Exercise 01:

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 abstract class BankAccount {

private int accountNumber;  
 private float balance;  
  
 abstract double calculateInterest();  
  
 public int getAccountNumber() {  
 return accountNumber;  
 }  
 public void setAccountNumber(int accountNumber) {  
 this.accountNumber = accountNumber;  
 }  
  
 public float getBalance(){  
 return balance;  
 }  
 public void setBalance(float balance){  
 this.balance = balance;  
 }

}

public class SavingsAccount extends BankAccount{

private static final double *interestForSaving* = 0.12;

@Override  
 double calculateInterest(){  
 return getBalance()\**interestForSaving*;  
 }  
}

public class CheckingAccount extends BankAccount{

private static final double *interestForChecking* = 0.02;

@Override  
 double calculateInterest(){  
 return getBalance()\**interestForChecking*;  
 }  
}

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

var savingAccount1 = new SavingsAccount();

savingAccount1.setAccountNumber(123456);  
 savingAccount1.setBalance(20\_000\_000);  
 System.*out*.println("Account no: "+savingAccount1.getAccountNumber());  
 System.*out*.println("Interest for saving account:

"+savingAccount1.calculateInterest());  
  
 var checkingAccount1 = new CheckingAccount();

checkingAccount1.setAccountNumber(456798);  
 checkingAccount1.setBalance(1\_000\_000);  
 System.*out*.println("Account no:

"+checkingAccount1.getAccountNumber());  
 System.*out*.println("Interest for checking account:

"+checkingAccount1.calculateInterest());  
 }  
  
}

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.

public class Circle implements Shape{

private int radius;  
  
 public Circle(int radius){  
 this.radius = radius;  
 }  
  
 @Override  
 public double calculatePerimeter() {  
 return 2\*Math.*PI*\*radius;  
 }  
  
 @Override  
 public double calculateArea() {  
 return Math.*PI*\*radius\*radius;  
 }  
  
 public int getRadius() {  
 return radius;  
 }  
 public void setRadius(int radius) {  
 this.radius = radius;  
 }

}

public class Rectangle implements Shape{  
 private int length;  
 private int width;  
  
 public Rectangle(int length, int width){  
 this.length = length;  
 this.width = width;  
 }  
  
 @Override  
 public double calculatePerimeter() {  
 return (length+width)\*2;  
 }  
  
 @Override  
 public double calculateArea() {  
 return length\*width;  
 }  
  
 public int getLength() {  
 return length;  
 }

public void setLength(int length) {  
 this.length = length;  
 }  
  
 public int getWidth() {  
 return width;  
 }  
 public void setWidth(int width) {  
 this.width = width;  
 }  
}

public class Triangle implements Shape{

private int length;  
  
 public Triangle(int length){  
 this.length = length;  
 }  
  
 @Override  
 public double calculatePerimeter() {  
 return length\*3;  
 }  
  
 @Override  
 public double calculateArea() {  
 return (length\*length)/2;  
 }  
  
 public int getLength() {  
 return length;  
 }

public void setLength(int length) {  
 this.length = length;  
 }

}

public interface Shape {

public double calculateArea();  
 public double calculatePerimeter();  
  
}

public class ShapePerimeter {

public static void main(String[] args){

var circle1 = new Circle(7);  
 System.*out*.println("Circle perimeter:

"+circle1.calculatePerimeter());  
 System.*out*.println("Circle area: "+circle1.calculateArea());  
  
 var rectangle1 = new Rectangle(5,6);  
 System.*out*.println("Rectangle Perimeter:

"+rectangle1.calculatePerimeter());  
 System.*out*.println("Rectangle area: "+rectangle1.calculateArea());  
  
 var triangle1 = new Triangle(8);  
 System.*out*.println("Triangle Perimeter:

"+triangle1.calculatePerimeter());  
 System.*out*.println("Triangle area: "+triangle1.calculateArea());  
 }

}