ASSIGNMENT

WEEK 6

Q1) Write a Java program to create a class called "Person" with a name and age attribute. Create two instances of the "Person" class, set their attributes using the constructor, and print their name and age.

CODE:

```
class Person {
    private String name;
    private int age;

    // Constructor
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    // Getter methods
    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
```

```
    }
}

public class Main {
    public static void main(String[] args) {
        // Create instances of the Person class
        Person person1 = new Person("Gaurav", 20);
        Person person2 = new Person("Bob", 30);

        // Print name and age for each person
        System.out.println("Person 1 - Name: " + person1.getName() + ", Age: " +
person1.getAge());
        System.out.println("Person 2 - Name: " + person2.getName() + ", Age: " +
person2.getAge());
    }
}
```


Output:

```
Person 1 - Name: Gaurav, Age: 20
Person 2 - Name: Bob, Age: 30


...Program finished with exit code 0
Press ENTER to exit console.
```

Q2) Write a Java program to create class called "TrafficLight" with attributes for color and duration, and methods to change the color and check for red or green.

CODE:

```
class TrafficLight {
    private String color;
    private int duration; // in seconds

    public TrafficLight(String initialColor, int initialDuration) {
        color = initialColor;
        duration = initialDuration;
    }

    public void changeColor(String newColor, int newDuration) {
        color = newColor;
        duration = newDuration;
    }

    public boolean isRed() {
        return color.equalsIgnoreCase("red");
    }

    public boolean isGreen() {
        return color.equalsIgnoreCase("green");
    }
}

public class Main {
```

```
    public static void main(String[] args) {
        TrafficLight trafficLight = new TrafficLight("red", 30);

        System.out.println("Initial light color: " + trafficLight.isRed());
        System.out.println("Is the light green? " + trafficLight.isGreen());

        trafficLight.changeColor("green", 45);

        System.out.println("Changed light color: " + trafficLight.isGreen());
        System.out.println("Is the light red? " + trafficLight.isRed());
    }
}
```

OUTPUT

```
Initial light color: true
Is the light green? false
Changed light color: true
Is the light red? false


...Program finished with exit code 0
Press ENTER to exit console.
```

Q3) Write a Java program to perform arithmetic operations using method overloading.

CODE:

```
class ArithmeticOperations {
    public int add(int a, int b) {
        return a + b;
    }

    public double add(double a, double b) {
        return a + b;
    }

    public int subtract(int a, int b) {
        return a - b;
    }

    public double subtract(double a, double b) {
        return a - b;
    }

    public int multiply(int a, int b) {
        return a * b;
    }

    public double multiply(double a, double b) {
        return a * b;
    }
```

```
    public int divide(int a, int b) {
        if (b != 0) {
            return a / b;
        } else {
            throw new ArithmeticException("Cannot divide by zero");
        }
    }

    public double divide(double a, double b) {
        if (b != 0) {
            return a / b;
        } else {
            throw new ArithmeticException("Cannot divide by zero");
        }
    }
}

public class Main {
    public static void main(String[] args) {
        ArithmeticOperations calculator = new ArithmeticOperations();

        System.out.println("Addition: " + calculator.add(5, 3));
        System.out.println("Subtraction: " + calculator.subtract(10, 4));
        System.out.println("Multiplication: " + calculator.multiply(6, 7));
        System.out.println("Division: " + calculator.divide(20, 4));
    }
}
```

OUTPUT:

```
Addition: 8
Subtraction: 6
Multiplication: 42
Division: 5


...Program finished with exit code 0
Press ENTER to exit console.
```

Q4) Write a Java program to create a class called Employee with methods called work() and getSalary(). Create a subclass called HRManager that overrides the work() method and adds a new method called addEmployee().

CODE:

```
class Employee {
    private String name;
    private double salary;

    public Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }

    public void work() {
        System.out.println(name + " is working.");
    }

    public double getSalary() {
```

```
        return salary;
    }

    public String getName() {
        return name;
    }
}

class HRManager extends Employee {
    public HRManager(String name, double salary) {
        super(name, salary);
    }

    @Override
    public void work() {
        System.out.println(getName() + " (HR Manager) is managing HR
tasks.");
    }

    public void addEmployee(Employee employee) {
        System.out.println("HR Manager is adding " + employee.getName() +
" to the company.");
    }
}

public class Main {
    public static void main(String[] args) {
        Employee employee1 = new Employee("Gaurav", 50000);
        HRManager hrManager = new HRManager("Bob", 75000);

        employee1.work();
```

```
        System.out.println("Employee 1's salary: " + employee1.getSalary());

        hrManager.work();
        System.out.println("HR Manager's salary: " +
hrManager.getSalary());

        Employee employee2 = new Employee("Charlie", 45000);
        hrManager.addEmployee(employee2);
    }
}
```

OUTPUT:

```
Gaurav is working.
Employee 1's salary: 50000.0
Bob (HR Manager) is managing HR tasks.
HR Manager's salary: 75000.0
HR Manager is adding Charlie to the company.


...Program finished with exit code 0
Press ENTER to exit console.
```

Q5) Write a Java program to create a class called Shape with methods called getPerimeter() andgetArea(). Create a subclass called Circle that overrides the getPerimeter() and getArea()methods to calculate the area and perimeter of a circle.

CODE

```
class Shape {
    public double getPerimeter() {
        return 0.0; // Placeholder value, to be overridden by subclasses
    }

    public double getArea() {
        return 0.0; // Placeholder value, to be overridden by subclasses
    }
}

class Circle extends Shape {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    public double getPerimeter() {
        return 2 * Math.PI * radius;
    }

    @Override
```

```
    public double getArea() {
        return Math.PI * radius * radius;
    }
}

public class Main {
    public static void main(String[] args) {
        Circle circle = new Circle(5.0);

        System.out.println("Circle Perimeter: " + circle.getPerimeter());
        System.out.println("Circle Area: " + circle.getArea());
    }
}
```

OUTPUT:

```
Circle Perimeter: 31.415926535897793
Circle Area: 78.53981633974483


...Program finished with exit code 0
Press ENTER to exit console.
```

APP