`

Batch: A3	**Roll No.: 1911034**
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:

Class Name
public class Exp01
Methods

1. public static void main(string args[]);
2. int gcd(int n1, int n2);//non static method in same class
3. static int gcdb(int n1, int n2);//static method in same class
4. int gcda(int n1, int n2);//non static method in another class
5. static int gcdc(int n1, int n2);// static method in another class

Objects
1. sc (belonging to scanner class for user input) 2. e1 (belonging to non static method of same class) 3. e2 (belonging to non static method of another class)
Variables
int a , b , gcd1, lcm1, gcd2,lcm2,gcd3,lcm3,fcd4,lcm4;

Algorithm:

Step01: Import the package java.util , in order to import the Scanner Class to take user input.

Step02: Declare two classes class Exp01 having the main method and a static and non static method and the class Exp02 for a static and non-static method.

Step03: In class Exp01, create the object of Scanner Class that invokes the function nextInt() in order to accept user input.

Step04: Accept two integers from the user and using 2 functions (one static and the other non static) in each of the two classes , we will find GCD and LCM of the two numbers.

Step05: To find the GCD and LCM of the two numbers using Non Static methods of the two classes , create an object for each of the two classes ,and call the method using objectname.methodname()

Step06: To call the static method inside the same class for calculating GCD , we can directly call the method by the method name gcd().

Step07: To call the static method in a different class, we can call the method by using ClassName.MethodName().

Step08: In each of the methods , we can calculate GCD by finding the modulus when the larger number is divided by the smaller , and calling the method recursively until it reaches 0

Step09: Return the value of the GCD back to the calling function(Main function in Exp01) and calculate the LCM by using GCDxLCM=axb(where a and b are the two numbers whose GCD and LCM was to be calculated.

Step10: Print the value of GCD and LCM calculate.

Implementation details:

Code :

```
import java.util.*;

public class Exp01
{
public static void main(String args[])
{
int a,b,gcd1,lcm1,gcd2,lcm2,gcd3,lcm3,lcm4,gcd4;
```

```
Scanner sc= new Scanner(System.in);//creating an object of Scanner Class
to accept the user input

System.out.println("Enter the two numbers");
a = sc.nextInt();
b= sc.nextInt();
Exp01 e1 = new Exp01();//creating an object of non static method of class
Exp01
gcd1 = e1.gcd(a,b);
lcm1 =(a*b)/gcd1;
System.out.println("The gcd of the two numbers using the same class with
non static method is" +gcd1);
System.out.println("The lcm of the two numbers using the same with non
static method is class is" +lcm1);
Exp02 e2 = new Exp02();//creating an object of non static method of class
Exp02
gcd2= e2.gcda(a,b);
lcm2 = (a*b)/gcd2;
System.out.println("The gcd of the two numbers using the different class
which has non static method is" +gcd2);
System.out.println("The lcm of the two numbers using the different class
which has non static method is" +lcm2);
gcd3 = gcdb(a,b);
lcm3= (a*b)/gcd3;
System.out.println("The gcd of the two numbers using the same class which
has  static method is" +gcd3);
System.out.println("The lcm of the two numbers using the same class which
has  static method is" +lcm3);
gcd4= Exp02.gcdc(a,b);
lcm4=(a*b)/gcd4;
System.out.println("The gcd of the two numbers using the different class
which has  static method is" +gcd4);
System.out.println("The lcm of the two numbers using the different class
which has  static method is" +lcm4);


}

static int gcdb(int n1, int n2)//Static Method to calculate GCD which is
located in the same class as main method
{
  if (n2 != 0){
         return gcdb(n2, n1 % n2);
      } else{
```

```
            return n1;
        }}

int gcd(int n1, int n2)// non static method in same class to calculate
GCD
{
  if (n2 != 0){
            return gcd(n2, n1 % n2);
        } else{
            return n1;
        }}

}

class Exp02
{
int gcda(int n1, int n2)//non static method in different class to
calculate GCD
{
 if (n2 != 0){
            return gcda(n2, n1 % n2);
        } else{
            return n1;
        }

}
static int gcdc(int n1, int n2)// static method in different class to
calculate GCD
{
  if (n2 != 0){
            return gcdc(n2, n1 % n2);
        } else{
            return n1;
        }}

}
```

Screenshots of Output:

Case 1: Both the numbers are positive :

```
C:\Users\Aditi Paretkar\Desktop\OOPM JAVA programs>java Exp01
Enter the two numbers
55
35
The gcd of the two numbers using the same class with non static method is5
The lcm of the two numbers using the same with non static method is class is385
The gcd of the two numbers using the different class which has non static method is5
The lcm of the two numbers using the different class which has non static method is385
The gcd of the two numbers using the same class which has  static method is5
The lcm of the two numbers using the same class which has  static method is385
The gcd of the two numbers using the different class which has  static method is5
The lcm of the two numbers using the different class which has  static method is385

C:\Users\Aditi Paretkar\Desktop\OOPM JAVA programs>
```

Case 2: One of the numbers is positive and the other is zero

```
C:\Users\Aditi Paretkar\Desktop\OOPM JAVA programs>java Exp01
Enter the two numbers
15
0
The gcd of the two numbers using the same class with non static method is15
The lcm of the two numbers using the same with non static method is class is0
The gcd of the two numbers using the different class which has non static method is15
The lcm of the two numbers using the different class which has non static method is0
The gcd of the two numbers using the same class which has  static method is15
The lcm of the two numbers using the same class which has  static method is0
The gcd of the two numbers using the different class which has  static method is15
The lcm of the two numbers using the different class which has  static method is0

C:\Users\Aditi Paretkar\Desktop\OOPM JAVA programs>
```

Case 3: One of the numbers is positive and the other is negative:

```
C:\Users\Aditi Paretkar\Desktop\OOPM JAVA programs>java Exp01
Enter the two numbers
-55
11
The gcd of the two numbers using the same class with non static method is11
The lcm of the two numbers using the same with non static method is class is-55
The gcd of the two numbers using the different class which has non static method is11
The lcm of the two numbers using the different class which has non static method is-55
The gcd of the two numbers using the same class which has  static method is11
The lcm of the two numbers using the same class which has  static method is-55
The gcd of the two numbers using the different class which has  static method is11
The lcm of the two numbers using the different class which has  static method is-55

C:\Users\Aditi Paretkar\Desktop\OOPM JAVA programs>
```

Case 04: Both the numbers are negative :

```
C:\Users\Aditi Paretkar\Desktop\OOPM JAVA programs>java Exp01
Enter the two numbers
-4
-18
The gcd of the two numbers using the same class with non static method is-2
The lcm of the two numbers using the same with non static method is class is-36
The gcd of the two numbers using the different class which has non static method is-2
The lcm of the two numbers using the different class which has non static method is-36
The gcd of the two numbers using the same class which has  static method is-2
The lcm of the two numbers using the same class which has  static method is-36
The gcd of the two numbers using the different class which has  static method is-2
The lcm of the two numbers using the different class which has  static method is-36

C:\Users\Aditi Paretkar\Desktop\OOPM JAVA programs>
```

Conclusion:

In this experiment we have learnt the concept of objects , and how they are used to call the non-static functions which are in the same class as well as in the different class. We have also learnt how to implement GCD using a recursive algorithm

6. For calling the non static function in the same class as that of the main method as well as in a different class, we create an object of that class and then call the method using the object
7. For calling the static function , we have two cases (we don't create object while calling the static function)
    a. For static function in the same class , we call it using the method name().
    b. For static function in a different class, we call it using ClassName.MethodName()

Date: _____ Signature of faculty in-char

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

Ans: b) void returns no data type

Q.2 write the output of following program

```
1.  public class BreakExample2 {
2.  public static void main(String[] args) {
3.      //outer loop
4.      for(int i=1;i<=3;i++){
5.          //inner loop
6.          for(int j=1;j<=3;j++){
7.              if(i==2&&j==2){
8.                  //using break statement inside the inner loop
9.                  break;
10.             }
11.             System.out.println(i+" "+j);
12.         }
13.     }
14. }
15. }
```

Output : 1 1
1 2
1 3
2 1
3 1
3 2
3 3

Q.3

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

answer- compile time error

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

```
import java.util.*;
class Main {
  public static void main(String[] args) {
    int num,res;
    Scanner sc= new Scanner(System.in);
    System.out.println("Enter the number who's factorial you wish to
find");

    num= sc.nextInt();
    res = fact(num);
    System.out.println("The factorial of the entered number is " +res);
  }
   public static int fact(int x)//static method to find the factorial of
the entered number recursively
   {

       if(x==1)
       return 1;
       else return x*fact(x-1);
```

```
    }
}
```