# JAVA

## LAB # 04

Instructor: Dr. Faheem Akhtar Rajput

Submitted by: Madan Lal

**Department of Computer Science**

Semester VI

**SIBAU**

CMS ID: 053-18-0005

Sukkur IBA University, Nisar Ahmed Siddique Road, Sukkur

# Lab Tasks

## Exercise 1 CalculateBMI.java

Write a Java application with the following prototypes that returns the user's body mass index (BMI)

**public static double calcluateBMI(double weight, double height)**

To calculate BMI based on weight in pounds (lb) and height in inches (in), use this formula:

$$\text{BMI} = \frac{\text{mass(lb)}}{(\text{height(in)})^2} \times 703$$

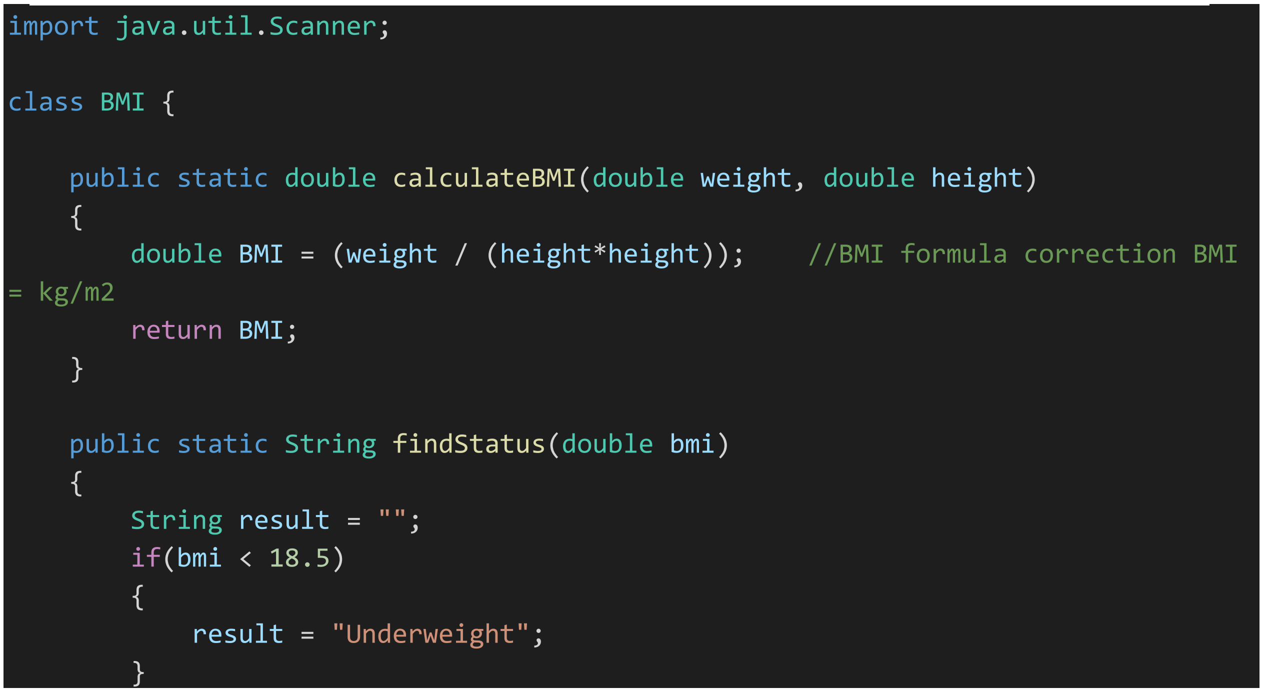
and

**public static String findStatus(double bmi)**

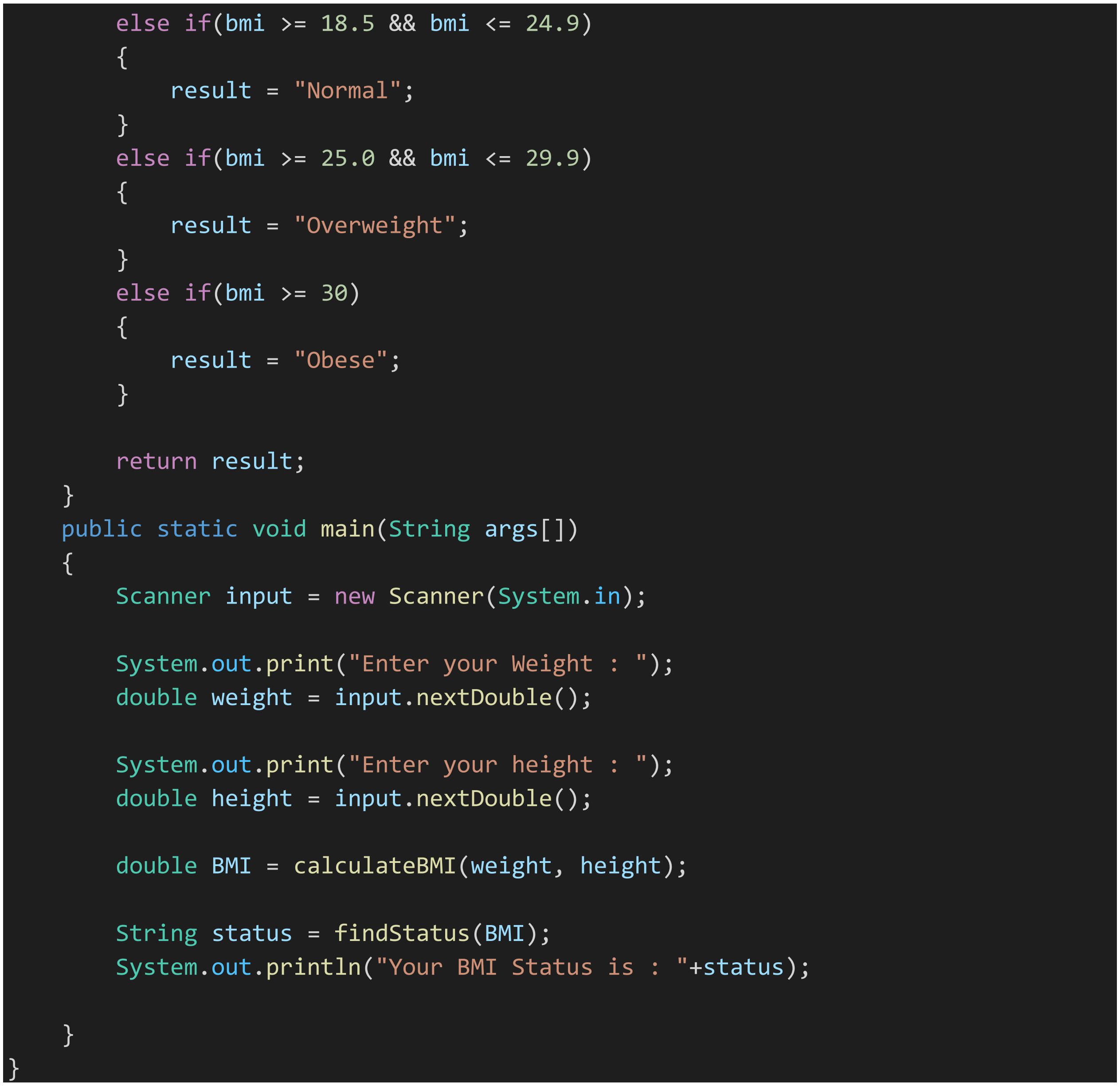
Categorizes it as underweight, normal, overweight, or obese, based on the table from the United States Centers for Disease Control:

| **BMI** | **Weight Status** |
|---|---|
| Below 18.5 | Underweight |
| 18.5 – 24.9 | Normal |
| 25.0-29.9 | Overweight |
| 30.0 and above | Obese |

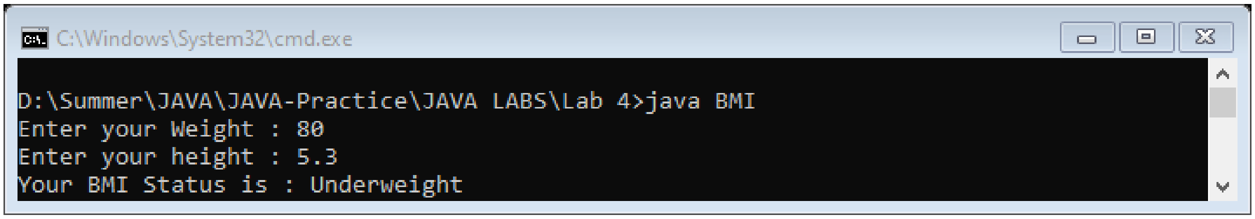
Prompt the user to enter weight in pounds and height in inches.



```
import java.util.Scanner;

class BMI {

    public static double calculateBMI(double weight, double height)
    {
        double BMI = (weight / (height*height));    //BMI formula correction BMI
= kg/m2
        return BMI;
    }

    public static String findStatus(double bmi)
    {
        String result = "";
        if(bmi < 18.5)
        {
            result = "Underweight";
        }
```



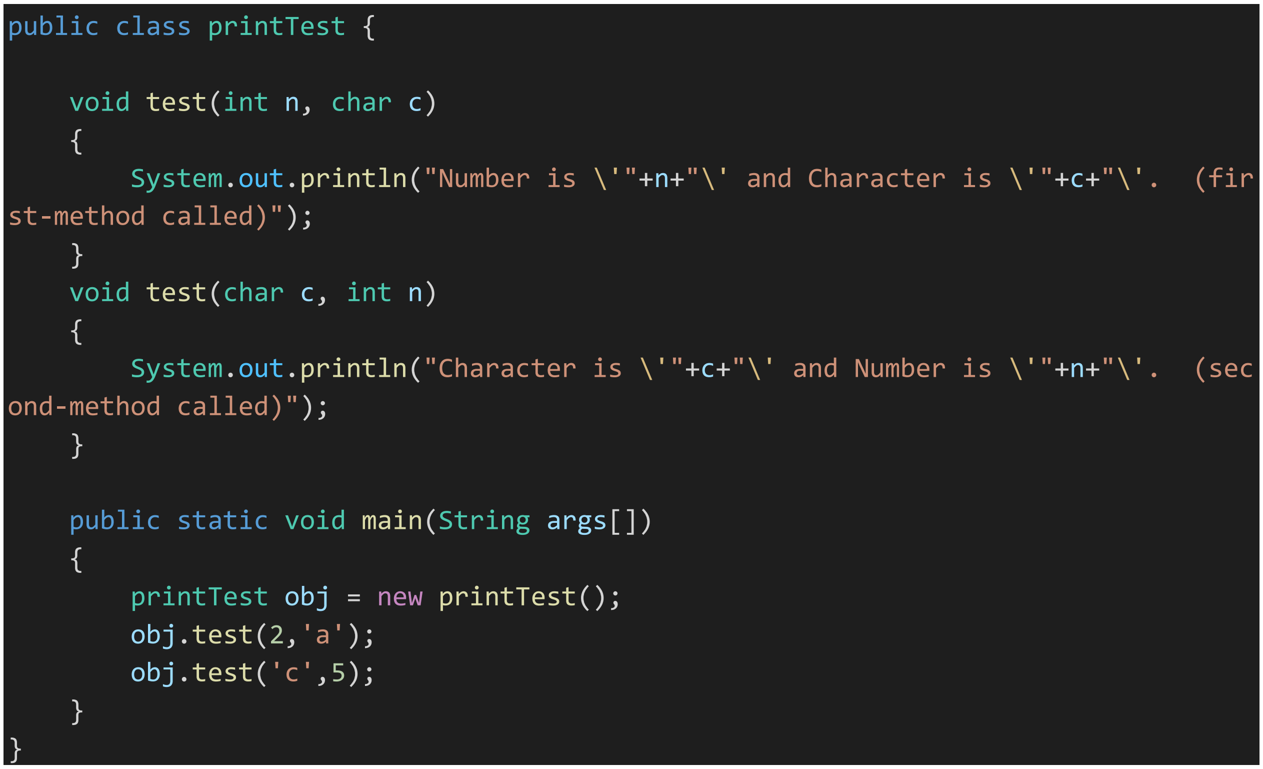
```java
        else if(bmi >= 18.5 && bmi <= 24.9)
        {
            result = "Normal";
        }
        else if(bmi >= 25.0 && bmi <= 29.9)
        {
            result = "Overweight";
        }
        else if(bmi >= 30)
        {
            result = "Obese";
        }

        return result;
    }
    public static void main(String args[])
    {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter your Weight : ");
        double weight = input.nextDouble();

        System.out.print("Enter your height : ");
        double height = input.nextDouble();

        double BMI = calculateBMI(weight, height);

        String status = findStatus(BMI);
        System.out.println("Your BMI Status is : "+status);

    }
}
```



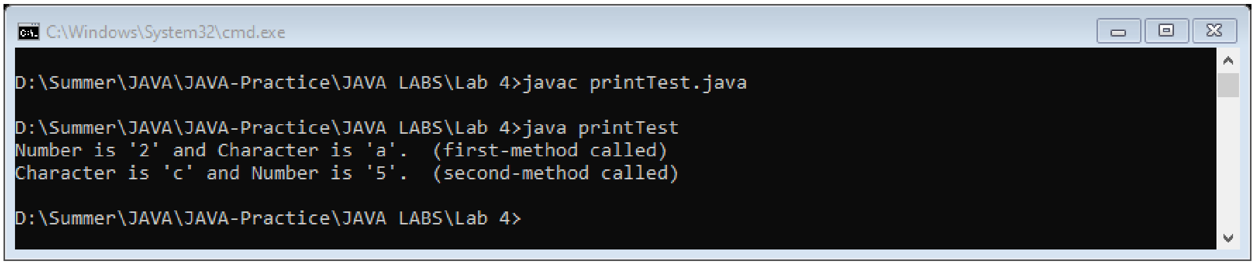
```
D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>java BMI
Enter your Weight : 80
Enter your height : 5.3
Your BMI Status is : Underweight
```

## Exercise 2 (Method Overloading) PrintTest.java

Create a class to print an integer and a character with two methods having the same name but different sequence of the integer and the character parameters. For example, if the parameters of the first method are of the form (int n, char c), then that of the second method will be of the form (char c, int n).



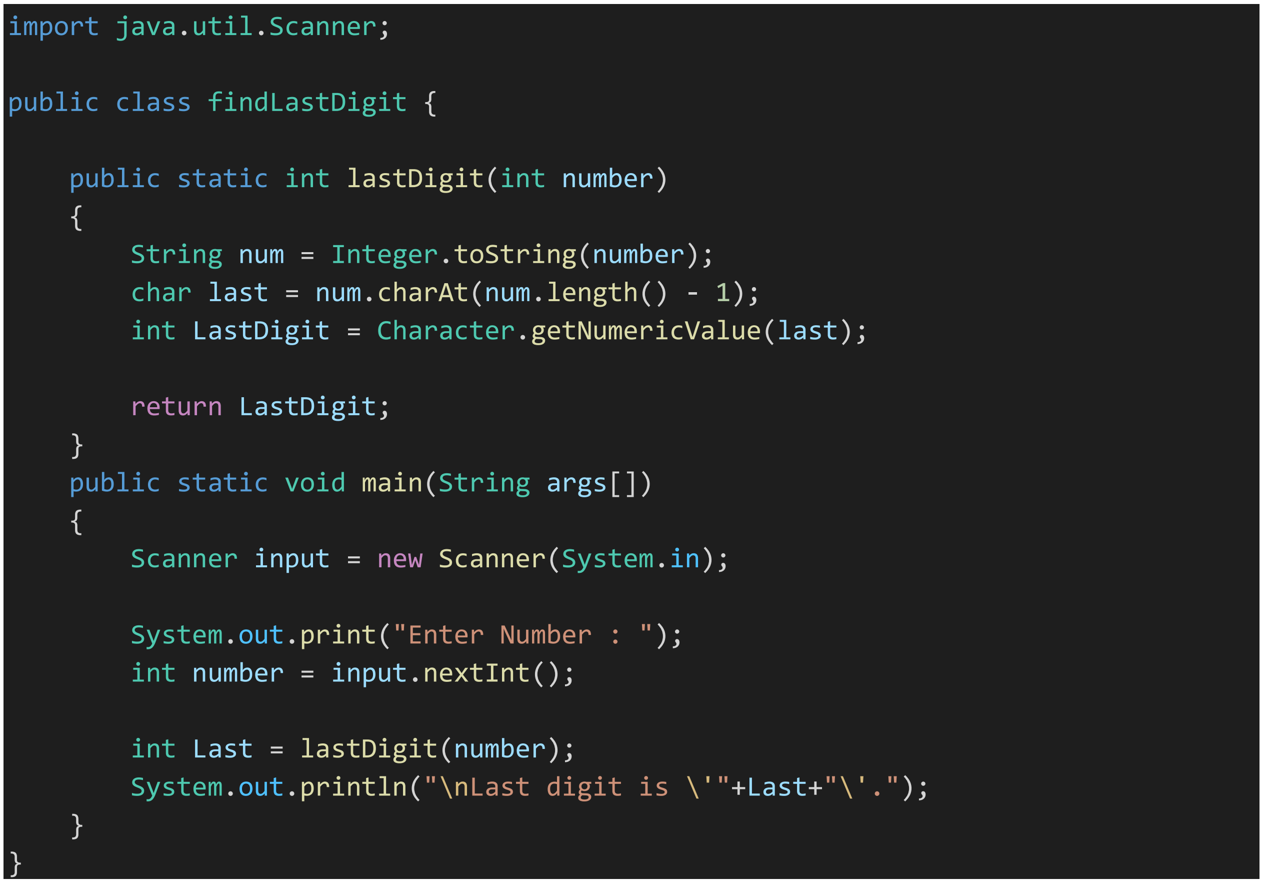
```java
public class printTest {

    void test(int n, char c)
    {
        System.out.println("Number is \'"+n+"\' and Character is \'"+c+"\'.  (first-method called)");
    }
    void test(char c, int n)
    {
        System.out.println("Character is \'"+c+"\' and Number is \'"+n+"\'.  (second-method called)");
    }

    public static void main(String args[])
    {
        printTest obj = new printTest();
        obj.test(2,'a');
        obj.test('c',5);
    }
}
```



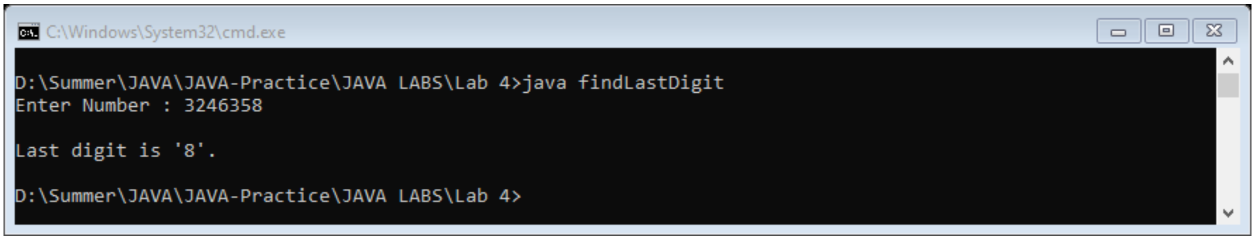
```
D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>javac printTest.java

D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>java printTest
Number is '2' and Character is 'a'.  (first-method called)
Character is 'c' and Number is '5'.  (second-method called)

D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>
```

## Exercise 3 (Static Method) FindLastDigit.java

Write a static method named lastDigit that returns the last digit of an integer. For example, lastDigit(3852) should return 2



```java
import java.util.Scanner;

public class findLastDigit {

    public static int lastDigit(int number)
    {
        String num = Integer.toString(number);
        char last = num.charAt(num.length() - 1);
        int LastDigit = Character.getNumericValue(last);

        return LastDigit;
    }
    public static void main(String args[])
    {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter Number : ");
        int number = input.nextInt();

        int Last = lastDigit(number);
        System.out.println("\nLast digit is \'"+Last+"\'.");
    }
}
```

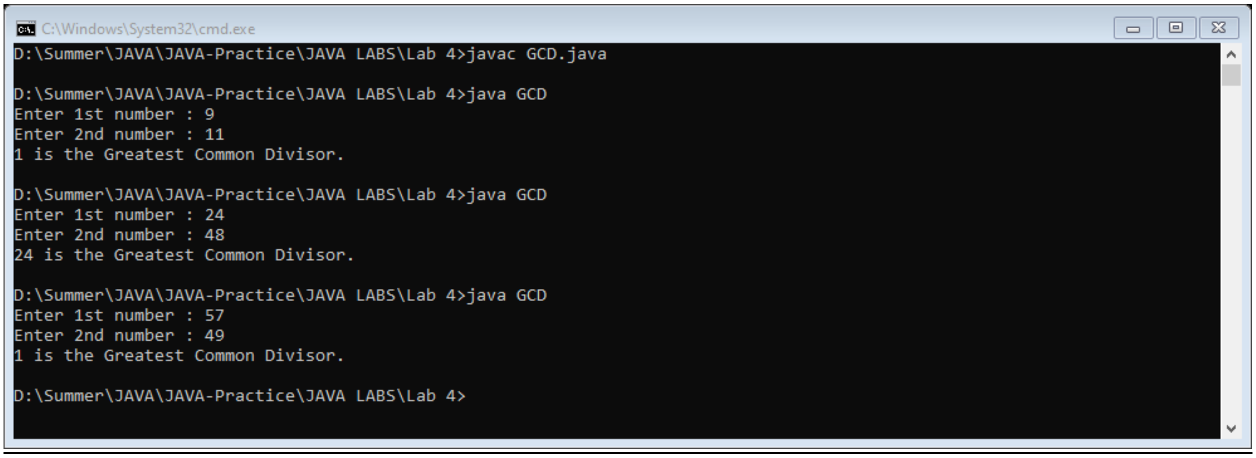


```
D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>java findLastDigit
Enter Number : 3246358

Last digit is '8'.

D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>
```

## Exercise 4 (Recursion) GCDCalculation.java

Find Greatest Common Divisor (GCD) of 2 numbers using recursion.

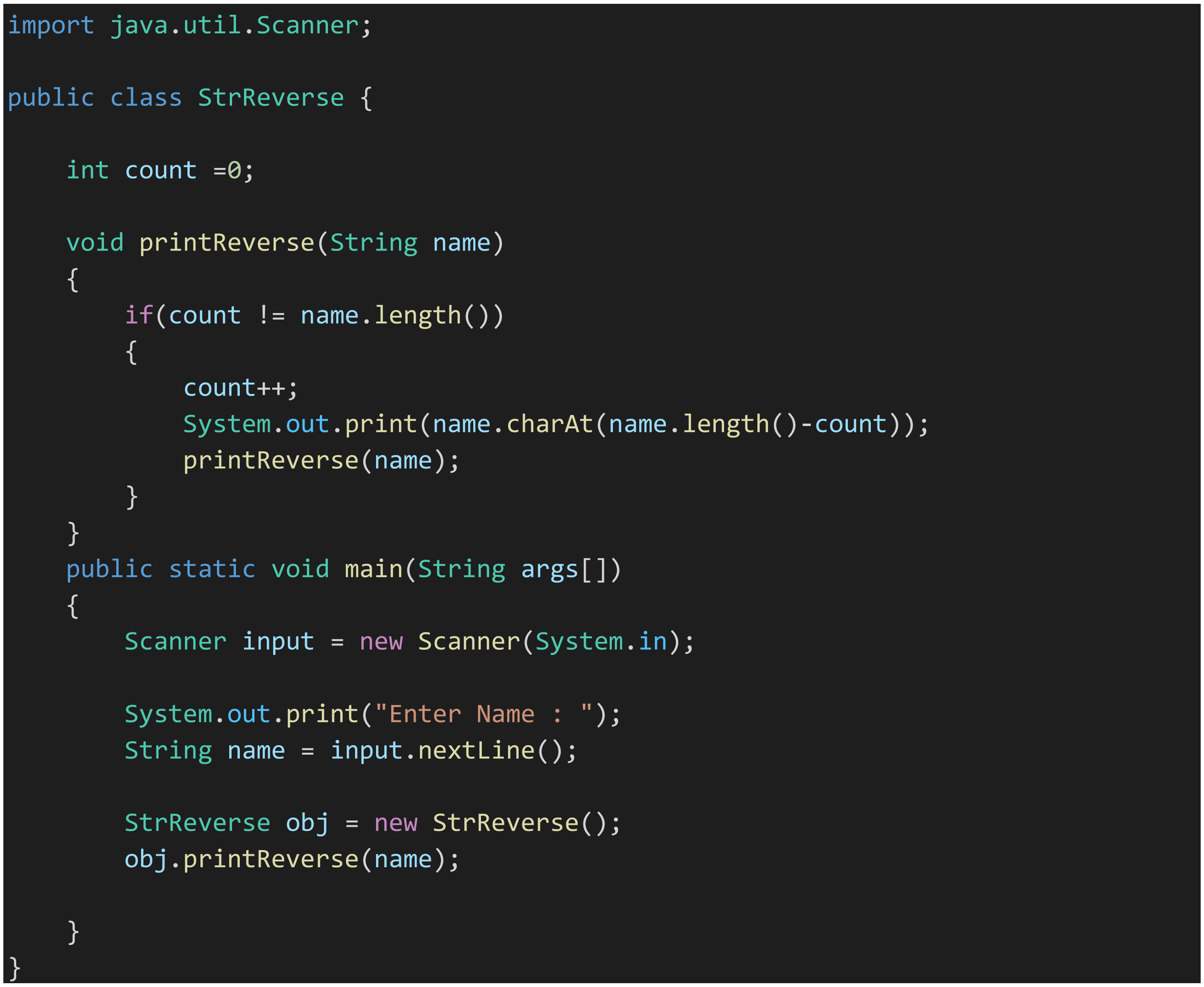
```java
import java.util.Scanner;

public class GCD {

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

        System.out.print("Enter 1st number : ");
        int n1 = input.nextInt();

        System.out.print("Enter 2nd number : ");
        int n2 = input.nextInt();

        GCD obj = new GCD();
        int Result = obj.gcd(n1,n2);
        System.out.println(Result+" is the Greatest Common Divisor.");
    }

    int gcd(int n1, int n2)
    {
        if (n2 != 0)
            return gcd(n2, n1 % n2);
        else
            return n1;
    }

}
```



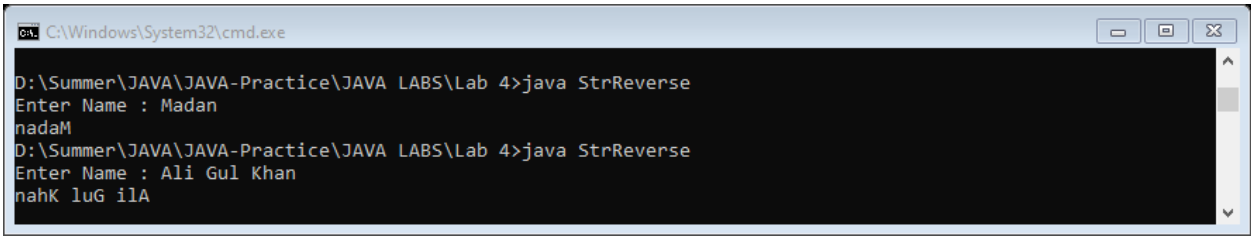
```
D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>javac GCD.java

D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>java GCD
Enter 1st number : 9
Enter 2nd number : 11
1 is the Greatest Common Divisor.

D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>java GCD
Enter 1st number : 24
Enter 2nd number : 48
24 is the Greatest Common Divisor.

D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>java GCD
Enter 1st number : 57
Enter 2nd number : 49
1 is the Greatest Common Divisor.

D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>
```

## Exercise 5 (Recursion) StrReverse.java

Write a recursive function that, given a string s= “OOP is Fun”, print the characters of s in reverse order



```java
import java.util.Scanner;

public class StrReverse {

    int count =0;

    void printReverse(String name)
    {
        if(count != name.length())
        {
            count++;
            System.out.print(name.charAt(name.length()-count));
            printReverse(name);
        }
    }
    public static void main(String args[])
    {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter Name : ");
        String name = input.nextLine();

        StrReverse obj = new StrReverse();
        obj.printReverse(name);

    }
}
```



```
D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>java StrReverse
Enter Name : Madan
nadaM
D:\Summer\JAVA\JAVA-Practice\JAVA LABS\Lab 4>java StrReverse
Enter Name : Ali Gul Khan
nahK luG ilA
```

**1. Can constructors be static in java? Try it out and justify.**

No, we cannot define a static constructor in Java, If we are trying to define a constructor with the static keyword a compile-time error will occur. ... A constructor will be used to assign initial values for the instance variables. Both static and constructor are different and opposite to each other.

**2. Why use iterations when we have recursion and vice versa?**

Recursion allows you to allocate additional automatic objects at each function call. The iterative alternative is to repeatedly dynamically allocate or resize memory blocks.

**End of Lab 4**