Experiment Name:

Java Fundamentals: User Input, Array, Logical & Bitwise Operators.

Objective:

1. To demonstrate user input and output in Java using Scanner class methods.
2. To explain limitations of nextInt(), nextLine(), and next() methods and provide solutions.
3. To illustrate creation and traversal of single and uneven multidimensional arrays in Java.
4. To illustrate the use of bitwise and logical operators.

Introduction:

The objective of this lab session is to explore Java fundamentals, specifically focusing on user input and output using Scanner class methods. Additionally, we aim to address the limitations of nextInt(), nextLine(), and next() methods and present effective solutions. Furthermore, we will create and traverse single and uneven multidimensional arrays using for loops to showcase their versatility. Finally, we will delve into the functionality of bitwise and logical operators for efficient data processing.

Methodology:

1. Write a Java program to obtain user input and display output for ID, Name, and CGPA.

2. Demonstrate the limitations of nextInt(), nextLine(), and next() methods and provide solutions by utilizing additional nextLine() calls.

3. Implement single and uneven multidimensional arrays with elements increasing in size (1, 2, 3) and traverse them using nested for loops.

4. Showcase bitwise AND, OR, logical AND, and logical OR operations for different numerical inputs.

Source Code:

```java
import java.util.Scanner;
public class Fundamentals {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter ID: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter Name: ");
        String name = scanner.nextLine();
        System.out.print("Enter CGPA: ");
        double cgpa = scanner.nextDouble();
        scanner.nextLine();
        System.out.println("ID: " + id + ", Name: " + name + ", CGPA: " + cgpa);
        System.out.print("\nEnter a word using next(): ");
        String word = scanner.next();
        System.out.print("Enter a line using nextLine(): ");
        scanner.nextLine();
        String line = scanner.nextLine();
        System.out.println("Word entered using next(): " + word);
        System.out.println("Line entered using nextLine(): " + line);
        int[] numbers = {10, 20, 30, 40, 50};
        int[][] unevenArray = new int[4][];
        for (int i = 0; i < unevenArray.length; i++) {
            unevenArray[i] = new int[i+1];
            for (int j = 0; j < i+1 ; j++) {
                unevenArray[i][j] = i+1;
            }
        }
        System.out.println("\nSingle-dimensional array:");
        for (int i = 0; i < numbers.length; i++) {
            System.out.print(numbers[i] + " ");
        }
        System.out.println("\n\nUneven multidimensional array:");
        for (int i = 0; i < unevenArray.length; i++) {
            for (int j = 0; j < unevenArray[i].length; j++) {
                System.out.print(unevenArray[i][j] + " ");
            }
            System.out.println();
        }
        int a = 5;
        int b = 3;
        int c = 2;
        int d = 4;
        System.out.println("\nBitwise AND: " + (a & b));
        System.out.println("Bitwise OR: " + (c | d));
        System.out.println("Logical AND: " + (a > b && c > d));
        System.out.println("Logical OR: " + (a > b || c > d));
        scanner.close();
    }
}
```

Sample Input and Output:

```
Enter ID: 220201075
Enter Name: Nahid Hasan Arif
Enter CGPA: 3.74
ID: 220201075, Name: Nahid Hasan Arif, CGPA: 3.74

Enter a word using next(): Nahid Hasan
Enter a line using nextLine(): Nahid Hasan Arif
Word entered using next(): Nahid
Line entered using nextLine(): Nahid Hasan Arif

Single-dimensional array:
10 20 30 40 50

Uneven multidimensional array:
1
2 2
3 3 3
4 4 4 4

Bitwise AND: 1
Bitwise OR: 6
Logical AND: false
Logical OR: true
```

Discussion:

Demonstrating user I/O using Scanner class methods. The limitations of nextInt() and nextDouble() methods, which leave a newline character in the input buffer. To overcome this issue, an additional nextLine() call to consume the newline character before using nextLine() to read the actual line of text, ensuring proper input reading and output display. The creation and traversal of single and uneven multidimensional arrays were successfully illustrated using for loops. Arrays offer flexibility by allowing varying row sizes, enabling efficient data representation in real-world scenarios. Bitwise AND, OR, logical AND, and logical OR operators which play a crucial role in manipulating individual bits of numbers and evaluating boolean expressions, facilitating efficient data processing and logical decision-making in programs.

Conclusion:

This lab provided an in-depth understanding of Java fundamentals related to user input/output, handling Scanner class limitations, creating multidimensional arrays, and using bitwise/logical operators. The acquired knowledge enhances Java programming skills and prepares us for more complex programming tasks in the future.