K. J. Somaiya College of Engineering, Mumbai-77

Batch: B1 Roll No.: 1711072

Experiment / assignment / tutorial No.01

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

TITLE : GCD and LCM

AIM: Write a recursive function ‘GCD’ to find the GCD of the given two numbers. Use this in main to find the GCD and LCM two given numbers. (Scanner class)

Expected OUTCOME of Experiment:

CO 2. Solve problems using Java basic constructs (like if else statement, control structures, and data types, array, string, vectors, packages, collection class).

Books/ Journals/ Websites referred:

1. Ralph Bravaco , Shai Simoson , “Java Programing From the Group Up” Tata McGraw-Hill.

2. Grady Booch, Object Oriented Analysis and Design .

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

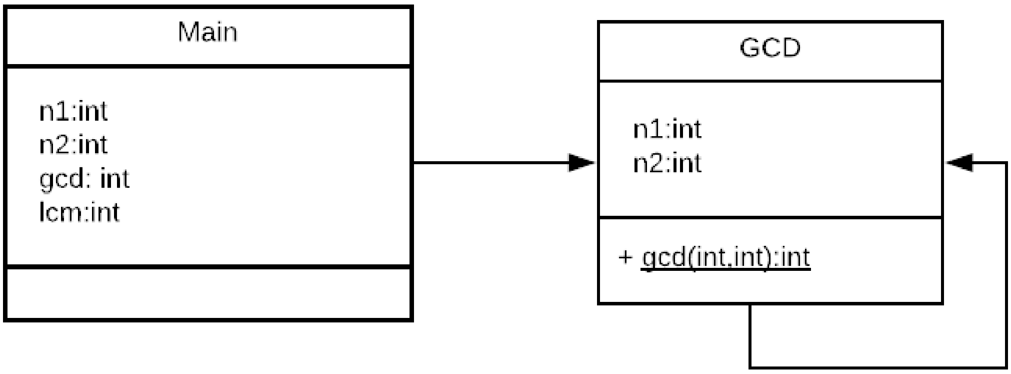
Method	Returns
int nextInt()	Returns the next token as an int. If the next token is not an integer,InputMismatchException is thrown.
long nextLong()	Returns the next token as a long. If the next token is not an integer,InputMismatchException is thrown.
float nextFloat()	Returns the next token as a float. 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:



Algorithm:

Main function

```
Step 1: START
Step 2: PRINT "Enter two numbers: "
Step 3: INPUT n1 and n2
Step 4: gcd = CALL calc.gcd(arguments : n1 , n2)
Step 5: lcm=(n1*n2)/gcd
Step 6: PRINT gcd
Step 7: PRINT lcm
Step 8: STOP
```

GCD (parameters n1 , n2) function

```
Step 1: START
Step 2: IF n1==0 Return n2
Step 3: ELSE Return RECURSIVELY gcd( n2%n1 , n1 )
```

Implementation details:

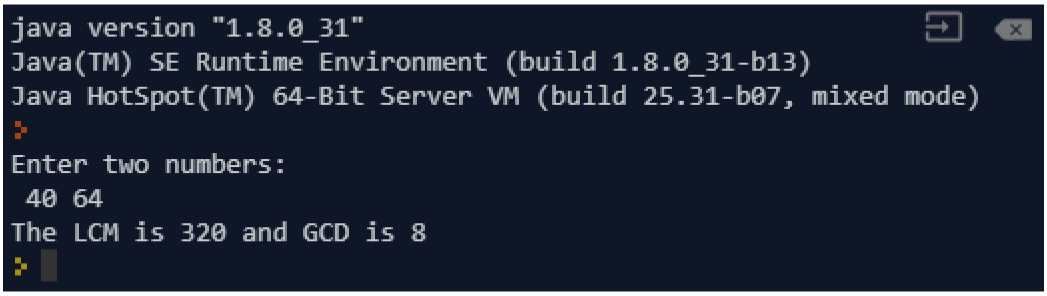
```
import java.util.Scanner;
class Main
{
  public static void main(String args[])
  {
    Scanner read = new Scanner(System.in);
    GCD calc = new GCD();
    int n1,n2;
    System.out.println("Enter two numbers: ");
    n1=read.nextInt();
    n2=read.nextInt();

    int gcd = calc.gcd(n1,n2);
    int lcm = n1*n2/gcd;

    System.out.println("The LCM is "+ lcm +" and GCD is "+ gcd);
  }
}
class GCD
{
  public static int gcd(int n1,int n2)
  {
    if(n1==0)
    return n2;
     else
  return gcd(n2%n1,n1);
  }
}
```

For cross verification, code is available at:
https://repl.it/@ARGHYADEEPDAS/DarkvioletZigzagTask

Output Screen:



```
java version "1.8.0_31"
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)
›
Enter two numbers:
 40 64
The LCM is 320 and GCD is 8
›
```

Conclusion: In this experiment, we learn the difference between static, non-static functions and members of functions. We implement the code for various cases, and got the expected output.

Post Lab Descriptive Questions (Add questions from examination point view)

Q.1What is the meaning of Return data type void?

a). An empty memory space is returned so that the developers can utilize it.

b). void returns no data type.

c). void is not supported in Java

d). None of the above

Ans. b) Void returns no data type.

Q.2 Write the output of following program

```
 Class Sample{
int a;
static int b=5;
public static void main(String args[]) {
        a=10;
        b=10;
System.out.println("a="+a+" b="+b);
}
```

1. a=5 b=10
2. a=10 b=10
3. compile time error
4. a=10 b=5

Ans. 3) The code will throw a compile time error as braces aren't closed for class.

Q.3 Write a recursive static method for calculation of factorial of n number.

```
import java.util.*;
class Factorial
{
        static int fact(int n)
        {
                int result;
        if(n==1)
            return 1;
        result = fact(n-1) * n;
        return result;
    }
}
class Main
{
    static int f,n;
    public static void main(String [] args)
    {

        int i;
        System.out.println("Enter number of numbers: ");
        Scanner s = new Scanner(System.in);
        Factorial u = new Factorial();
        i = s.nextInt();
        do
        {
            System.out.println("Enter number: ");
            f = s.nextInt();
            n = u.fact(f);
            System.out.print("The Factorial of " +f );
            System.out.println(" is "+n);
            i--;
        }while(i>0);
    }
}
```

Date: 21/07/2018 **Signature of faculty in-charge**