1.Write a Java program to create a base class Shape with a method calledcalculateArea(). Create three subclasses: Circle, Rectangle, and Triangle.Override the calculateArea() method in each subclass to calculate andreturn the shape's area.?

A.

```
abstract class Shape {
    public abstract double calculateArea();
}
class Circle extends Shape {
    private double radius;
    public Circle(double radius) {
        this.radius = radius;
    }
    public double calculateArea() {
        return Math.PI * radius * radius;
    }
}
class Rectangle extends Shape {
    private double length;
    private double width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    public double calculateArea() {
        return length * width;
    }
}
class Triangle extends Shape {
    private double base;
    private double height;
    public Triangle(double base, double height) {
        this.base = base;
```

```
        this.height = height;
    }
    public double calculateArea() {
        return 0.5 * base * height;
    }
}
public class ShapeTest {
    public static void main(String[] args) {
        Circle circle = new Circle(5);
        Rectangle rectangle = new Rectangle(4, 6);
        Triangle triangle = new Triangle(3, 8);
        System.out.println("Area of Circle: " + circle.calculateArea());
        System.out.println("Area of Rectangle: " + rectangle.calculateArea());
        System.out.println("Area of Triangle: " + triangle.calculateArea());
    }
}
```

2. Write a Java program to create a class Employee with a method calledcalculateSalary(). Create two subclasses Manager and Programmer. Ineach subclass, override the calculateSalary() method to calculate andreturn the salary based on their specific roles.?

A.

```
abstract class Employee {
    protected String name;
    protected String role;
    public Employee(String name, String role) {
        this.name = name;
        this.role = role;
    }
    public abstract double calculateSalary();
}
class Manager extends Employee {
    private double baseSalary;
    private double bonusPercentage;
```

```
    public Manager(String name, double baseSalary, double bonusPercentage) {
        super(name, "Manager");
        this.baseSalary = baseSalary;
        this.bonusPercentage = bonusPercentage;
    }
    public double calculateSalary() {
        return baseSalary + (baseSalary * bonusPercentage / 100);
    }
}
class Programmer extends Employee {
    private double hourlyRate;
    private int hoursWorked;
    public Programmer(String name, double hourlyRate, int hoursWorked) {
        super(name, "Programmer");
        this.hourlyRate = hourlyRate;
        this.hoursWorked = hoursWorked;
    }
    public double calculateSalary() {
        return hourlyRate * hoursWorked;
    }
}
public class EmployeeTest {
    public static void main(String[] args) {
        Manager manager = new Manager("John Doe", 60000.0, 10.0); // Base salary $60,000, 10% bonus
        Programmer programmer = new Programmer("Jane Smith", 50.0, 160); // $50/hour, worked 160 hours
        System.out.println("Salary of Manager " + manager.name + ": $" + manager.calculateSalary());
        System.out.println("Salary of Programmer " + programmer.name + ": $" + programmer.calculateSalary());
    }
}
```

3. Write a Java program to create a class Shape with methods getArea()and getPerimeter(). Create three subclasses: Circle, Rectangle, andTriangle. Override the getArea() and getPerimeter() methods in eachsubclass to calculate and return the area and perimeter of the respective shapes.

A.

```
abstract class Shape {
    public abstract double getArea();
    public abstract double getPerimeter();
}
class Circle extends Shape {
    private double radius;
    public Circle(double radius) {
        this.radius = radius;
    }
    public double getArea() {
        return Math.PI * radius * radius;
    }
    public double getPerimeter() {
        return 2 * Math.PI * radius;
    }
}
class Rectangle extends Shape {
    private double length;
    private double width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    public double getArea() {
        return length * width;
    }
    public double getPerimeter() {
        return 2 * (length + width);
```

```
    }
}
class Triangle extends 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;
    }
    public double getArea() {
        double s = (sideA + sideB + sideC) / 2;
        return Math.sqrt(s * (s - sideA) * (s - sideB) * (s - sideC));
    }
    public double getPerimeter() {
        return sideA + sideB + sideC;
    }
}
public class ShapeTest {
    public static void main(String[] args) {
        Circle circle = new Circle(5);
        Rectangle rectangle = new Rectangle(4, 6);
        Triangle triangle = new Triangle(3, 4, 5);
        System.out.println("Area of Circle: " + circle.getArea());
        System.out.println("Perimeter of Circle: " + circle.getPerimeter());
        System.out.println("Area of Rectangle: " + rectangle.getArea());
        System.out.println("Perimeter of Rectangle: " + rectangle.getPerimeter());
        System.out.println("Area of Triangle: " + triangle.getArea());
        System.out.println("Perimeter of Triangle: " + triangle.getPerimeter());
    }
```

```
}
```

4. Write a Java program to create a base class BankAccount withmethods deposit() and withdraw(). Create two subclassesSavingsAccount and CheckingAccount. Override the withdraw() method in each subclass to impose different withdrawal limits and fees.

A.

```
class BankAccount {
    protected double balance;
    public BankAccount(double balance) {
        this.balance = balance;
    }
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: $" + amount);
        System.out.println("Current Balance: $" + balance);
    }
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
        } else {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
        System.out.println("Current Balance: $" + balance);
    }
}
class SavingsAccount extends BankAccount {
    private double withdrawalLimit;
    public SavingsAccount(double balance, double withdrawalLimit) {
        super(balance);
        this.withdrawalLimit = withdrawalLimit;
    }
    public void withdraw(double amount) {
```

```
        if (amount <= balance && amount <= withdrawalLimit) {
            balance -= amount;
            System.out.println("Withdrawn (Savings Account): $" + amount);
        } else if (amount > withdrawalLimit) {
            System.out.println("Withdrawal amount exceeds daily withdrawal limit.");
        } else {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
        System.out.println("Current Balance (Savings Account): $" + balance);
    }
}
class CheckingAccount extends BankAccount {
    private double overdraftFee;
    public CheckingAccount(double balance, double overdraftFee) {
        super(balance);
        this.overdraftFee = overdraftFee;
    }
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn (Checking Account): $" + amount);
        } else {
            balance -= (amount + overdraftFee); // Deducting overdraft fee
            System.out.println("Withdrawn (Checking Account with overdraft fee): $" + amount);
            System.out.println("Overdraft Fee Applied: $" + overdraftFee);
        }
        System.out.println("Current Balance (Checking Account): $" + balance);
    }
}

public class BankAccountTest {
```

```
    public static void main(String[] args) {
        SavingsAccount savingsAccount = new SavingsAccount(1000.0, 500.0); // Balance $1000, Withdrawal limit $500
        CheckingAccount checkingAccount = new CheckingAccount(2000.0, 30.0); // Balance $2000, Overdraft fee $30
        savingsAccount.deposit(500.0);
        savingsAccount.withdraw(200.0); // Within limit
        savingsAccount.withdraw(700.0); // Exceeds limit
        System.out.println();
        checkingAccount.deposit(1000.0);
        checkingAccount.withdraw(1500.0); // Within balance
        checkingAccount.withdraw(2200.0); // Overdraft
    }
}
```

5. Write a Java program to create a base class Shape with methodsdraw() and calculateArea(). Create two subclasses Circle and Cylinder.Override the draw() method in each subclass to draw the respectiveshape. In addition, override the calculateArea() method in the Cylindersubclass to calculate and return the total surface area of the cylinder.

A.

```
abstract class Shape {
    public abstract void draw();
    public abstract double calculateArea();
}
class Circle extends Shape {
    private double radius;
    public Circle(double radius) {
        this.radius = radius;
    }
    public void draw() {
        System.out.println("Drawing Circle with radius " + radius);
    }
    public double calculateArea() {
        return Math.PI * radius * radius;
```

```
    }
}
class Cylinder extends Circle {
    private double height;
    public Cylinder(double radius, double height) {
        super(radius);
        this.height = height;
    }
    public void draw() {
        System.out.println("Drawing Cylinder with radius " + radius + " and height " + height);
    }
    public double calculateArea() {
        return 2 * super.calculateArea() + 2 * Math.PI * radius * height;
    }
}

public class ShapeTest {
    public static void main(String[] args) {
        Circle circle = new Circle(5);
        Cylinder cylinder = new Cylinder(3, 7);
        circle.draw();
        System.out.println("Area of Circle: " + circle.calculateArea());
        System.out.println();
        cylinder.draw();
        System.out.println("Surface Area of Cylinder: " + cylinder.calculateArea());
    }
}
```