Experiment - 10

Interfaces

LEARNING OUTCOMES:

Student will be able to:

1. Define an Interface
2. Implement the Interface
3. Define Interface Methods

OBJECTIVE:

To be able to write a Java program to implement an interface and provide definitions for the interface methods.

REQUIRED APPARATUS:

1. Notepad/Editors (VS Code), JDK 1.7
2. Personal Computer with 2GB RAM, 320GB HDD, and Pentium2 processor or above

PRECAUTIONS AND SAFETY MEASURES FOR A COMPUTER LAB:

1. Don't touch the switchboards with wet hands.
2. Don't operate a system if the walls are wet.
3. Keep food and beverages outside the workspace.
4. Shutdown and switch off the systems properly to avoid system crashes.
5. Keep footwear outside the lab to protect equipment from dust.
6. Know the place of the fire extinguisher in the lab.

BRIEF THEORY:

In Java, an interface is an abstract class that contains a collection of methods and constant variables. It is one of the core concepts in Java used to achieve abstraction, polymorphism, and multiple inheritance.

Syntax:

```
interface TestInterface {
    // declare constant variables
    // declare abstract methods
}
```

Interfaces in Java are implemented using the 'implements' keyword with classes to inherit the properties of an interface.

Example Program to Create an Interface and Provide Definitions for Interface Methods:

```
// Define the ArithmeticOperations interface
interface ArithmeticOperations {
    int add(int a, int b);
    int subtract(int a, int b);
    int multiply(int a, int b);
    double divide(int a, int b);
}

// Implement the ArithmeticOperations interface in a class
class Calculator implements ArithmeticOperations {
    public int add(int a, int b) {
        return a + b;
    }
    public int subtract(int a, int b) {
        return a - b;
    }
    public int multiply(int a, int b) {
        return a * b;
    }
    public double divide(int a, int b) {
        if (b == 0) {
            throw new ArithmeticException("Division by zero is not allowed.");
        }
        return (double) a / b;
    }
}

// Main class to use the Calculator
public class ArithmeticOperationsExample {
    public static void main(String[] args) {
        Calculator calculator = new Calculator();
        int a = 10, b = 20;

        System.out.println("Addition: " + calculator.add(a, b));
        System.out.println("Subtraction: " + calculator.subtract(a, b));
        System.out.println("Multiplication: " + calculator.multiply(a, b));
        System.out.println("Division: " + calculator.divide(a, b));
    }
}
```

ACTIVITY:

1. Create an interface called 'Student' and provide definitions for all methods declared in the Student interface.
   (Note: Define the methods as instructed by your lab instructor.)

Sample Viva Questions:

1. What is an interface in Java?
2. What happens if definitions are not given to methods in an interface?
3. Which inheritance type is implemented using interfaces?
4. Can we create an object of an interface? Yes/No. Give a reason.
5. Can an interface contain method implementations?
6. What is a static method in an interface?
7. Can an interface extend another interface?