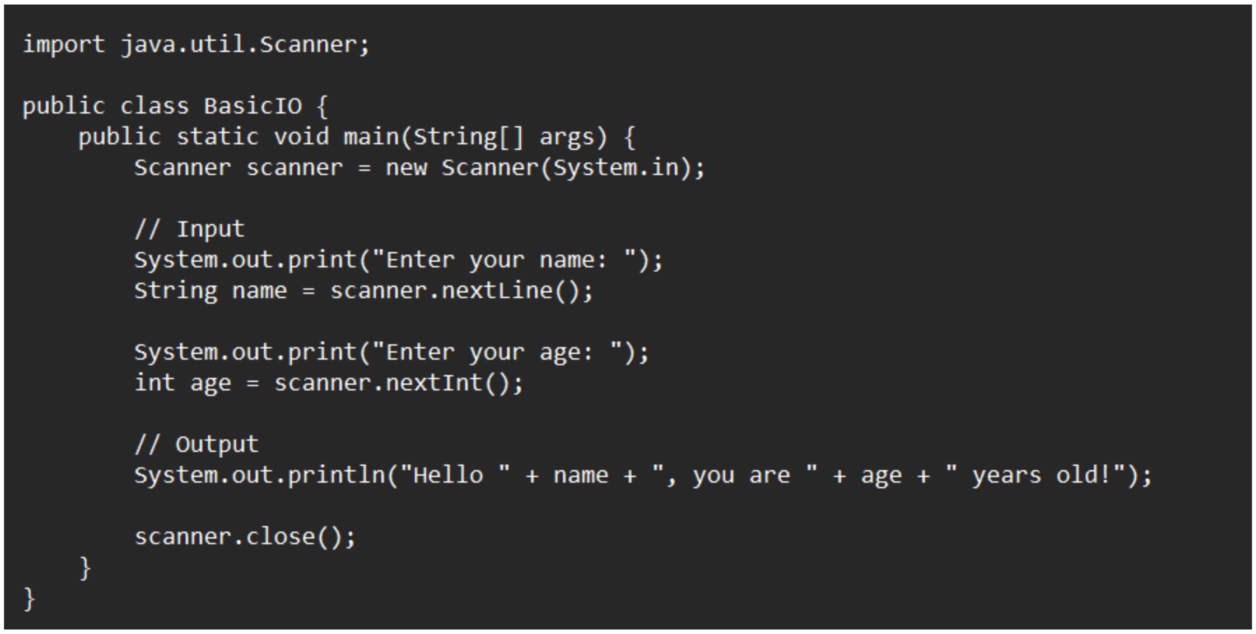
EXERCISE-01

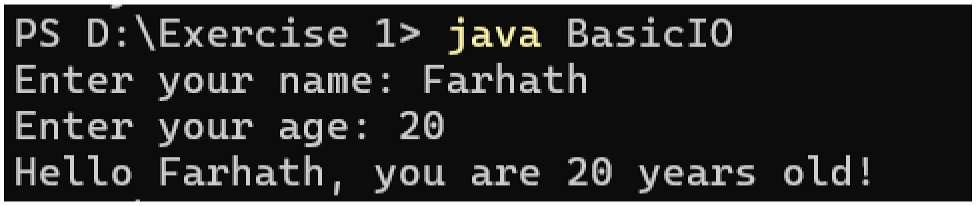
Basic Input and Output Statements

Program:



```
import java.util.Scanner;

public class BasicIO {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input
        System.out.print("Enter your name: ");
        String name = scanner.nextLine();

        System.out.print("Enter your age: ");
        int age = scanner.nextInt();

        // Output
        System.out.println("Hello " + name + ", you are " + age + " years old!");

        scanner.close();
    }
}
```

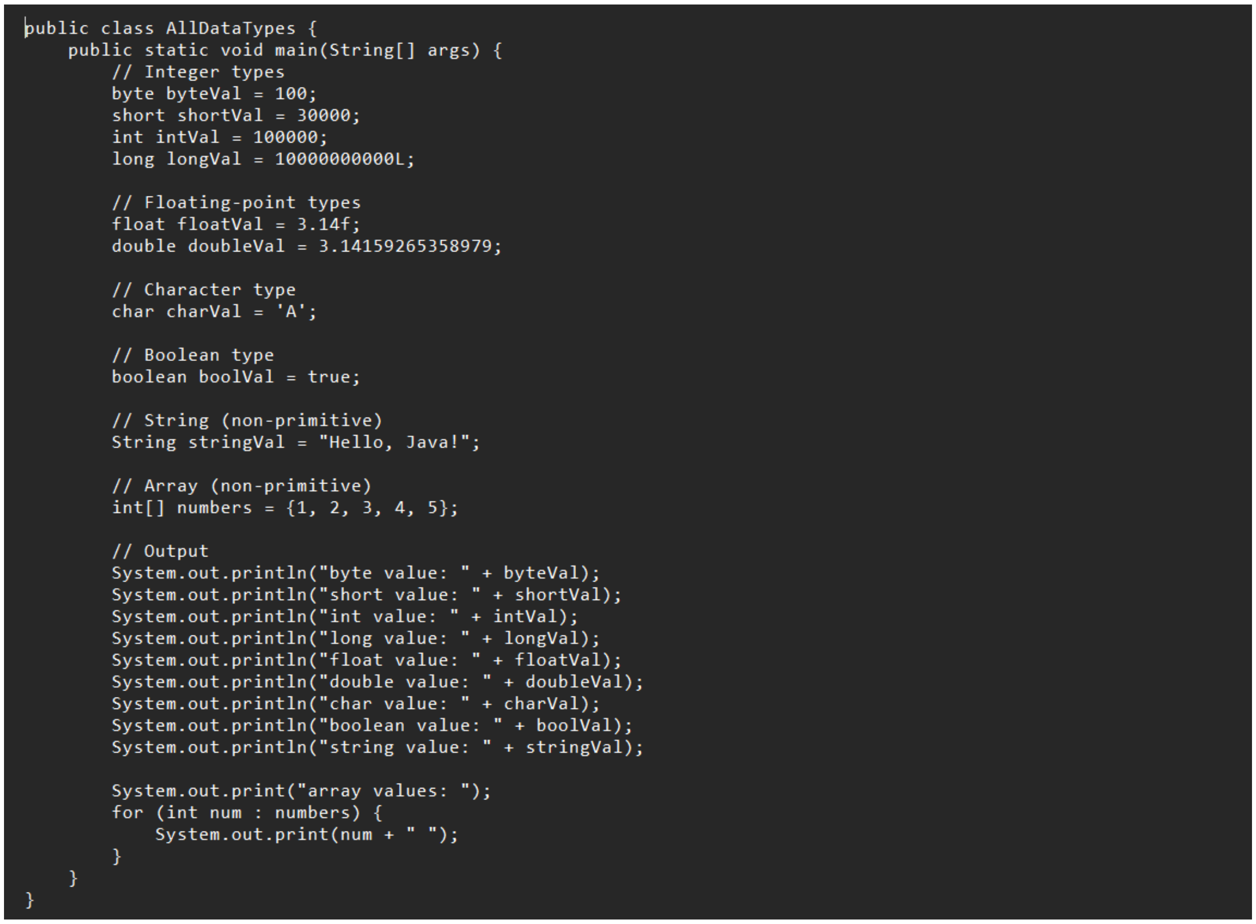
Output:



```
PS D:\Exercise 1> java BasicIO
Enter your name: Farhath
Enter your age: 20
Hello Farhath, you are 20 years old!
```

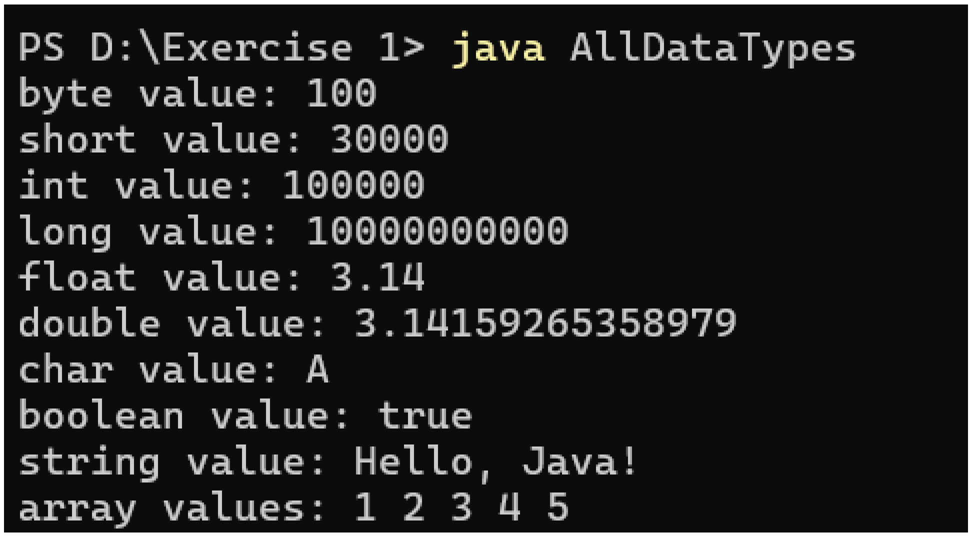
Data Types

Program:



```
public class AllDataTypes {
    public static void main(String[] args) {
        // Integer types
        byte byteVal = 100;
        short shortVal = 30000;
        int intVal = 100000;
        long longVal = 10000000000L;

        // Floating-point types
        float floatVal = 3.14f;
        double doubleVal = 3.14159265358979;

        // Character type
        char charVal = 'A';

        // Boolean type
        boolean boolVal = true;

        // String (non-primitive)
        String stringVal = "Hello, Java!";

        // Array (non-primitive)
        int[] numbers = {1, 2, 3, 4, 5};

        // Output
        System.out.println("byte value: " + byteVal);
        System.out.println("short value: " + shortVal);
        System.out.println("int value: " + intVal);
        System.out.println("long value: " + longVal);
        System.out.println("float value: " + floatVal);
        System.out.println("double value: " + doubleVal);
        System.out.println("char value: " + charVal);
        System.out.println("boolean value: " + boolVal);
        System.out.println("string value: " + stringVal);

        System.out.print("array values: ");
        for (int num : numbers) {
            System.out.print(num + " ");
        }
    }
}
```

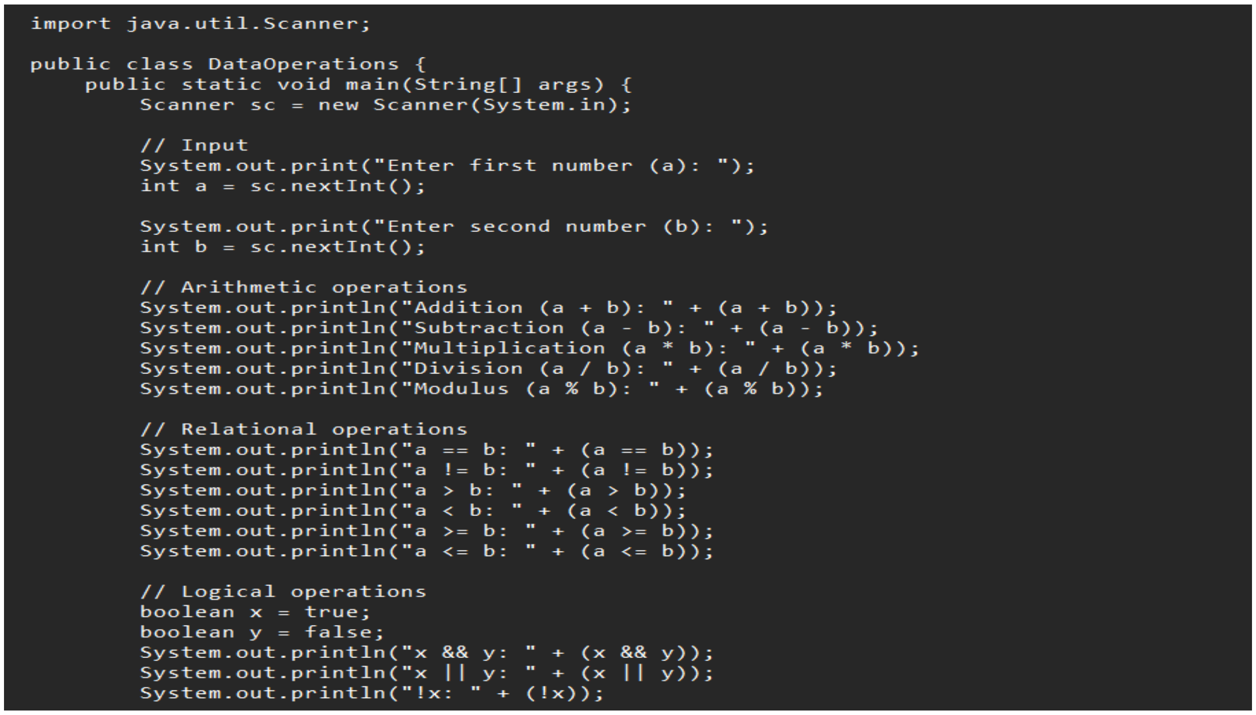
Output:



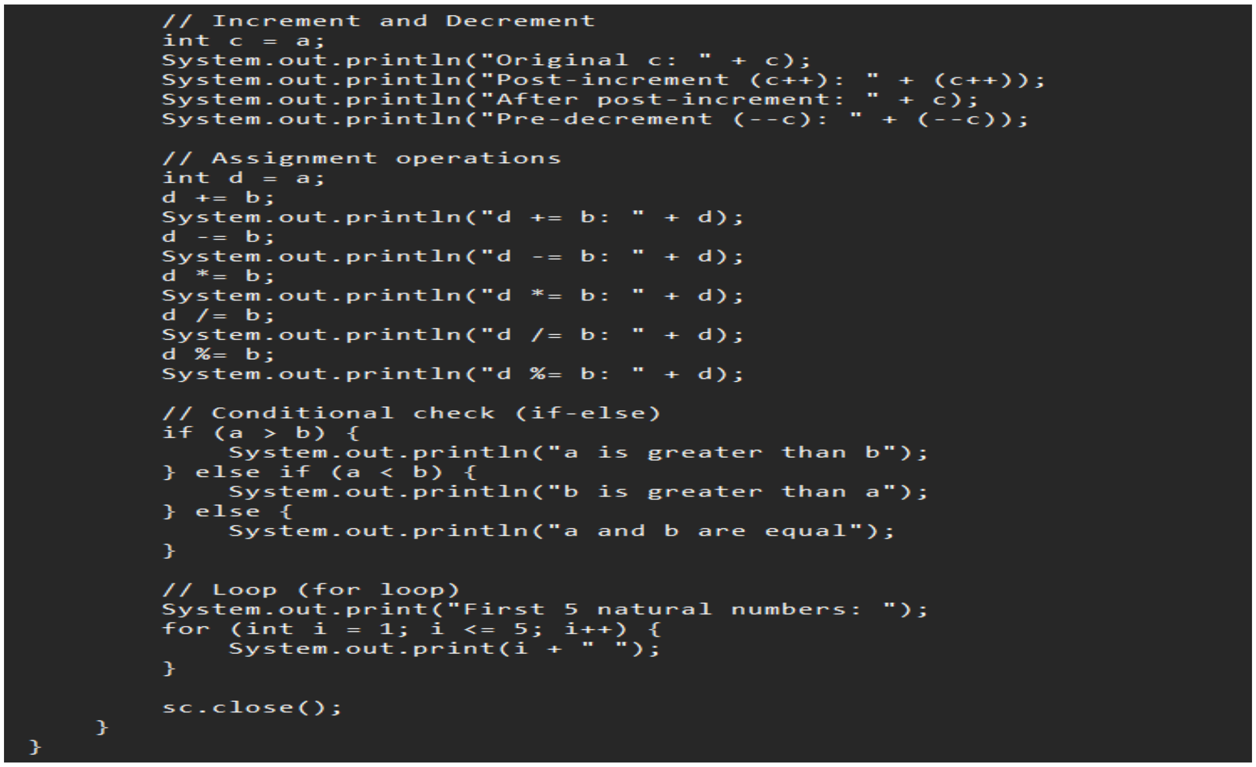
```
PS D:\Exercise 1> java AllDataTypes
byte value: 100
short value: 30000
int value: 100000
long value: 10000000000
float value: 3.14
double value: 3.14159265358979
char value: A
boolean value: true
string value: Hello, Java!
array values: 1 2 3 4 5
```

Data Operations

Program:

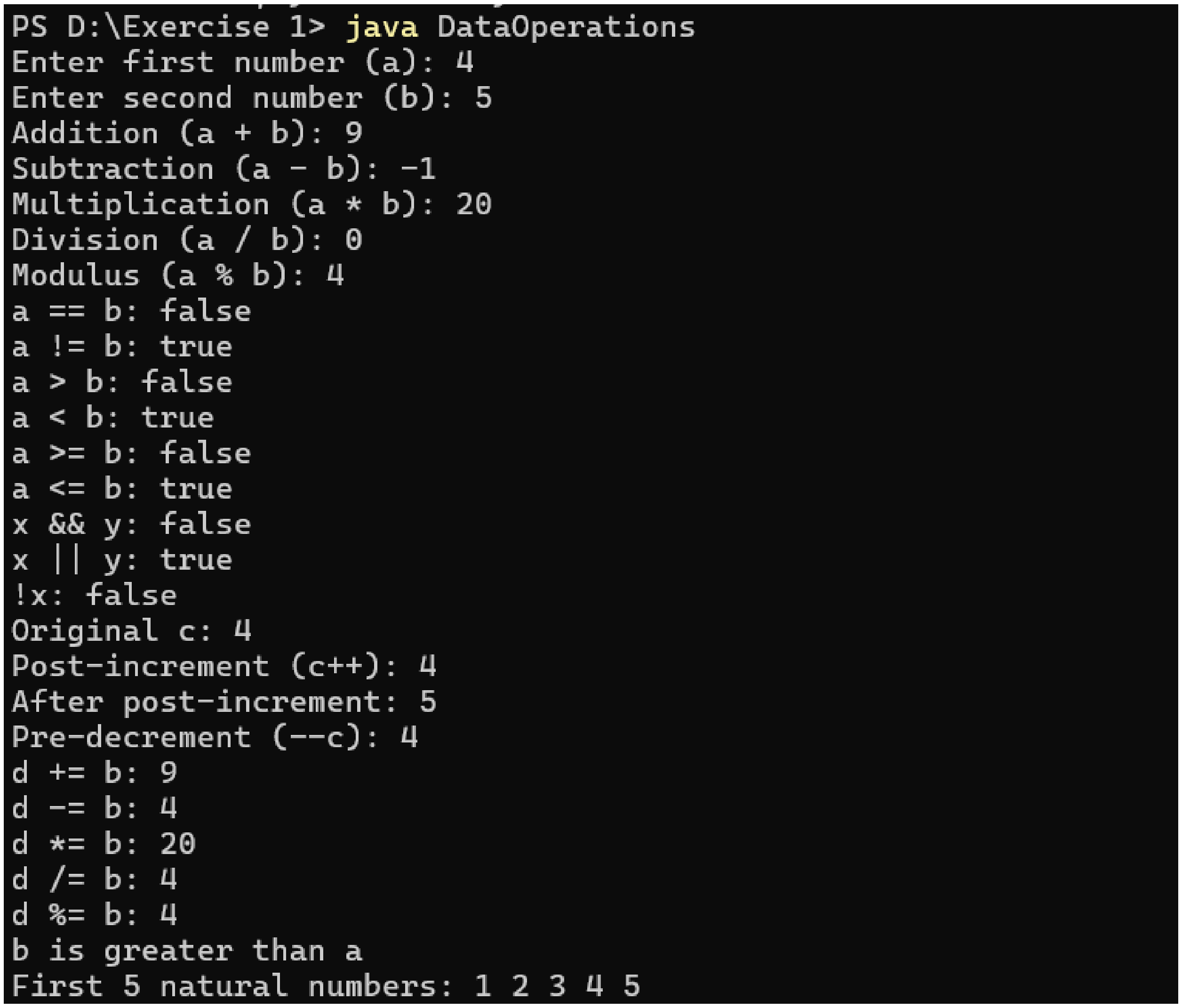


```
import java.util.Scanner;

public class DataOperations {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Input
        System.out.print("Enter first number (a): ");
        int a = sc.nextInt();

        System.out.print("Enter second number (b): ");
        int b = sc.nextInt();

        // Arithmetic operations
        System.out.println("Addition (a + b): " + (a + b));
        System.out.println("Subtraction (a - b): " + (a - b));
        System.out.println("Multiplication (a * b): " + (a * b));
        System.out.println("Division (a / b): " + (a / b));
        System.out.println("Modulus (a % b): " + (a % b));

        // Relational operations
        System.out.println("a == b: " + (a == b));
        System.out.println("a != b: " + (a != b));
        System.out.println("a > b: " + (a > b));
        System.out.println("a < b: " + (a < b));
        System.out.println("a >= b: " + (a >= b));
        System.out.println("a <= b: " + (a <= b));

        // Logical operations
        boolean x = true;
        boolean y = false;
        System.out.println("x && y: " + (x && y));
        System.out.println("x || y: " + (x || y));
        System.out.println("!x: " + (!x));
```



```
        // Increment and Decrement
        int c = a;
        System.out.println("Original c: " + c);
        System.out.println("Post-increment (c++): " + (c++));
        System.out.println("After post-increment: " + c);
        System.out.println("Pre-decrement (--c): " + (--c));

        // Assignment operations
        int d = a;
        d += b;
        System.out.println("d += b: " + d);
        d -= b;
        System.out.println("d -= b: " + d);
        d *= b;
        System.out.println("d *= b: " + d);
        d /= b;
        System.out.println("d /= b: " + d);
        d %= b;
        System.out.println("d %= b: " + d);

        // Conditional check (if-else)
        if (a > b) {
            System.out.println("a is greater than b");
        } else if (a < b) {
            System.out.println("b is greater than a");
        } else {
            System.out.println("a and b are equal");
        }

        // Loop (for loop)
        System.out.print("First 5 natural numbers: ");
        for (int i = 1; i <= 5; i++) {
            System.out.print(i + " ");
        }

        sc.close();
    }
}
```

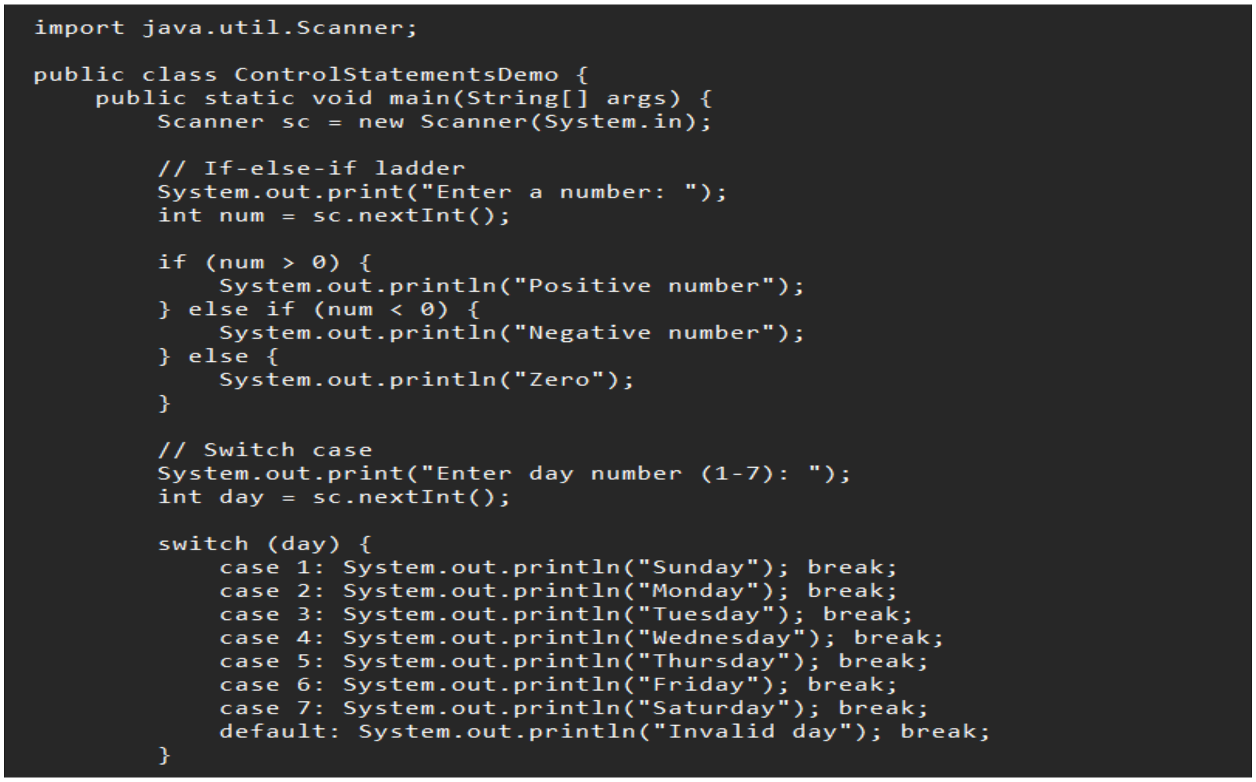
Output:



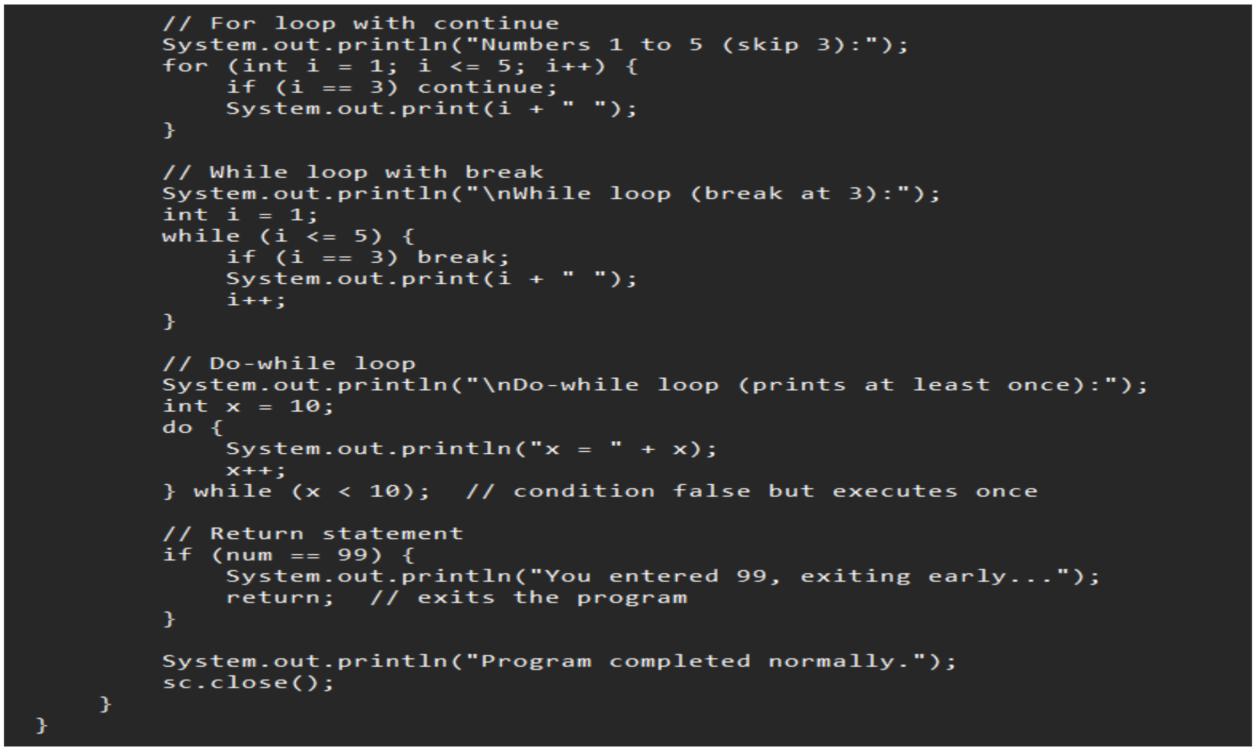
```
PS D:\Exercise 1> java DataOperations
Enter first number (a): 4
Enter second number (b): 5
Addition (a + b): 9
Subtraction (a - b): -1
Multiplication (a * b): 20
Division (a / b): 0
Modulus (a % b): 4
a == b: false
a != b: true
a > b: false
a < b: true
a >= b: false
a <= b: true
x && y: false
x || y: true
!x: false
Original c: 4
Post-increment (c++): 4
After post-increment: 5
Pre-decrement (--c): 4
d += b: 9
d -= b: 4
d *= b: 20
d /= b: 4
d %= b: 4
b is greater than a
First 5 natural numbers: 1 2 3 4 5
```

Control Statements

Program:

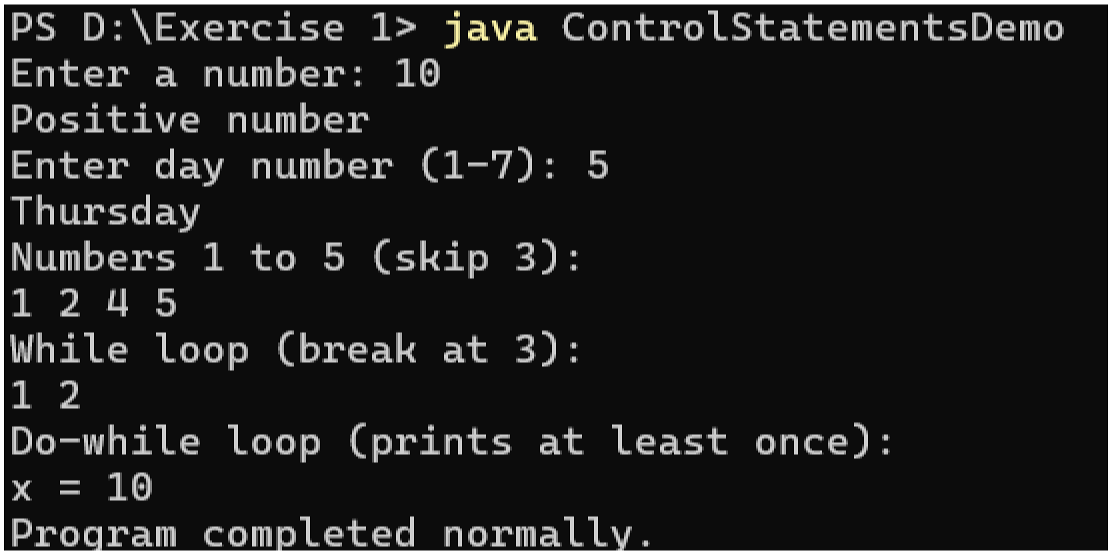


```java
import java.util.Scanner;

public class ControlStatementsDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // If-else-if ladder
        System.out.print("Enter a number: ");
        int num = sc.nextInt();

        if (num > 0) {
            System.out.println("Positive number");
        } else if (num < 0) {
            System.out.println("Negative number");
        } else {
            System.out.println("Zero");
        }

        // Switch case
        System.out.print("Enter day number (1-7): ");
        int day = sc.nextInt();

        switch (day) {
            case 1: System.out.println("Sunday"); break;
            case 2: System.out.println("Monday"); break;
            case 3: System.out.println("Tuesday"); break;
            case 4: System.out.println("Wednesday"); break;
            case 5: System.out.println("Thursday"); break;
            case 6: System.out.println("Friday"); break;
            case 7: System.out.println("Saturday"); break;
            default: System.out.println("Invalid day"); break;
        }
```



```java
        // For loop with continue
        System.out.println("Numbers 1 to 5 (skip 3):");
        for (int i = 1; i <= 5; i++) {
            if (i == 3) continue;
            System.out.print(i + " ");
        }

        // While loop with break
        System.out.println("\nWhile loop (break at 3):");
        int i = 1;
        while (i <= 5) {
            if (i == 3) break;
            System.out.print(i + " ");
            i++;
        }

        // Do-while loop
        System.out.println("\nDo-while loop (prints at least once):");
        int x = 10;
        do {
            System.out.println("x = " + x);
            x++;
        } while (x < 10);  // condition false but executes once

        // Return statement
        if (num == 99) {
            System.out.println("You entered 99, exiting early...");
            return;  // exits the program
        }

        System.out.println("Program completed normally.");
        sc.close();
    }
}
```

Output:



```
PS D:\Exercise 1> java ControlStatementsDemo
Enter a number: 10
Positive number
Enter day number (1-7): 5
Thursday
Numbers 1 to 5 (skip 3):
1 2 4 5
While loop (break at 3):
1 2
Do-while loop (prints at least once):
x = 10
Program completed normally.
```

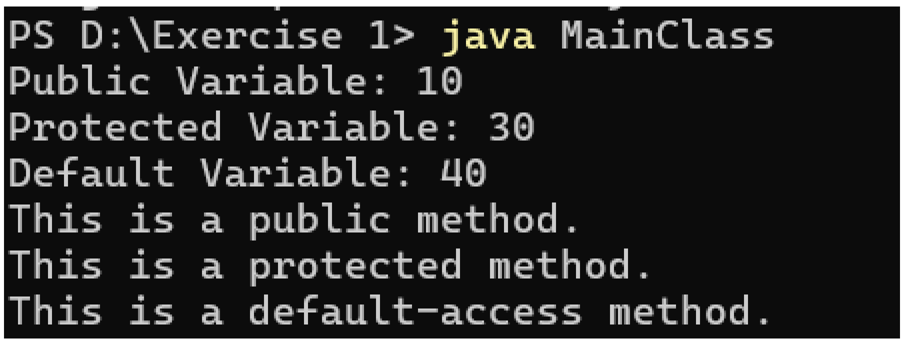
Access Modifiers

Program:

```
public class AccessModifiers {

    public int publicVar = 10;
    private int privateVar = 20;
    protected int protectedVar = 30;
    int defaultVar = 40; // default access

    public void publicMethod() {
        System.out.println("This is a public method.");
    }

    private void privateMethod() {
        System.out.println("This is a private method.");
    }

    protected void protectedMethod() {
        System.out.println("This is a protected method.");
    }

    void defaultMethod() {
        System.out.println("This is a default-access method.");
    }
}
```

```
public class MainClass {
    public static void main(String[] args) {
        AccessModifiers obj = new AccessModifiers();

        // Accessing members with different modifiers
        System.out.println("Public Variable: " + obj.publicVar);
        // System.out.println("Private Variable: " + obj.privateVar);
        System.out.println("Protected Variable: " + obj.protectedVar);
        System.out.println("Default Variable: " + obj.defaultVar);

        // Calling methods
        obj.publicMethod();
        // obj.privateMethod();
        obj.protectedMethod();
        obj.defaultMethod();
    }
}
```

Output:



```
PS D:\Exercise 1> java MainClass
Public Variable: 10
Protected Variable: 30
Default Variable: 40
This is a public method.
This is a protected method.
This is a default-access method.
```

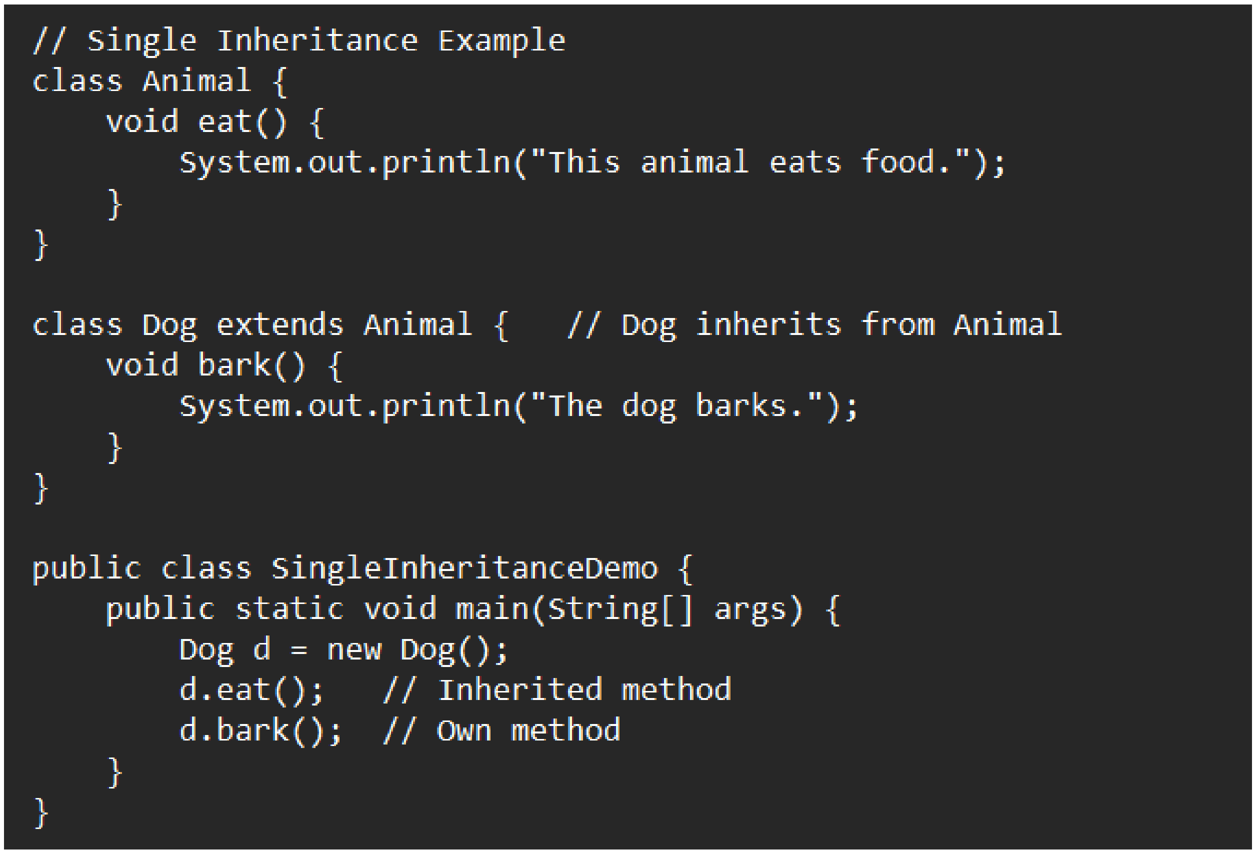
Exercise-02

NAME: AJAY K

ROLL NO: CH.EN.U4CCE22003

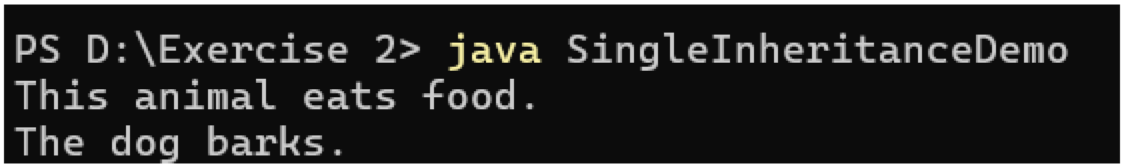
Single Inheritance

Program:



```java
// Single Inheritance Example
class Animal {
    void eat() {
        System.out.println("This animal eats food.");
    }
}

class Dog extends Animal {   // Dog inherits from Animal
    void bark() {
        System.out.println("The dog barks.");
    }
}

public class SingleInheritanceDemo {
    public static void main(String[] args) {
        Dog d = new Dog();
        d.eat();   // Inherited method
        d.bark();  // Own method
    }
}
```

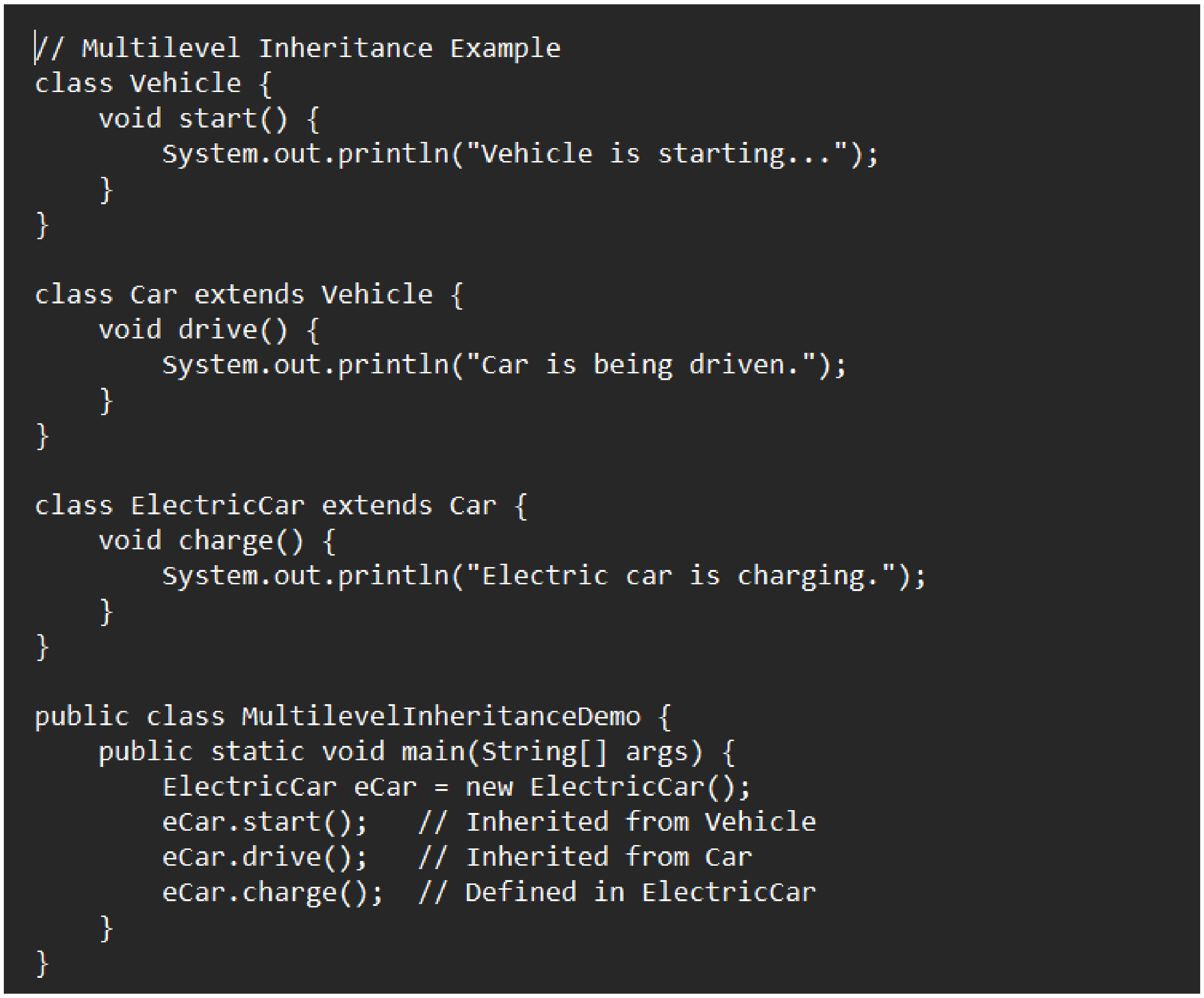
Output:



```
PS D:\Exercise 2> java SingleInheritanceDemo
This animal eats food.
The dog barks.
```

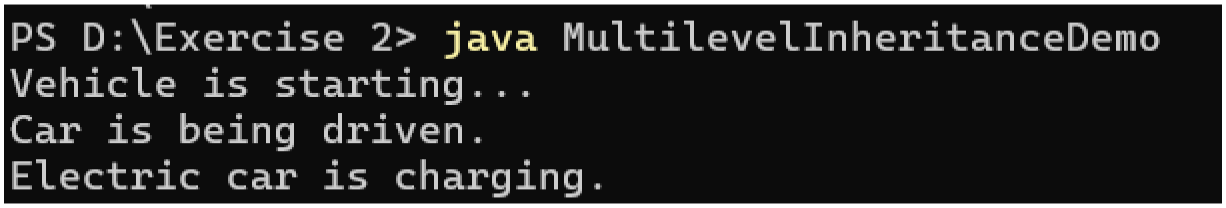
Multilevel Inheritance

Program:



```java
// Multilevel Inheritance Example
class Vehicle {
    void start() {
        System.out.println("Vehicle is starting...");
    }
}

class Car extends Vehicle {
    void drive() {
        System.out.println("Car is being driven.");
    }
}

class ElectricCar extends Car {
    void charge() {
        System.out.println("Electric car is charging.");
    }
}

public class MultilevelInheritanceDemo {
    public static void main(String[] args) {
        ElectricCar eCar = new ElectricCar();
        eCar.start();   // Inherited from Vehicle
        eCar.drive();   // Inherited from Car
        eCar.charge();  // Defined in ElectricCar
    }
}
```

Output:



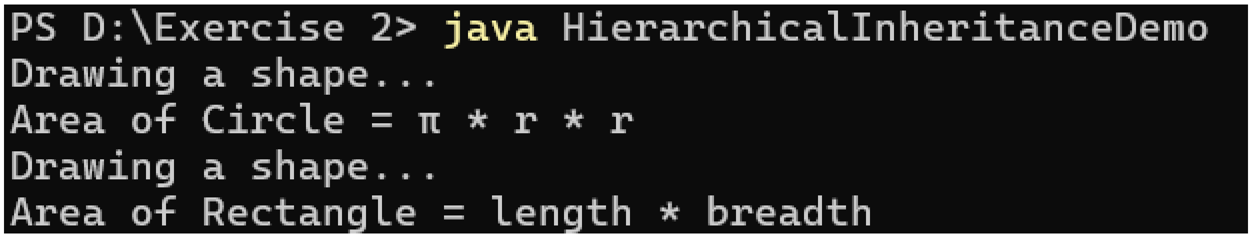
```
PS D:\Exercise 2> java MultilevelInheritanceDemo
Vehicle is starting...
Car is being driven.
Electric car is charging.
```

Hierarchical Inheritance

Program:

```
// Hierarchical Inheritance Example
class Shape {
    void draw() {
        System.out.println("Drawing a shape...");
    }
}

class Circle extends Shape {
    void area() {
        System.out.println("Area of Circle = π * r * r");
    }
}

class Rectangle extends Shape {
    void area() {
        System.out.println("Area of Rectangle = length * breadth");
    }
}

public class HierarchicalInheritanceDemo {
    public static void main(String[] args) {
        Circle c = new Circle();
        c.draw();   // Inherited method
        c.area();   // Circle's own method

        Rectangle r = new Rectangle();
        r.draw();   // Inherited method
        r.area();   // Rectangle's own method
    }
}
```

Output:



```
PS D:\Exercise 2> java HierarchicalInheritanceDemo
Drawing a shape...
Area of Circle = π * r * r
Drawing a shape...
Area of Rectangle = length * breadth
```

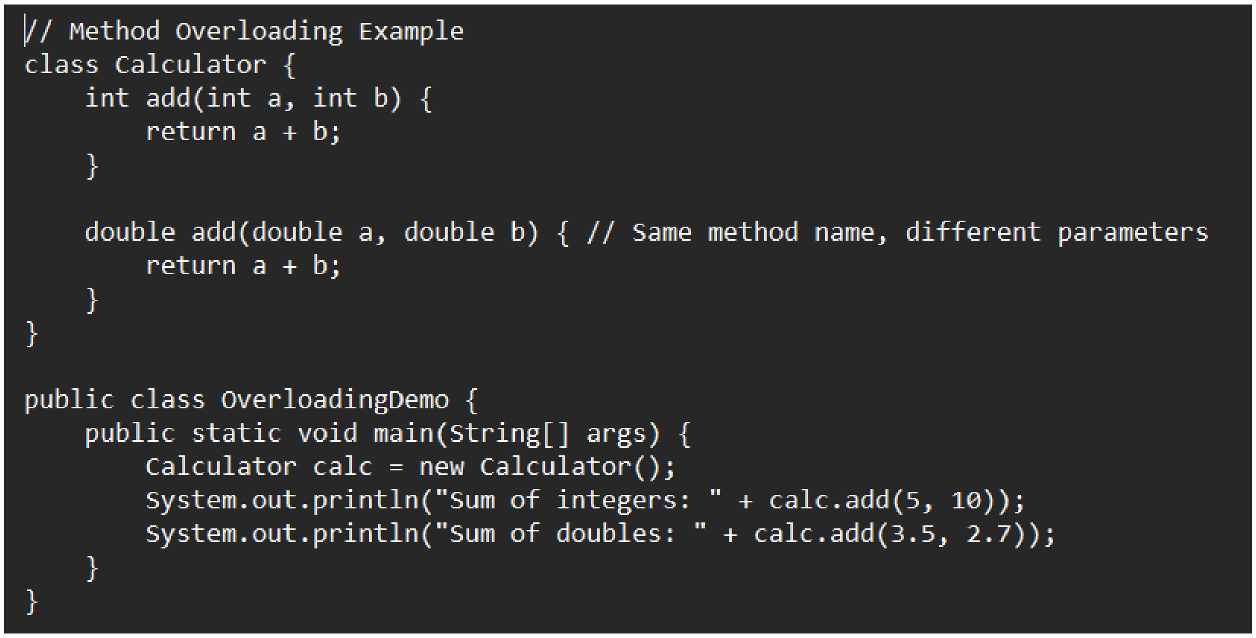
EXERCISE-03

NAME: AJAY K

ROLL NO: CH.EN.U4CCE22050

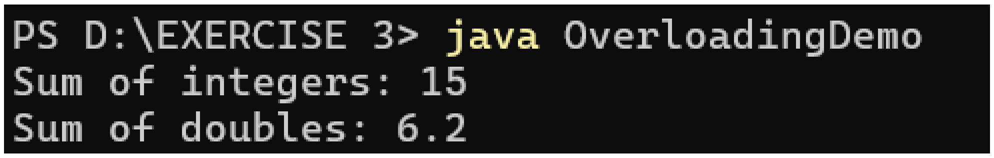
Compile-time Polymorphism (Method Overloading)

Program:



```java
// Method Overloading Example
class Calculator {
    int add(int a, int b) {
        return a + b;
    }

    double add(double a, double b) { // Same method name, different parameters
        return a + b;
    }
}

public class OverloadingDemo {
    public static void main(String[] args) {
        Calculator calc = new Calculator();
        System.out.println("Sum of integers: " + calc.add(5, 10));
        System.out.println("Sum of doubles: " + calc.add(3.5, 2.7));
    }
}
```

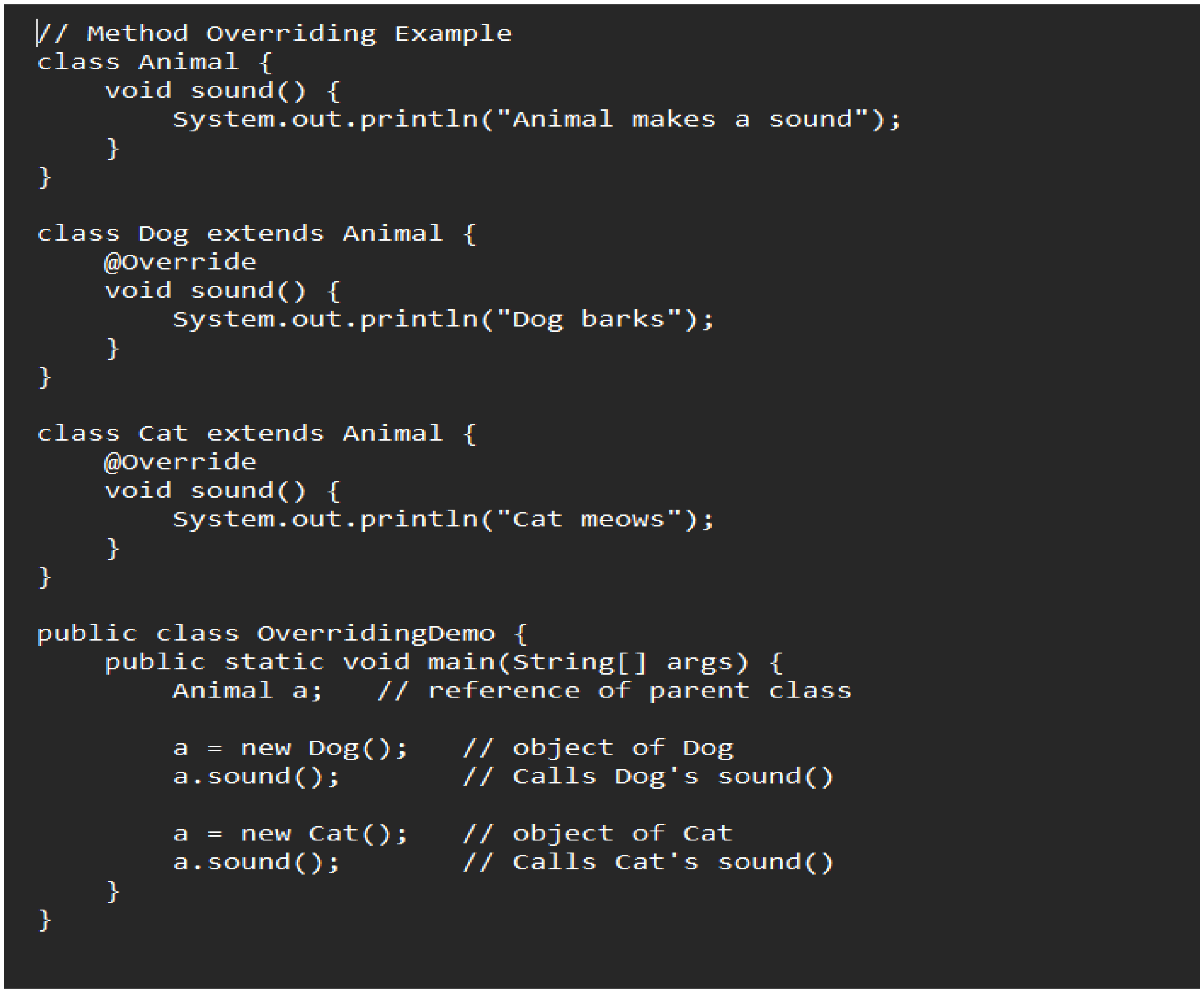
Output:



```
PS D:\EXERCISE 3> java OverloadingDemo
Sum of integers: 15
Sum of doubles: 6.2
```

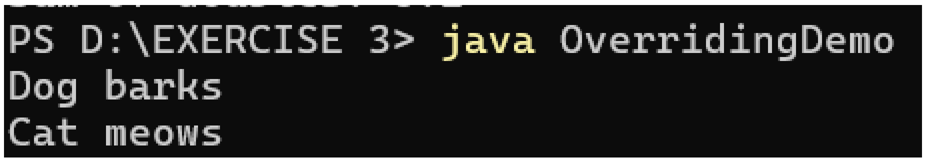
Runtime Polymorphism (Method Overriding)

Program:



```java
// Method Overriding Example
class Animal {
    void sound() {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {
    @Override
    void sound() {
        System.out.println("Dog barks");
    }
}

class Cat extends Animal {
    @Override
    void sound() {
        System.out.println("Cat meows");
    }
}

public class OverridingDemo {
    public static void main(String[] args) {
        Animal a;   // reference of parent class

        a = new Dog();    // object of Dog
        a.sound();        // Calls Dog's sound()

        a = new Cat();    // object of Cat
        a.sound();        // Calls Cat's sound()
    }
}
```

Output:



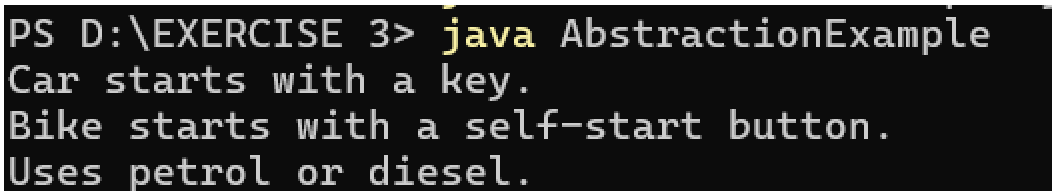
```
PS D:\EXERCISE 3> java OverridingDemo
Dog barks
Cat meows
```

Abstraction

Program:

```java
// Abstraction Example in Java
abstract class Vehicle {
    abstract void start(); // abstract method (no implementation)

    void fuelType() {
        System.out.println("Uses petrol or diesel.");
    }
}

class Car extends Vehicle {
    void start() {
        System.out.println("Car starts with a key.");
    }
}

class Bike extends Vehicle {
    void start() {
        System.out.println("Bike starts with a self-start button.");
    }
}

public class AbstractionExample {
    public static void main(String[] args) {
        Vehicle v1 = new Car();
        Vehicle v2 = new Bike();

        v1.start();
        v2.start();
        v1.fuelType();
    }
}
```

Output:



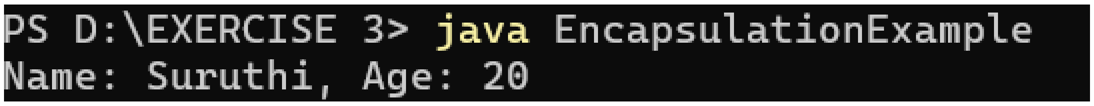
```
PS D:\EXERCISE 3> java AbstractionExample
Car starts with a key.
Bike starts with a self-start button.
Uses petrol or diesel.
```

Encapsulation

Program:

```java
// Encapsulation Example in Java
class Student {
    private String name;  // data is hidden
    private int age;

    // getter and setter methods control access
    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setAge(int age) {
        if(age > 0) this.age = age; // validation example
        else System.out.println("Invalid age!");
    }

    public int getAge() {
        return age;
    }
}

public class EncapsulationExample {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.setName("Suruthi");
        s1.setAge(20);

        System.out.println("Name: " + s1.getName() + ", Age: " + s1.getAge());
    }
}
```

Output:



```
PS D:\EXERCISE 3> java EncapsulationExample
Name: Suruthi, Age: 20
```