**`**

**Batch: B2 Roll No.: 1611103**

**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();

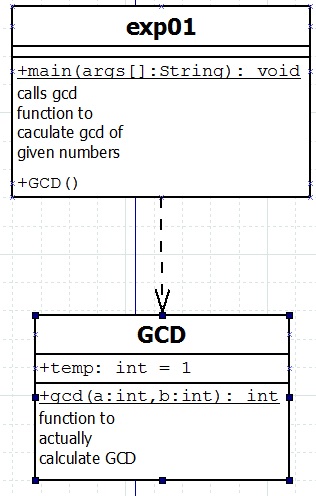
float real = in.nextFloat();

long number2 = in.nextLong();

double real2 = in.nextDouble();

String string = in.next();

**Class Diagram:**

****

**Algorithm:**

1. **Start**
2. **Take 2 numbers in as a and b.**
3. **Initialise temp variable to 1.**
4. **Store remainder of division of a by b in temp.**
5. **If temp is 0 then b is GCD.**
6. **Else load a with b and b with temp and go to step 4.**
7. **Initialise LCM variable to a\*b/GCD.**
8. **Stop.**

**Implementation details:**

**import java.util.Scanner;**

**class GCD**

**{**

**int temp = 1;**

**/\* static \*/ int gcd(int a, int b)**

**{**

**/\*int temp = 1\*/ //if using static mode.**

**temp = a%b;**

**if(temp == 0)**

**return b;**

**else**

**return gcd(b,temp);**

**}**

**}**

**class exp01**

**{**

**public static void main(String args[])**

**{**

**int num,deno;int MAX,MIN;**

**GCD obj = new GCD();**

**Scanner sc = new Scanner(System.in);**

**System.out.print("Enter the first number : ");**

**num = sc.nextInt();**

**System.out.print("Enter the second number : ");**

**deno = sc.nextInt();**

**if(num > deno)**

**{MAX = num;MIN = deno;}**

**else**

**{MAX = deno;MIN = num;} //GCD.gcd(MAX,MIN); for static access from Gcd class.**

**System.out.println("The GCD of given nos : " + obj.gcd(MAX,MIN)); //gcd(MAX,MIN); for ststic int gcd.**

**int LCM = (MAX\*MIN)/obj.gcd(MAX,MIN); //exp01 obj1;obj1.gcd(MAX,MIN); for non-static access from same class.**

**System.out.println("The LCM of given nos : " + LCM);**

**}**

**// /\* static \*/ int gcd(int a, int b)**

**// {**

**// /\*int temp = 1\*/ //if using static mode.**

**// temp = a%b;**

**// if(temp == 0)**

**// return b;**

**// else**

**// return gcd(b,temp);**

**// }**

**// int gcd(int a, int b)**

**// {**

**// /\*int temp = 1\*/ //if using static mode.**

**// temp = a%b;**

**// if(temp == 0)**

**// return b;**

**// else**

**// return gcd(b,temp);**

**// }**

**}**

**Conclusion:**

Hence difference between static and non-static functions are studied and both are implemented successfully.

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**

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

**Q.1 What 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

**(b).**

**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

**(3). Compile time error**

**Due to referencing of non-static member a inside main.**

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

static int fact(int x)

{

if (x == 0 || x == 1)

return 1;

else

return x\*fact(x-1);

}