "Shape" interface, the getter and setter methods in each class, and the constructors in each class.

Practical 08

```
public interface Shape
{
   public abstract double calculateArea();

   public abstract double calculatePerimeter();

}



public class Circle implements Shape
{
   //variables
   private double radius;
   protected static final double pi=3.14159;

   //constructor
   public Circle(double r)
   {
      radius=r;
   }

   //getters and setters
   public void setRadius(double r)
   {
      radius=r;
   }
   public double getRadius()
   {
      return radius;
   }

   //abstract methods
   @Override
   public double calculateArea()
   {
      return pi*radius*radius;
   }

   @Override
   public double calculatePerimeter()
   {
      return 2*pi*radius;
   }
}
```

Practical 08

```
public class Rectangle implements Shape
{
  //variables
  private double width,length;

  //constructor
  public Rectangle(double w,double l)
  {
    length=l;
    width=w;
  }

  //setters and getters
  public void setLength(double l)
  {
    length=l;
  }
  public double getLength()
  {
    return length;
  }
  public void setWidth(double w)
  {
    width=w;
  }
  public double getWidth()
  {
    return width;
  }

  @Override
  public double calculateArea()
  {
    return length*width;
  }

  @Override
  public double calculatePerimeter()
  {
    return 2*(length+width);
  }
}
```

Practical 08

```
public class Triangle implements Shape
{
   //variables
   private double length1,length2,length3;

   //constructor
   public Triangle(double L1,double L2,double L3)
   {
      length1=L1;
      length2=L2;
      length3=L3;
   }

   //setters and getters
   public void setLength1(double L1)
   {
      length1=L1;
   }
   public double getLength1()
   {
      return length1;
   }
   public void setLength2(double L2)
   {
      length2=L2;
   }
   public double getLength2()
   {
      return length2;
   }
   public void setLength3(double L3)
   {
      length3=L3;
   }
   public double getLength3()
   {
      return length3;
   }

   @Override
   public double calculateArea()
   {
      double sp=(length1+length2+length3)/2;
      return Math.sqrt(sp*(sp-length1)*(sp-length2)*(sp-length3));
   }

   @Override
   public double calculatePerimeter()
```

```
    {
        return length1+length2+length3;
    }
}



public class Shape3
{
    public static void main(String[] args)
    {
        //circle
        Circle c1=new Circle(5.0);
        System.out.println("Circle Radius: "+c1.getRadius());
        System.out.println("Circle Area: "+c1.calculateArea());
        System.out.println("Circle Perimeter: "+c1.calculatePerimeter());
        System.out.println();

        //rectangle
        Rectangle r1=new Rectangle(3.0,8.0);
        System.out.println("Rectangle Width: "+r1.getWidth());
        System.out.println("Rectangle Length: "+r1.getLength());
        System.out.println("Rectangle Area: "+r1.calculateArea());
        System.out.println("Rectangle Perimeter: "+r1.calculatePerimeter());
        System.out.println();

        //triangle
        Triangle t1=new Triangle(2.0,3.0,4.0);
        System.out.println("Triangle Length 1: "+t1.getLength1());
        System.out.println("Triangle Length 2: "+t1.getLength2());
        System.out.println("Triangle Length 3: "+t1.getLength3());
        System.out.println("Tringle Area: "+t1.calculateArea());
        System.out.println("Triangle Perimeter: "+t1.calculatePerimeter());
    }
}
```