NAME:-VAIBHAV SHARMA

ROLL NO:-2300290119008

## EXPERIMENT NO:-4

In Java, you can achieve abstraction by using abstract classes.

~~~~
```java
// Abstract class
abstract class Animal {
    // Abstract method (does not have a body)
    public abstract void makeSound();

    // Regular method
    public void sleep() {
        System.out.println("Zzz...");
    }
}

// Subclass (inherits from Animal)
class Dog extends Animal {
    // Implement the abstract method
    public void makeSound() {
~~~~

```
        System.out.println("Woof!");
    }
}

// Another subclass (inherits from Animal)
class Cat extends Animal {
    // Implement the abstract method
    public void makeSound() {
        System.out.println("Meow!");
    }
}

public class Main {
    public static void main(String[] args) {
        Dog myDog = new Dog();
        Cat myCat = new Cat();

        myDog.makeSound();
        myDog.sleep();

        myCat.makeSound();
        myCat.sleep();
```

~~~
    }
}
```
~~~

- Implement Abstraction by using interface. in java

```java
// Define an interface called 'Animal'
interface Animal {
    // Abstract method (does not have a body)
    void makeSound();

    // Abstract method
    void move();
}

// Implement the 'Animal' interface in a class called 'Dog'
class Dog implements Animal {
    // Provide implementation for the makeSound method
    public void makeSound() {
        System.out.println("Woof");
    }

    // Provide implementation for the move method
```

```
    public void move() {
        System.out.println("The dog runs");
    }
}

// Implement the 'Animal' interface in a class called 'Cat'
class Cat implements Animal {
    // Provide implementation for the makeSound method
    public void makeSound() {
        System.out.println("Meow");
    }

    // Provide implementation for the move method
    public void move() {
        System.out.println("The cat jumps");
    }
}

public class Main {
    public static void main(String[] args) {
        // Create an instance of Dog
        Animal myDog = new Dog();
        myDog.makeSound(); // Outputs: Woof
        myDog.move(); // Outputs: The dog runs

        // Create an instance of Cat
        Animal myCat = new Cat();
        myCat.makeSound(); // Outputs: Meow
        myCat.move(); // Outputs: The cat jumps
    }
}
```

- Implement Multiple Inheritance by using interface. in java

```
// Define the first interface
interface InterfaceA {
    void methodA();
}

// Define the second interface
interface InterfaceB {
    void methodB();
}

// Implement both interfaces in a single class
public class MultipleInheritanceExample implements
InterfaceA, InterfaceB {
    // Implement method from InterfaceA
    @Override
    public void methodA() {
        System.out.println("Method A from InterfaceA");
    }

    // Implement method from InterfaceB
    @Override
    public void methodB() {
        System.out.println("Method B from InterfaceB");
    }

    public static void main(String[] args) {
        MultipleInheritanceExample example = new
MultipleInheritanceExample();
        example.methodA();
        example.methodB();
    }
}
```