

**Statement Purpose:**

The purpose of this lab is to familiarize students with Conditional statements and explain how to control the sequence of statement execution, depending on the value of an expression using the decision making statements (if-then, if-then-else). There are some exercises, through which they will understand the concept learned in this chapter.

**Activity Outcomes:**

Students will learn how to write Java programs that make decisions, and execute different parts depending on a Boolean true/false value. Students will solve different problems that will help them to choose appropriate selection control statement (single if, block if, multiple if) for a given task. There are some exercises, through which they will understand the concept learned in this chapter.

**Theory Review** (5 minutes)**Selection Statements**

Java provides *selection statements*, which allow you to choose actions based on certain conditions. Selection statements use conditions to test if something is true or false. Java provides **six** relational operators(also known as comparison operators) that can be used to compare two values.

Relational Operators				
Java Operator	Mathematics Symbol	Name	Example (radius is 5)	Result
<	<	less than	radius < 0	false
<=	≤	less than or equal to	radius <= 0	false
>	>	greater than	radius > 0	true
>=	≥	greater than or equal to	radius >= 0	true
==	=	equal to	radius == 0	false
!=	≠	not equal to	radius != 0	true

**Caution**

The equality testing operator is two equal signs (==), not a single equal sign (=). The latter symbol is for assignment.

Java has several types of selection statements:

– one-way if statements, two-way if-else statements, nested if statements, multi-way if-else statements, switch statements, and conditional expressions.

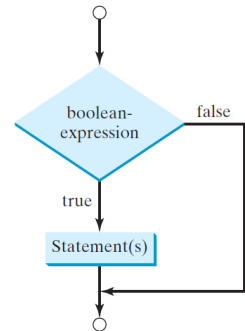


### One-way if Statements

A one-way if statement executes an action if and only if the condition is true.

**Syntax:**

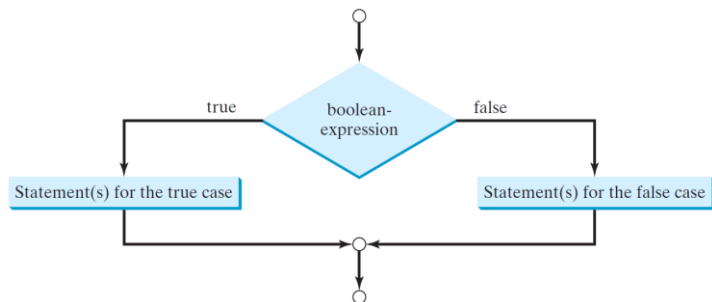
```
if (boolean-expression) {  
    statement(s);  
}
```



### Two-way if-else Statements

**Syntax:**

```
if (boolean-expression) {  
    statement(s)-for-the-true-case;  
}  
else {  
    statement(s)-for-the-false-case;  
}
```



If the **boolean-expression** evaluates to true, the **statement(s) for the true case** are executed. Else, the **statement(s) for the false case** are executed.

### Nested if and Multi-Way if-else Statements

- The statement inside an if or if-else statement can be any legal Java statement including another if or if-else statement!
- The inner if statement is said to be nested inside the outer if statement.



**Practice Activity with Lab Instructor** (10 minutes)

Write a program that prompts the user to enter a number, and displays **This number is positive** if the entered number is positive otherwise, displays **This number is negative** if it is negative . Here is a sample run:

```
Enter integer number: 10
This number is positive
```

```
Enter integer number: -5
This number is negative
```

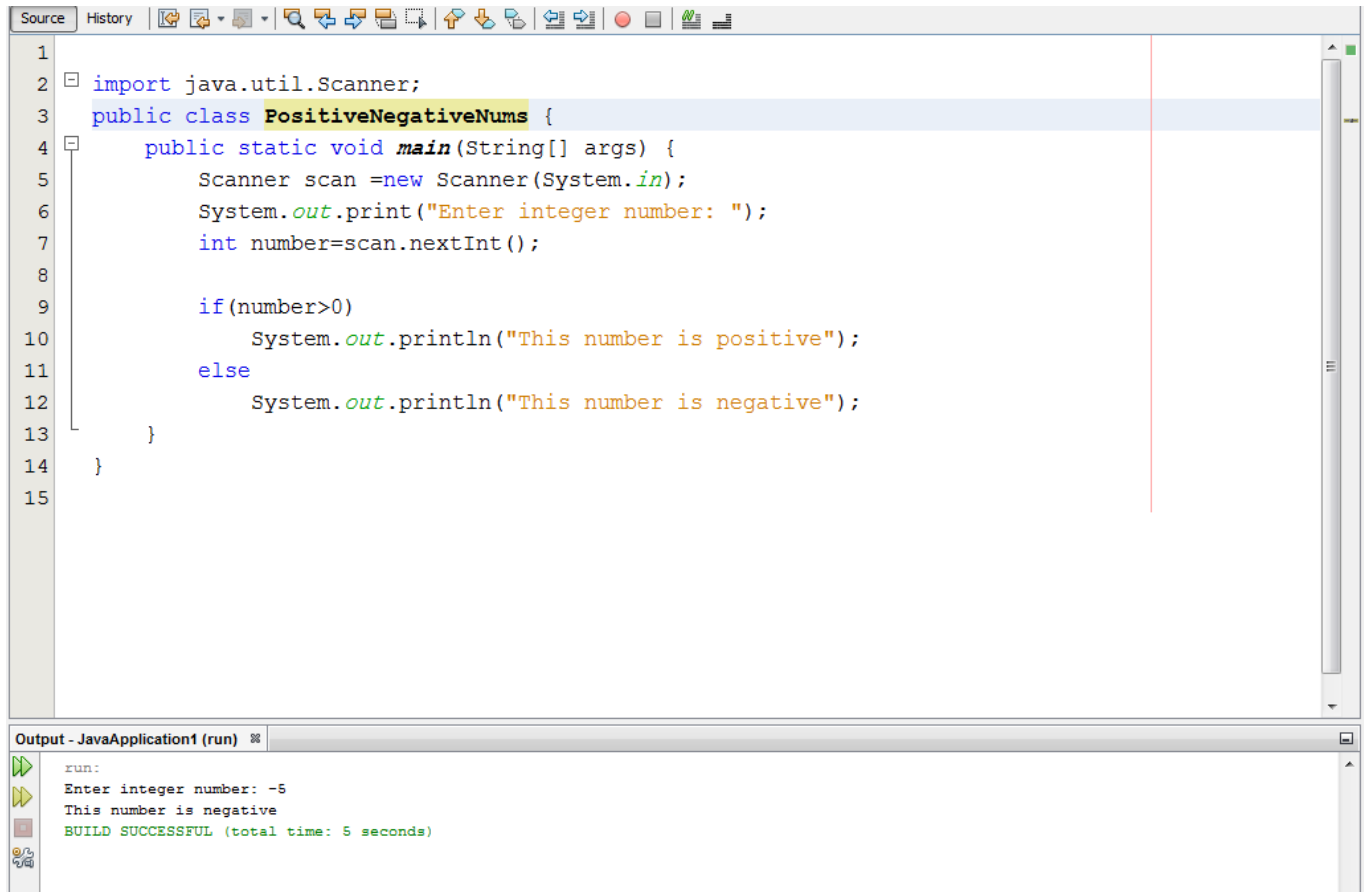
**Input:** Value of variable *number*.

**Processing and Output:** Check the *number*, if it is greater than zero display the first message, else display the second message.



**Solution:**

1. Open NetBeans and create a new project
2. Create a new java main class and write its name as **PositiveNegativeNums**
3. Write the following code inside the main method



```
1
2 import java.util.Scanner;
3 public class PositiveNegativeNums {
4     public static void main(String[] args) {
5         Scanner scan =new Scanner(System.in);
6         System.out.print("Enter integer number: ");
7         int number=scan.nextInt();
8
9         if(number>0)
10             System.out.println("This number is positive");
11         else
12             System.out.println("This number is negative");
13     }
14 }
15
```

Output - JavaApplication1 (run) ✖

```
run:
Enter integer number: -5
This number is negative
BUILD SUCCESSFUL (total time: 5 seconds)
```

**Short Exercise** (5 minutes)

In the above program if the user enters **zero**, the message **This number is negative**. Modify the program such that the message **Neither positive nor negative** is displayed when the **zero** value is entered.



**Individual Activities:** (60 minutes)

1. (Game: addition quiz) The program shown in the figure randomly generates a subtraction question with single digit. **Revise the program to randomly generate an addition question with two integers less than 100.**

```
1  import java.util.Scanner;
2
3  public class SubtractionQuiz {
4      public static void main(String[] args) {
5          // 1. Generate two random single-digit integers
6          int number1 = (int)(Math.random() * 10);
7          int number2 = (int)(Math.random() * 10);
8
9          // 2. If number1 < number2, swap number1 with number2
10         if (number1 < number2) {
11             int temp = number1;
12             number1 = number2;
13             number2 = temp;
14         }
15
16         // 3. Prompt the student to answer "What is number1 - number2?"
17         System.out.print
18         ("What is " + number1 + " - " + number2 + "? ");
19         Scanner input = new Scanner(System.in);
20         int answer = input.nextInt();
21
22         // 4. Grade the answer and display the result
23         if (number1 - number2 == answer)
24             System.out.println("You are correct!");
25         else {
26             System.out.println("Your answer is wrong.");
27             System.out.println(number1 + " - " + number2 +
28                 " should be " + (number1 - number2));
29         }
30     }
31 }
```

What is 6 - 6? 0 Enter  
You are correct!

What is 9 - 2? 5 Enter  
Your answer is wrong  
9 - 2 is 7



2. (**Game: heads or tails**) Write a program that lets the user guess whether the flip of a coin results in heads or tails. The program randomly generates an integer 0 or 1, which represents head or tail. The program prompts the user to enter a guess and reports whether the guess is correct or incorrect.
3. (**Algebra: solve quadratic equations**) The two roots of a quadratic equation  $ax^2 + bx + c = 0$  can be obtained using the following formula:

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

$b^2 - 4ac$  is called the discriminant of the quadratic equation. If it is positive, the equation has two real roots. If it is zero, the equation has one root. If it is negative, the equation has no real roots.

Write a program that prompts the user to enter values for a, b, and c and displays the result based on the discriminant. If the discriminant is positive, display two roots. If the discriminant is 0, display one root. Otherwise, display “The equation has no real roots”.

**Note** that you can use **Math.pow(x, 0.5)** to compute  $\sqrt{x}$ . Here are some sample runs.

```
Enter a, b, c: 1.0 3 1 ↵ Enter
The equation has two roots -0.381966 and -2.61803
```

```
Enter a, b, c: 1 2.0 1 ↵ Enter
The equation has one root -1
```

```
Enter a, b, c: 1 2 3 ↵ Enter
The equation has no real roots
```

