# Exploring Basic Arithmetic Operations and User Input in Java

## Introduction:

Welcome, budding Java enthusiasts! Today, we'll dive into some practical examples of basic arithmetic operations and user input in Java.

## Basic Arithmetic Operations:

Let's start by understanding how Java performs basic arithmetic operations.

```java
int sum1 = 100 + 50;
int sum2 = sum1 + 50;
int sum3 = sum1 + sum2;

System.out.println("Sum 1 : " + sum1);
System.out.println("Sum 2 : " + sum2);
System.out.println("Sum 3 : " + sum3);
```

- **Explanation:**

In this snippet, we're calculating and displaying the sums of numbers. This illustrates how variables can store and manipulate numeric values.

# More Arithmetic Operations:

```
int x = 10;
int y = 5;


System.out.println("X - Y = " + (x - y)); // subtraction
System.out.println("X + Y = " + (x + y)); // summation
System.out.println("X / Y = " + (x / y)); // division
System.out.println("X % Y = " + (x % y)); // modulo
```

- **Explanation:**

Here, we're showcasing subtraction, summation, division, and modulo operations. Understanding these operations is crucial for manipulating numerical data in your programs.
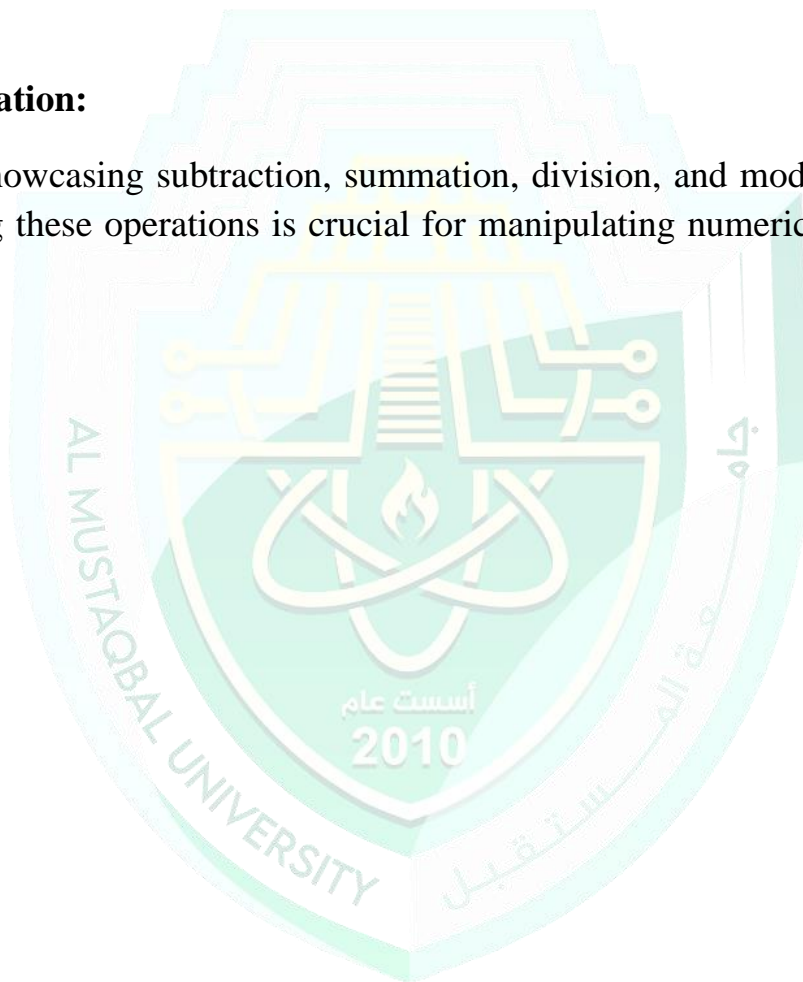
## Increment and Decrement:

Now, let's look at increment and decrement operations:

```
int x = 1;
System.out.println(++x); // increment and print
System.out.println(--x); // decrement and print
System.out.println(x += 3); // increment by 3 and print
System.out.println(x -= 1); // decrement by 1 and print
```

- **Explanation:**

This demonstrates how you can modify variables using increment and decrement operations, a handy technique in programming.

## User Input:

Finally, let's explore user input:

```java
import java.util.Scanner;
public class Main

{

    public static void main(String[] args) {

      System.out.println("What is your name?");
      Scanner scanner = new Scanner(System.in);
      String name = scanner.nextLine();
      System.out.println(name + " is a nice name!");


    }

}
```

Here, we're using Scanner to take user input, specifically their name. This interaction makes your programs dynamic and responsive to user input.

## Conclusion:

Understanding these fundamental concepts is key to building more complex Java programs. As you practice and experiment with these examples, you'll gain confidence in handling variables, performing arithmetic operations, and incorporating user input into your programs.