

LAB SHEET 08 – ANSWERS

Exercise 01:

```
// BankAccount class  
abstract class BankAccount  
{  
    private int accountNumber;  
    private double balance;  
  
    public BankAccount(int accountNumber, double balance)  
{  
        this.accountNumber = accountNumber;  
        this.balance = balance;  
}  
  
// Getter methods  
    public int getAccountNumber()  
{  
        return accountNumber;  
}  
  
    public double getBalance()  
{
```

```
    return balance;
}
```

```
// Setter methods
```

```
public void setAccountNumber(int accountNumber)
{
    this.accountNumber = accountNumber;
}
```

```
public void setBalance(double balance)
{
    this.balance = balance;
}
```

```
// Abstract method to be implemented in subclasses
```

```
public abstract double calculateInterest();
}
```

```
// SavingsAccount class
```

```
class SavingsAccount extends BankAccount
```

```
{
    private static final double INTEREST_RATE = 0.12; // 12%
```

```
    public SavingsAccount(int accountNumber, double balance)
```

```
{  
    super(accountNumber, balance);  
}  
  
@Override  
public double calculateInterest()  
{  
    return getBalance() * INTEREST_RATE;  
}  
}
```

// CheckingAccount class

class CheckingAccount extends BankAccount

```
{  
    private static final double INTEREST_RATE = 0.02; // 2%  
  
    public CheckingAccount(int accountNumber, double balance)  
    {  
        super(accountNumber, balance);  
    }  
  
    @Override  
    public double calculateInterest()  
    {
```

```

        return getBalance() * INTEREST_RATE;
    }
}

public class Main
{
    public static void main(String[] args)
    {
        SavingsAccount savingsAccount = new SavingsAccount(1, 20000000); // 20
        million in savings account

        CheckingAccount checkingAccount = new CheckingAccount(2, 1000000); // 1
        million in checking account

        double savingsInterest = savingsAccount.calculateInterest();
        double checkingInterest = checkingAccount.calculateInterest();

        System.out.println("Interest for Savings Account: $" + savingsInterest);
        System.out.println("Interest for Checking Account: $" + checkingInterest);
    }
}

```

Output of this program is :-

Interest for Savings Account: 2,400,000.00

Interest for Checking Account: 20,000.00

Exercise 02:

```
public interface Shape  
  
{  
  
    double calculateArea();  
  
    double calculatePerimeter();  
  
}
```

```
// Circle.java  
  
public class Circle implements Shape  
  
{  
  
    private double radius;  
  
  
    public Circle(double radius)  
  
{  
  
        this.radius = radius;  
  
    }
```

```
  
  
// Getter and Setter for radius  
  
    public double getRadius()  
  
{  
  
        return radius;
```

```
}
```

```
public void setRadius(double radius)
```

```
{
```

```
    this.radius = radius;
```

```
}
```

```
@Override
```

```
public double calculateArea()
```

```
{
```

```
    return Math.PI * radius * radius;
```

```
}
```

```
@Override
```

```
public double calculatePerimeter()
```

```
{
```

```
    return 2 * Math.PI * radius;
```

```
}
```

```
}
```

```
// Rectangle.java  
  
public class Rectangle implements Shape  
  
{  
  
    private double length;  
  
    private double width;  
  
  
    public Rectangle(double length, double width)  
  
{  
  
        this.length = length;  
  
        this.width = width;  
  
}  
  
  
// Getters and Setters for length and width  
  
public double getLength()  
  
{  
  
    return length;  
  
}  
  
  
public void setLength(double length)  
  
{  
  
    this.length = length;  
  
}  
  
  
public double getWidth()
```

```
{  
    return width;  
}  
  
public void setWidth(double width)  
{  
    this.width = width;  
}  
  
@Override  
public double calculateArea()  
{  
    return length * width;  
}  
  
@Override  
public double calculatePerimeter()  
{  
    return 2 * (length + width);  
}  
}
```



```
// Triangle.java

public class Triangle implements Shape
{
    private double sideA;
    private double sideB;
    private double sideC;

    public Triangle(double sideA, double sideB, double sideC)
    {
        this.sideA = sideA;
        this.sideB = sideB;
        this.sideC = sideC;
    }

    // Getters and Setters for sideA, sideB, and sideC
    public double getSideA()
    {
        return sideA;
    }

    public void setSideA(double sideA)
    {
        this.sideA = sideA;
    }
}
```

```
    public double getSideB()
{
    return sideB;
}
```

```
    public void setSideB(double sideB)
{
    this.sideB = sideB;
}
```

```
    public double getSideC()
{
    return sideC;
}
```

```
    public void setSideC(double sideC)
{
    this.sideC = sideC;
}
```

@Override

```
    public double calculateArea()
{
```

```
// Using Heron's formula to calculate the area of the triangle  
double s = (sideA + sideB + sideC) / 2;  
return Math.sqrt(s * (s - sideA) * (s - sideB) * (s - sideC));  
}
```

```
@Override  
public double calculatePerimeter()  
{  
    return sideA + sideB + sideC;  
}  
}
```

```
// main.java  
public class Main  
{  
    public static void main(String[] args)  
    {  
        // Circle  
        Circle circle = new Circle(5.0);  
        System.out.println("Circle Area: " + circle.calculateArea());  
        System.out.println("Circle Perimeter: " + circle.calculatePerimeter());  
    }  
}
```

```
// Rectangle
Rectangle rectangle = new Rectangle(4.0, 6.0);
System.out.println("Rectangle Area: " + rectangle.calculateArea());
System.out.println("Rectangle Perimeter: " + rectangle.calculatePerimeter());

// Triangle
Triangle triangle = new Triangle(3.0, 4.0, 5.0);
System.out.println("Triangle Area: " + triangle.calculateArea());
System.out.println("Triangle Perimeter: " + triangle.calculatePerimeter());
}
}
```

Output of this program is :-

Circle Area: 78.53981633974483

Circle Perimeter: 31.41592653589793

Rectangle Area: 24.0

Rectangle Perimeter: 20.0

Triangle Area: 6.0

Triangle Perimeter: 12.0