

**Batch: B1          Roll No.: 16010124080**

**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.

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

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

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

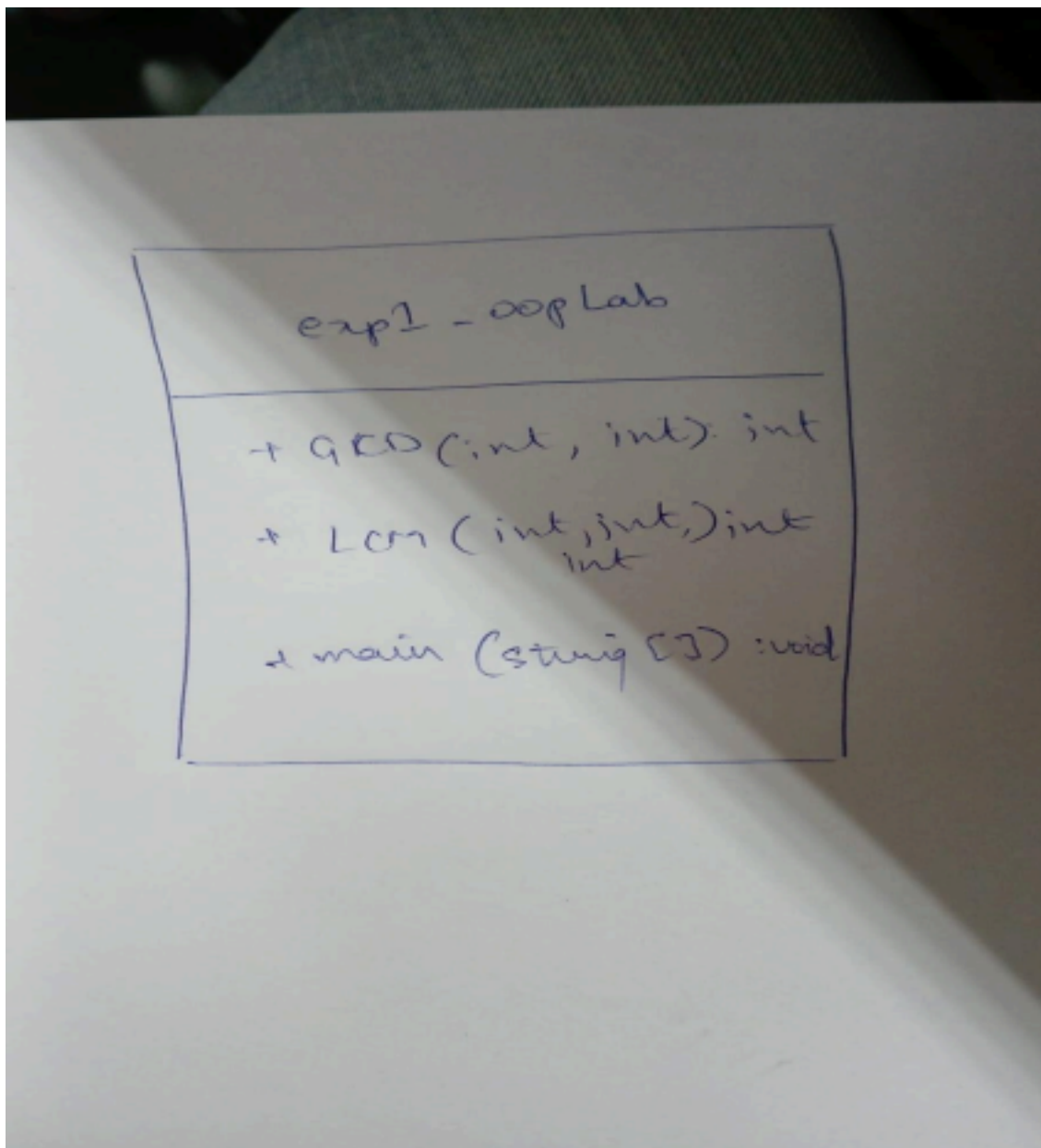
**Department of Computer Engineering**

Page No OOPM Sem III/July - Nov 2024

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

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

**Class Diagram:**



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

**Algorithm:**

**Step 1: Start**

**Step 2: Input two integers `a` and `b`**

**Step 3: Call the `GCD` method:**

- If `y == 0` or `x == 0`, return `max(x, y)`
- Else return `GCD(y, x % y)` recursively

**Step 4: Store the returned value in `ans_gcd`**

**Step 5: Call the `LCM` method:**

- Use formula:  $LCM = a \times b / GCD$
- Store the result in `ans_lcm`

**Step 6: Print `ans_gcd` and `ans_lcm`**

**Step 7: End**

**Implementation details:**

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

```
1  import java.util.*;
2
3  public class exp1_oopLab {
4      public static int GCD(int x, int y) {
5          if (y == 0 || x == 0) {
6              return Math.max(x, y);
7          }
8          return GCD(y, x % y);
9      }
10
11     public static int LCM(int x, int y, int gcd) {
12         return (x * y) / gcd;
13     }
14
15
16
17     public static void main(String[] args) {
18         Scanner sc = new Scanner(System.in);
19         System.out.println("Enter 1st number:");
20         int a = sc.nextInt();
21         System.out.println("Enter 2nd number:");
22         int b = sc.nextInt();
23         ans_gcd = GCD(a, b);
24         ans_lcm = LCM(a, b, ans_gcd);
25         System.out.println("GCD of " + a + " and " + b + " is: " + ans_gcd);
26         System.out.println("LCM of " + a + " and " + b + " is: " + ans_lcm);
27     }
28 }
29
```

**Output:**

```
PS C:\Users\Shounak Dutta\Documents\The Matrix\collegeMaterial\java> javac exp1_ooptab.java
PS C:\Users\Shounak Dutta\Documents\The Matrix\collegeMaterial\java> java exp1_ooptab
Enter 1st number:
78
Enter 2nd number:
28
GCD of 78 and 28 is: 2
LCM of 78 and 28 is: 788
PS C:\Users\Shounak Dutta\Documents\The Matrix\collegeMaterial\java> |
```

### Department of Computer Engineering

Page No OOPM Sem III/July - Nov 2024

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

### Conclusion:

This Java program efficiently computes the **Greatest Common Divisor (GCD)** and **Least Common Multiple (LCM)** of two user-input integers. By using a recursive approach for GCD based on the Euclidean algorithm, the code ensures optimal performance. The LCM is calculated using the mathematical relationship between GCD and LCM

**Date: 29/7/25 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.

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

```
patterns > J circle.java > Circles
1  class Circles {
2      public float area_r(int r) {
3          return 3.14f * r * r;
4      }
5
6      public float circumference(int r) {
7          return 2 * 3.14f * r;
8      }
9  }
10 public class circle {
11     public static void main(String[] args) {
12         Circles example = new Circles();
13         float area = example.area_r(4);
14         float circum = example.circumference(7);
15         System.out.println("Area of circle " + area);
16         System.out.println("Circumference of circle " + circum);
17     }
18 }
```

Q.2 Write the output of following program

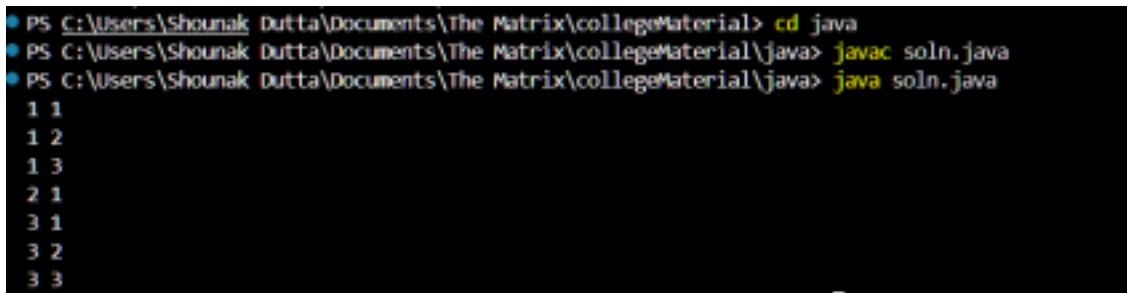
1. **public class** BreakExample2 {
2. **public static void** main(String[] args) {
3. //outer loop
4. **for**(**int** i=1;i<=3;i++){
5. //inner loop
6. **for**(**int** j=1;j<=3;j++){
7. **if**(i==2&&j==2){
8. //using break statement inside the inner loop 9. **break**;
10. }
11. System.out.println(i+"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

## Output:

### Department of Computer Engineering

Page No OOPM Sem III/July - Nov 2024

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



```
PS C:\Users\Shounak Dutta\Documents\The Matrix\collegeMaterial> cd java
PS C:\Users\Shounak Dutta\Documents\The Matrix\collegeMaterial\java> javac soln.java
PS C:\Users\Shounak Dutta\Documents\The Matrix\collegeMaterial\java> java soln.java
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 known as a **platform-independent language** because **Java programs are compiled into bytecode**, which can run on any device that has a **Java Virtual Machine (JVM)**.

- Unlike C/C++ which compiles code into machine-dependent binaries, Java's **.class** files (bytecode) are **not tied to any specific operating system or processor**.
- The **JVM acts as an interpreter** between the bytecode and the machine it runs on, making Java applications portable across different systems (Windows, macOS, Linux, etc.).

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



Department of Computer Engineering

Page No OOPM Sem III/July - Nov 2024



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

```
java > J Factorial.java > Factorial > main(String[])
1 public class Factorial {
2
3     public static int factorial(int n) {
4         if (n == 0 || n == 1)
5             return 1;
6         else
7             return n * factorial(n - 1);
8     }
9
10    Run | Debug
11    public static void main(String[] args) {
12        int number = 5; // Example number to calculate factorial
13        int result = factorial(number);
14        System.out.println("Factorial of " + number + " is: " + result);
15    }
16
```

**Department of Computer Engineering**

Page No OOPM Sem III/July - Nov 2024