Batch:B4 Roll No.:16010122221
Experiment / assignment / tutorial No.01
Grade: AA / AB / BB / BC / CC / CD /DD
Signature of the Staff In-charge with date

TITLE : Perfect Number

AIM: Define a class Perfect which accepts the range of numbers from the user. Create a static function check_per , which checks if the number is a perfect number or not and sends the result back to the main function which counts and displays the perfect numbers within that range.

Variations :

Implementation of Program with One class

Accessibility with static and non-static methods within class and outside class.

Expected OUTCOME of Experiment:

CO2: Explore arrays, vectors, classes and objects in C++ and Java

Books/ Journals/ Websites referred:

1. E. Balagurusamy , “Programming with Java” McGraw-Hill.

2. Sachin Malhotra, Saurabh Choudhary, “Programming in Java”, Oxford Publications.

Pre Lab/ Prior Concepts:

The Scanner class is a class in java.util, which allows the user to read values of various types. There are far more methods in class Scanner than you will need in this course. We only cover a small useful subset, ones that allow us to read in numeric values from either the keyboard or file without having to convert them from strings and determine if there are more values to be read.

Scanner in = new Scanner(System.in); // System.in is an InputStream
Numeric and String Methods

Method	Returns
int nextInt()	Returns the next token as an int. If the next token is not an integer,InputMismatchException is thrown.
long nextLong()	Returns the next token as a long. If the next token is not an integer,InputMismatchException is thrown.
float nextFloat()	Returns the next token as a float. If the next token is not a float or is out of range, InputMismatchException is thrown.
double nextDouble()	Returns the next token as a long. If the next token is not a float or is out of range, InputMismatchException is thrown.
String next()	Finds and returns the next complete token from this scanner and returns it as a string; a token is usually ended by whitespace such as a blank or line break. If not token exists,NoSuchElementException is thrown.
String nextLine()	Returns the rest of the current line, excluding any line separator at the end.
void close()	Closes the scanner.

The Scanner looks for tokens in the input. A token is a series of characters that ends with what Java calls whitespace. A whitespace character can be a blank, a tab character, a carriage return. Thus, if we read a line that has a series of numbers separated by blanks, the scanner will take each number as a separate token.
The numeric values may all be on one line with blanks between each value or may be on separate lines.   Whitespace characters (blanks or carriage returns) act as separators. The next method returns the next input value as a string, regardless of what is keyed. For example, given the following code segment and data

- int number = in.nextInt();
- float real = in.nextFloat();
- long number2 = in.nextLong();
- double real2 = in.nextDouble();
- String string = in.next();

Class Diagram:

CLASS	**METHOD**
Perfect	int calc (int n)
Main	public static void main (string arg sc)

Perfect		Main
-n : int		-a, x : int
-sum : int	←	-s : perfect
-i : int		
+ calc (n:int):int		+ calc(a): int

Algorithm:

1. Define a class named "Perfect".

2. Inside the "Perfect" class:

2.1 Define instance variables "start" and "end" to store the range of numbers.

2.2 Define a constructor that accepts "start" and "end" as parameters and initializes the instance variables.

2.3 Define a static method named "check_per" that accepts a number as a parameter and checks if it is a perfect number:

2.3.1 Initialize a variable "sumDivisors" to 0.

2.3.2 Loop from 1 to (number / 2):

2.3.2.1 If the current number divides "number" evenly, add it to "sumDivisors".

2.3.3 If "sumDivisors" is equal to "number", return true (perfect number), otherwise, return false.

2.4 Define a non-static method named "countAndDisplayPerfectNumbers" to count and display perfect numbers within the range:

2.4.1 Initialize a variable "perfectCount" to 0.

2.4.2 Loop through the range of numbers:

2.4.2.1 Call the static method "check_per" with the current number to check if it's a perfect number.

2.4.2.2 If the number is perfect, increment "perfectCount" and display it.

2.4.3 Display the total count of perfect numbers.

3. In the main part of the code:

3.1 Create a scanner object to get user input.

3.2 Prompt the user to enter the start and end of the range.

3.3 Read the input for "startRange" and "endRange".

3.4 Create an instance of the "Perfect" class with the provided range.

3.5 Call the "countAndDisplayPerfectNumbers" method on the instance to display perfect numbers.

3.6 Close the scanner.

Implementation details:

variation-1

```
import java.util.Scanner;


class Perfect {

    public static boolean check_per(int number) {

        int sumDivisors = 0;

        for (int i = 1; i <= number / 2; i++) {

            if (number % i == 0) {

                sumDivisors += i;

            }
        }
        return sumDivisors == number;
    }

    public void countAndDisplayPerfectNumbers(int start, int end) {
        int perfectCount = 0;

        for (int num = start; num <= end; num++) {
            if (check_per(num)) {
                System.out.println(num + " is a perfect number.");
                perfectCount++;
            }
        }

        System.out.println("Total perfect numbers in the range: " + perfectCount);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter start of range: ");
        int startRange = scanner.nextInt();
        System.out.print("Enter end of range: ");
        int endRange = scanner.nextInt();

        Perfect perfect = new Perfect();
        perfect.countAndDisplayPerfectNumbers(startRange, endRange);

        scanner.close();
    }
}
```

variation-2

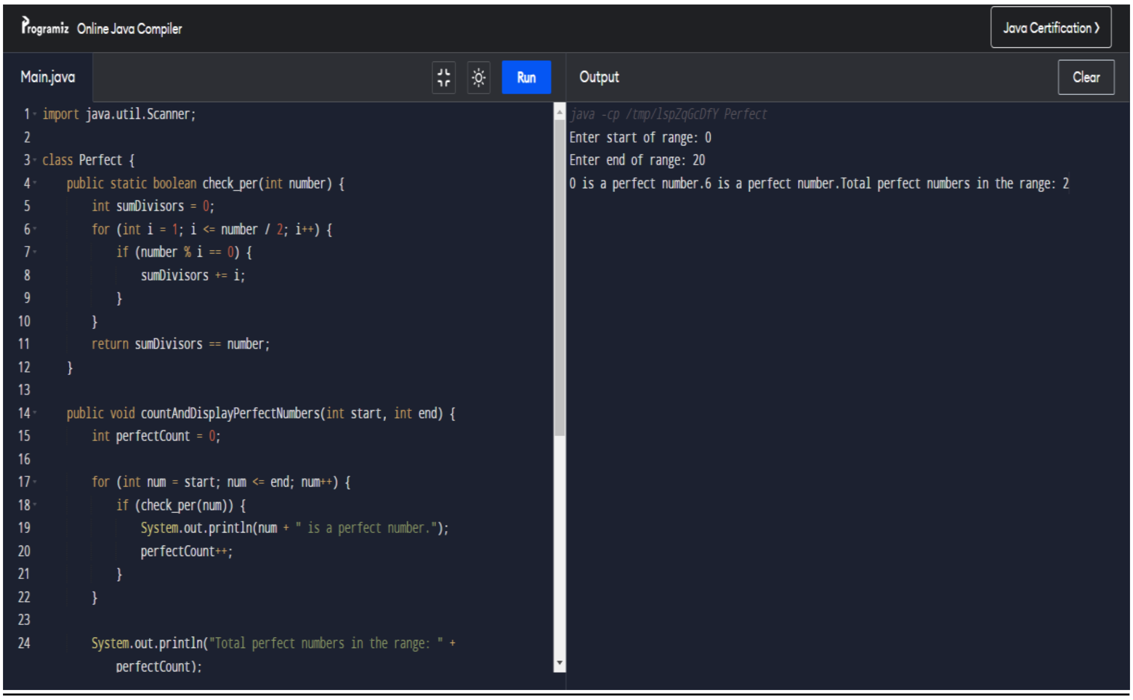
```
import java.util.ArrayList;
import java.util.Scanner;

class Perfect {
    private int start;
    private int end;

    public Perfect(int start, int end) {
        this.start = start;
        this.end = end;
    }

    public static boolean checkPer(int number) {
        int sum = 0;
        for (int i = 1; i <= number / 2; i++) {
            if (number % i == 0) {
                sum += i;
            }
        }
        return sum == number;
    }

    public ArrayList<Integer> findPerfectNumbers() {
        ArrayList<Integer> perfectNumbers = new ArrayList<>();
        for (int num = start; num <= end; num++) {
            if (checkPer(num)) {
                perfectNumbers.add(num);
            }
        }
        return perfectNumbers;
```

```
}

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

    System.out.print("Enter the start of the range: ");
    int startRange = scanner.nextInt();

    System.out.print("Enter the end of the range: ");
    int endRange = scanner.nextInt();

    Perfect perfectObj = new Perfect(startRange, endRange);
    ArrayList<Integer> perfectNumbers = perfectObj.findPerfectNumbers();

    int count = perfectNumbers.size();
    System.out.println("Number of perfect numbers within the range: " + count);

    System.out.println("Perfect numbers within the range:");
    for (int num : perfectNumbers) {
        System.out.println(num + " is a perfect number.");
    }
}
```

```
}
```

Output:

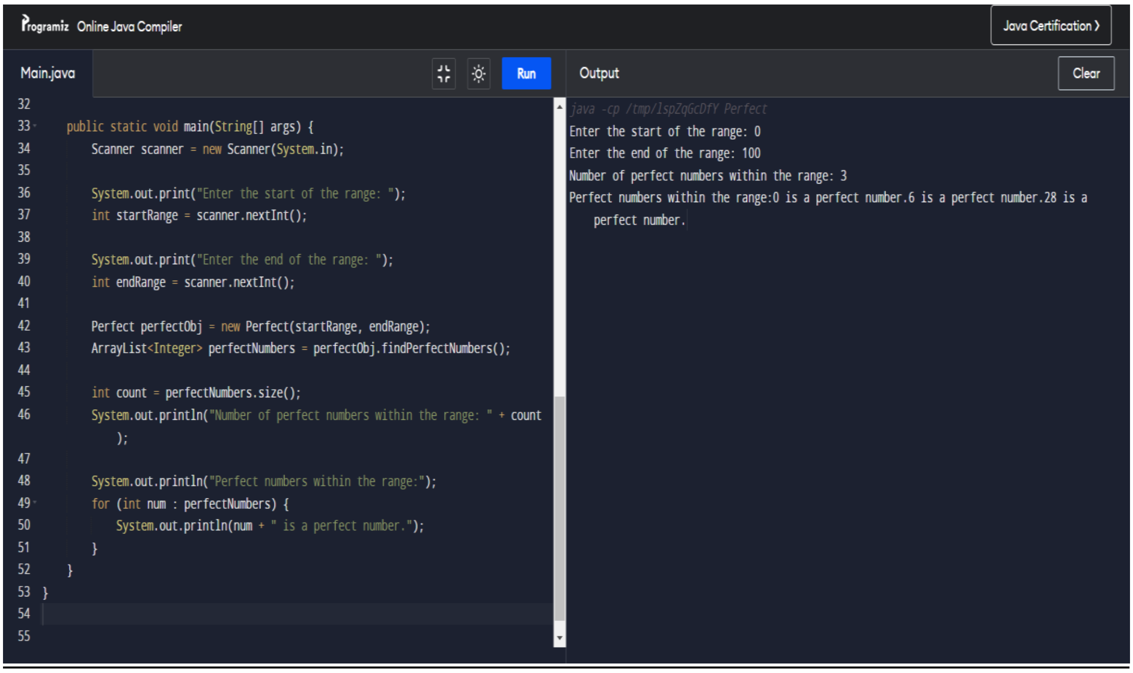
VARIATION-1



```java
import java.util.Scanner;

class Perfect {
    public static boolean check_per(int number) {
        int sumDivisors = 0;
        for (int i = 1; i <= number / 2; i++) {
            if (number % i == 0) {
                sumDivisors += i;
            }
        }
        return sumDivisors == number;
    }

    public void countAndDisplayPerfectNumbers(int start, int end) {
        int perfectCount = 0;

        for (int num = start; num <= end; num++) {
            if (check_per(num)) {
                System.out.println(num + " is a perfect number.");
                perfectCount++;
            }
        }

        System.out.println("Total perfect numbers in the range: " +
            perfectCount);
```

```
java -cp /tmp/lspZqGcDfY Perfect
Enter start of range: 0
Enter end of range: 20
0 is a perfect number.6 is a perfect number.Total perfect numbers in the range: 2
```

VARIATION-2



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

        System.out.print("Enter the start of the range: ");
        int startRange = scanner.nextInt();

        System.out.print("Enter the end of the range: ");
        int endRange = scanner.nextInt();

        Perfect perfectObj = new Perfect(startRange, endRange);
        ArrayList<Integer> perfectNumbers = perfectObj.findPerfectNumbers();

        int count = perfectNumbers.size();
        System.out.println("Number of perfect numbers within the range: " + count
            );

        System.out.println("Perfect numbers within the range:");
        for (int num : perfectNumbers) {
            System.out.println(num + " is a perfect number.");
        }
    }
}
```

```
java -cp /tmp/lspZqGcDfY Perfect
Enter the start of the range: 0
Enter the end of the range: 100
Number of perfect numbers within the range: 3
Perfect numbers within the range:0 is a perfect number.6 is a perfect number.28 is a
    perfect number.
```

Conclusion:

In this experiment, we learnt how to implement a program with one class and multiple classes. Along with accessibility with static and non-static methods within class and outside class.

Date: ______________

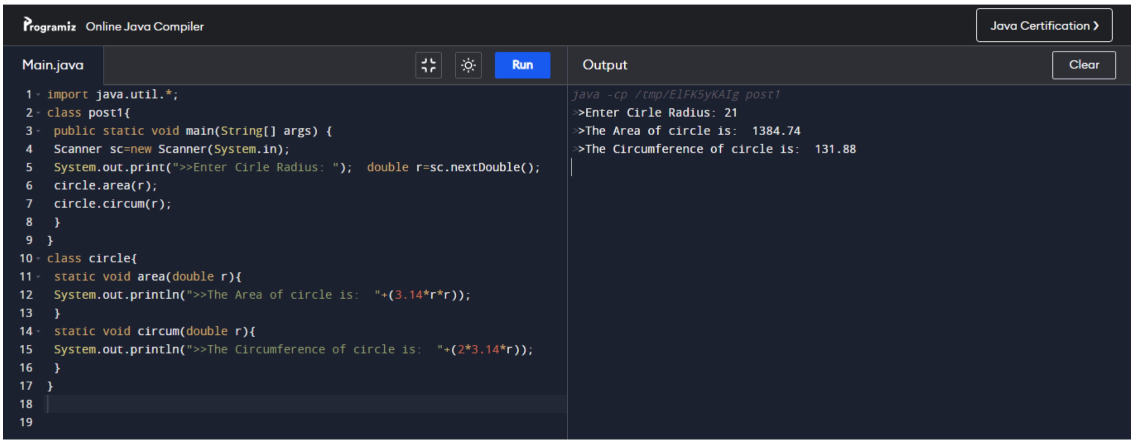
Signature of faculty in-charge

Post Lab Descriptive Questions:

Q.1 Write a program to find the area and circumference of a circle using two classes.

CODE-:

```
import java.util.*;
class post1{
 public static void main(String[] args) {
 Scanner sc=new Scanner(System.in);
 System.out.print(">>Enter Circle Radius: ");  double r=sc.nextDouble();
 circle.area(r);
 circle.circum(r);
 }
}
class circle{
 static void area(double r){
 System.out.println(">>The Area of circle is:  "+(3.14*r*r));
 }
 static void circum(double r){
 System.out.println(">>The Circumference of circle is:  "+(2*3.14*r));
 }
}
```



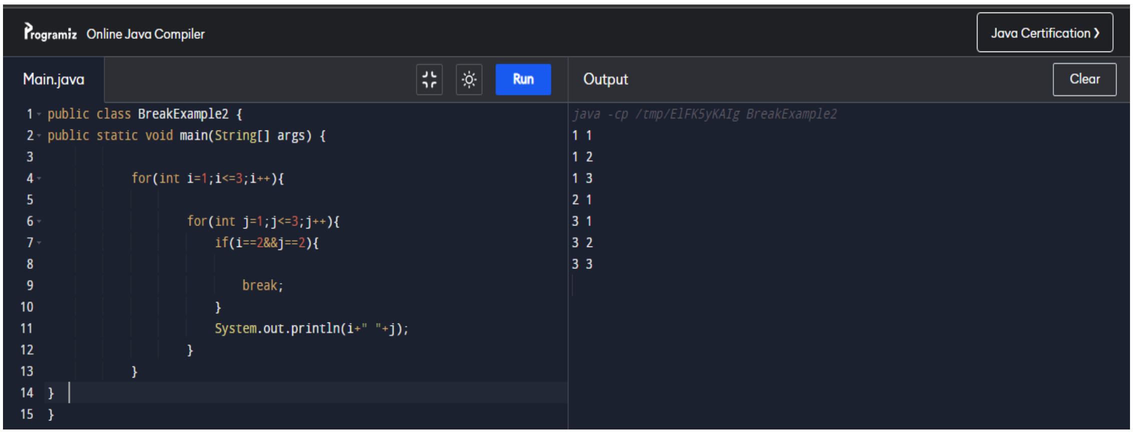
```java
import java.util.*;
class post1{
 public static void main(String[] args) {
 Scanner sc=new Scanner(System.in);
 System.out.print(">>Enter Cirle Radius: ");  double r=sc.nextDouble();
 circle.area(r);
 circle.circum(r);
 }
}
class circle{
 static void area(double r){
 System.out.println(">>The Area of circle is:  "+(3.14*r*r));
 }
 static void circum(double r){
 System.out.println(">>The Circumference of circle is:  "+(2*3.14*r));
 }
}
```

```
java -cp /tmp/ElFK5yKAIg post1
>>Enter Cirle Radius: 21
>>The Area of circle is:  1384.74
>>The Circumference of circle is:  131.88
```

Q.2 Write the output of following program

```java
public class BreakExample2 {
public static void main(String[] args) {
        //outer loop
        for(int i=1;i<=3;i++){
            //inner loop
            for(int j=1;j<=3;j++){
               if(i==2&&j==2){
                  //using break statement inside the inner loop
                  break;
               }
               System.out.println(i+" "+j);
            }
        }
}
}
```

Output:



```java
public class BreakExample2 {
public static void main(String[] args) {

        for(int i=1;i<=3;i++){

            for(int j=1;j<=3;j++){
               if(i==2&&j==2){

                  break;
               }
               System.out.println(i+" "+j);
            }
        }
}
}
```

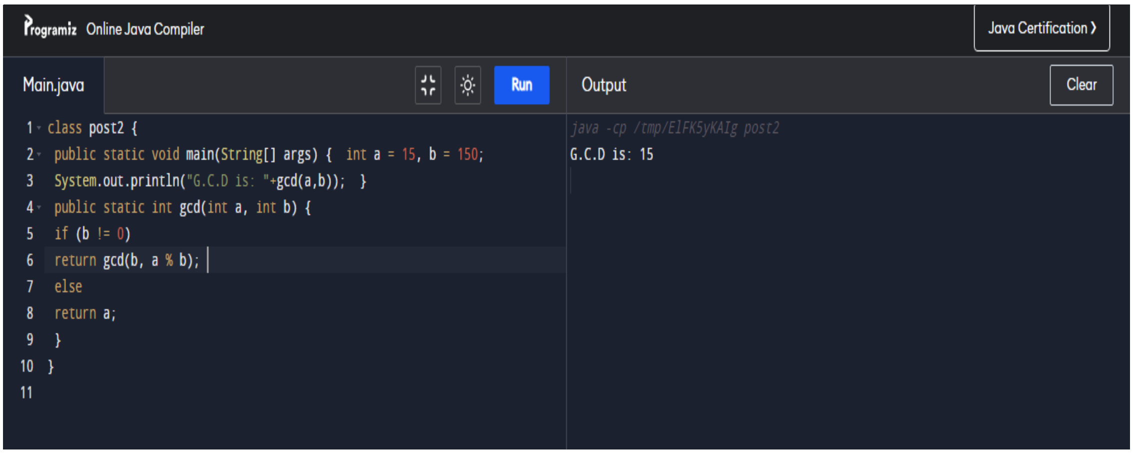
```
java -cp /tmp/ElFK5yKAIg BreakExample2
1 1
1 2
1 3
2 1
3 1
3 2
3 3
```

Q.3 Why is Java known as a platform independent language?

Java is based on the Write-Once-Run-Anywhere concept that makes it Platform independent. It is platform independent because the program written in it is not directly converted into machine code but instead is converted into byte code by Java compiler, this byte code is then converted into machine readable code by Java Virtual Machine (JVM). JDK including JVM must be installed in a platform.

Q.4 Write a recursive static method for calculation of gcd of a number.

```
class post2 {
 public static void main(String[] args) {  int a = 15, b = 150;
 System.out.println("G.C.D is: "+gcd(a,b));  }
 public static int gcd(int a, int b) {
 if (b != 0)
 return gcd(b, a % b);
 else
 return a;
 }
}
```



```
class post2 {
 public static void main(String[] args) {  int a = 15, b = 150;
 System.out.println("G.C.D is: "+gcd(a,b));  }
 public static int gcd(int a, int b) {
 if (b != 0)
 return gcd(b, a % b);
 else
 return a;
 }
}
```

```
java -cp /tmp/ElFK5yKAIg post2
G.C.D is: 15
```