Lab Manual

**CS111L – Programming Fundamentals Lab**

# Lab No: **06**

Topic: Control Structures (if , if else, nested if, nested if/else, switch statements)

Class: **BSSE** Semester:  **1** Session: Fall 2024

# Instructor: Mr. Zain Abubaker

# Lab Instructor: Mr. Zain Abubaker

A blue logo with a bird and a cross

Description automatically generated**Air University Islamabad**

**FACULTY OF COMPUTING & ARTIFICAL INTELLIGENCE**

**Department of Creative Technologies**

# Instructions

**Submission:** Use proper naming convention for your submission file. Name the submission file as Lab\_NO\_DEGREE\_ROLLNUM (e.g. Lab\_01\_BSSE\_00000). Submit the file on Google Classroom within the deadline. Failure to submit according to the above format would result in deduction of 10% marks. Submissions on the email will not be accepted.

**Plagiarism:** Plagiarism cases will be dealt with strictly. If found plagiarized, both the involved parties will be awarded zero marks in the assignment, all of the remaining assignments, or even an F grade in the course. Copying from the internet is the easiest way to get caught!

**Deadline:** The deadlines to submit the assignment are hard. Late submission with marks deduction will be accepted according to the course policy shared by the instructor. Correct and timely submission of the assignment is the responsibility of every student; hence no relaxation will be given to anyone.

**Comments:** Comment your code properly. Bonus marks (maximum 10%) will be awarded to well comment code. Write your name and roll number (as a block comment) at the beginning of the solution to each problem.

Objectives

In this lab, you will learn:

o About Structure ADT.

o How to define a Structure, initialize and refer to individual members of a Structure.

**Tip:** For timely completion of the assignment, start as early as possible. Furthermore, work smartly - as some of the problems can be solved using smarter logic.

**1.** Note: Follow the given instructions to the letter, failing to do so will result in a zero.

# Objectives

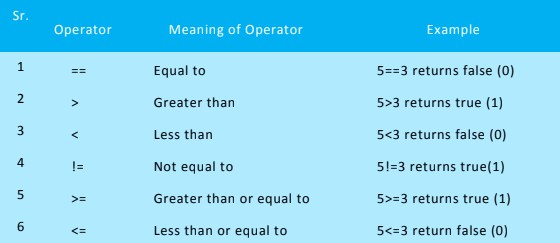
In this lab, you will learn:

* What is selection and control?
* How to use selection and control statement
* How to use Switch statements

Concepts

## Comparison or Relational operators:

You have already seen that the statement total = 5 is an assignment statement; that is, the integer 5 is placed in the variable called total. Nothing relevant to our everyday understanding of equality is present here. So how do we deal with equality in a program? How about greater than or less than? C++ allows the programmer to compare numeric values using relational operators. They are the following:



An expression of the form num1 > num2 is called a relational expression.

**Note** that it does not assert that num1 is greater than num2. It actually tests to see if this is true.

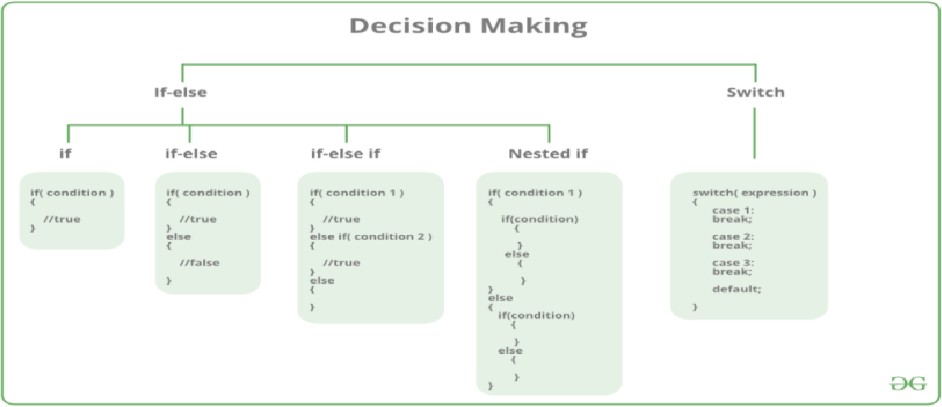
So relational expressions are boolean. Their value must be either true or false.

Relational operators check relationship between two operands. If the relation is true, it returns value 1 and if the relation is false, it returns value 0.

#### For example:

a>b

**Conditional Statements:**

Decision-making or conditional statements in programming languages decide the direction of the flow of program execution. Decision-making statements available C++ are shown in figure below:

### The If Statement:

Sometimes we may only want a portion of code executed under certain conditions.

To do so, we use conditional statements. For example, if you are writing a payroll program to compute wages, then the program should only compute overtime pay if the employee worked more than 40 hours in a given week. Otherwise, when the program is executed the overtime portion of the code should be bypassed. An if statement is one kind of conditional statement.

Consider the following program:

// This program prints "You Pass" if a student's average is 60 or higher and prints

// "You Fail" otherwise #include <iostream> using namespace std: int main()

{

float average;

cout << "Input your average" << endl; cin >> average;

if (average >= 60) // note the use of a relational operator cout << "You Pass" << endl;

if (average < 60)

cout << "You Fail" << endl; return 0;

}

Note that it is not possible for this program to print out both “You Pass” and “You Fail”. Only one of the if statements will be executed. Later we will see a way to write this program without using 2 if statements.

If you want to conditionally execute several statements using if, the following syntax is required:

if (expression)

{

statement\_1; statement\_2;

:

statement\_n;

}

**Note** the curly braces surrounding the set of statements to be conditionally executed.

### The If…else Statement:

If statement can be followed by an optional else statement, which executes when the Boolean expression is false.

In Sample Program given above we used two if statements. A more elegant approach would be to use the if/else statement as follows:

if (average >= 60)

cout << "You Pass" << endl;

else

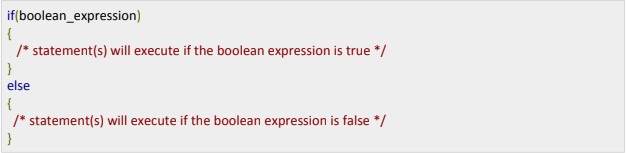
cout << "You Fail" << endl;

In every if/else statement the program can take only one of two possible paths.

Multiple statements can be handled using curly braces in the same way as the if statement.

#### Syntax:

The syntax of if, else statement in C++ programming language is:

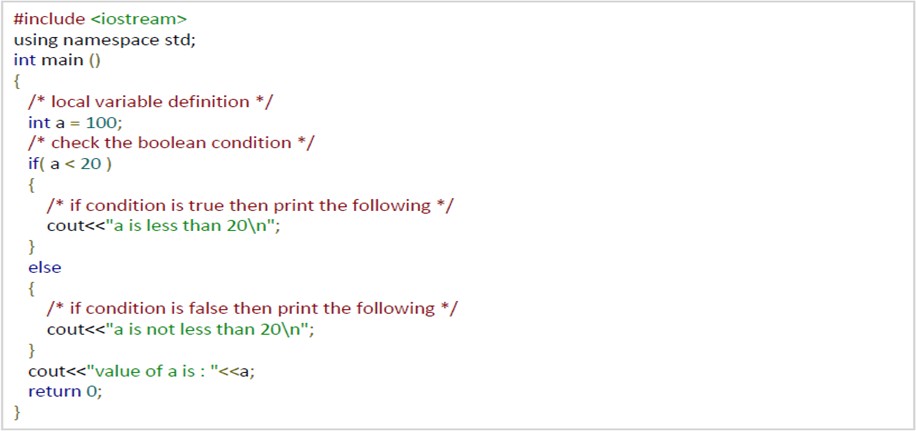


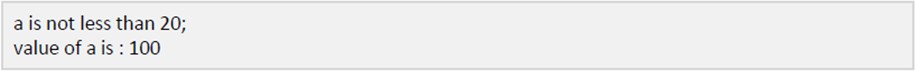
If the Boolean expression evaluates to true, then if block of code will be executed, otherwise

**else block** of code will be executed.

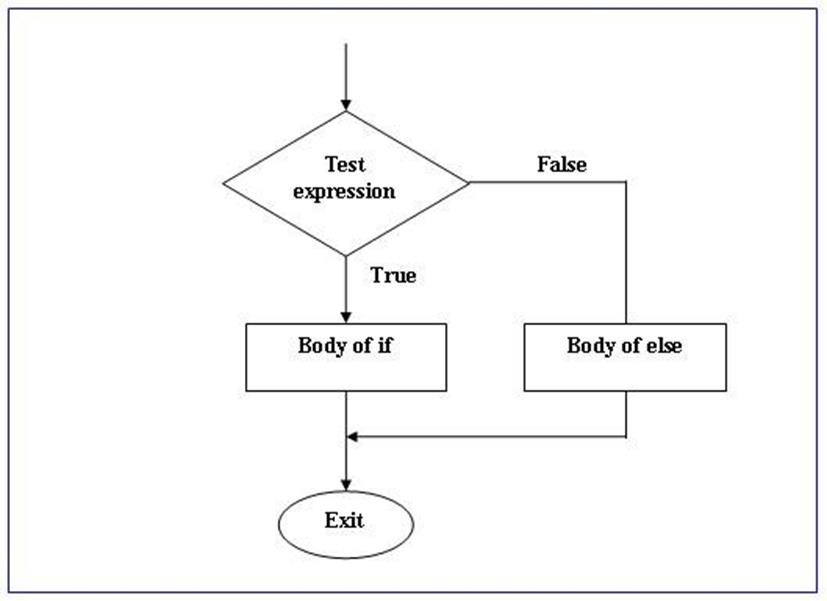
C++ programming language assumes any non-zero and non-null values as true, and if it is either zero or null, then it is assumed as false value.

#### Example:



When the above code is compiled and executed, it produces the following result:

**Flow chart:**



### The if...else if Statement

If statement can be followed by an optional else if...else statement, which is very useful to test various conditions using single if...else if statement.

When using if, else if, else statements there are few points to keep in mind:

* If can have zero or one else's and it must come after any else if's.
* If can have zero to many else if's and they must come before the else.
* Once an else if succeeds, none of the remaining else if's or else's will be tested.

#### Syntax:

The syntax of an if...else if statement in C++ programming language is:

The if/else statement works well if there are only two possible paths to follow. However, what if there are more than two possibilities? For example, suppose we need to decide what kind of vacation to take based on a yearly work bonus:

if the bonus is less than $1,000, we set up a tent and eat hot dogs in the back yard

if the bonus is less than $10,000 and greater than or equal to $1,000, we go to Disney World

if the bonus is $10,000, we go to Hawaii

We could code this using the if/else if statement as follows:

float bonus;

cout << "Please input the amount of your yearly bonus" << endl; cin >> bonus;

if (bonus < 1000)

cout << "Another vacation eating hot dogs on the lawn" << endl; else if (bonus < 10000)

cout << "Off to Disney World!" << endl; else if (bonus == 10000)

cout << "Lets go to Hawaii!" << endl;

Can you explain why the first else if conditional statement does not require a greater than or equal to 1000 condition?

In general we can use as many else if expressions as needed to solve a given problem.

#### The Trailing else:

What happens in the code above if the bonus entered is greater than $10,000?

Actually, nothing will happen since none of the conditional expressions are true in this case. Sometimes it is advantageous to add a final or trailing else at the end of a chain of if/else if statements to handle “all other cases.”

For example, we could modify the code to read: if (bonus < 1000)

cout << "Another vacation on the lawn" << endl; else if (bonus < 10000)

cout << "Off to Disney World!" << endl; else if (bonus == 10000)

cout << "Lets go to Hawaii!" << endl;

else

{

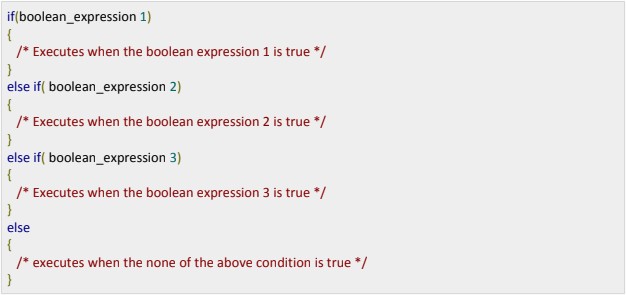
}

cout << bonus << " is not a valid bonus" << endl;

cout << "Please run the program again with valid data" << endl;

// Note the necessary use of the curly brackets here

Of course, few would complain about a bonus greater than $10,000 and the Hawaii trip could still be done on this budget. However, if the maximum possible bonus is $10,000, then the trailing else will let the user know that an illegal value has been entered.



#### Example:

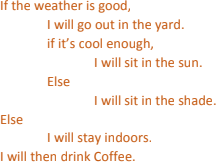
When the above code is compiled and executed, it produces the following result: None of the values is matching

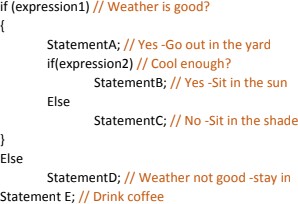
Exact value of a is: 100

### Nested if Statements:

Often programmers use an if statement within another if statement. For example, suppose a software engineering company wants to screen applicants first for experienced programmers and second for C++ programmers specifically.

A nested if in C++ is an if statement that is the target of another if statement. Nested if statements mean an if statement inside another if statement. C++ allows us to nested if statements within if statements, i.e., we can place an if statement inside another if statement.

It’s also possible to have ifs within ifs. These are called nested ifs. For example:

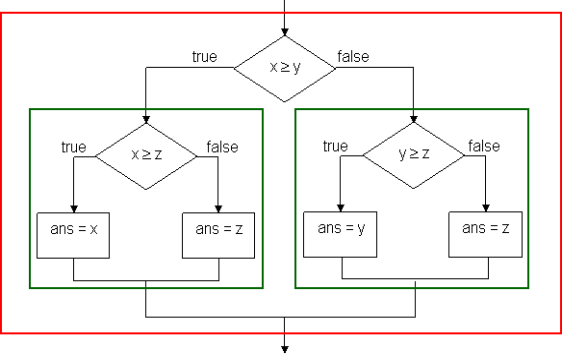
In programming terms, this corresponds to the following:

Here, the second if condition, expression2, is only checked if the first if condition, expression1, is true.

The braces enclosing StatementA and the second if are necessary to make both of these statements a part of what is executed when expression1 is true. Note how the else is aligned with the if to which it belongs.

Note how C++ programmers are identified using a nested if statement. Also note how the trailing else is used to detect invalid input.

**Flow chart:**

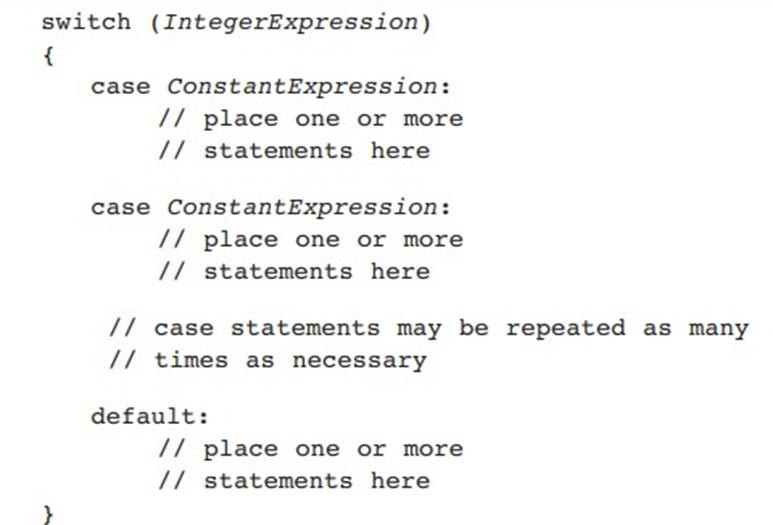


### Switch Statement:

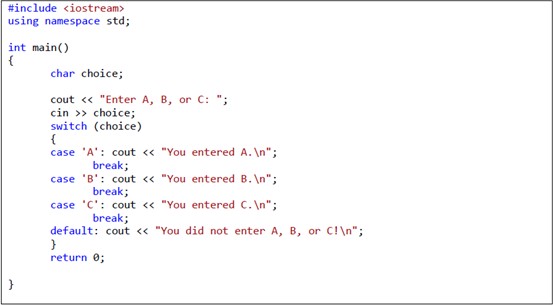
We have already seen how if statements can affect the branching of a program during execution. Another way to do this is using the switch statement. It is also a conditional statement. The switch statement uses the value of an integer expression to determine which group of statements to branch through.

#### Syntax:

Below is the syntax of the switch statement:



#### Example:



Note the use of the curly braces that enclose the cases and the use of break; after each case. Also, consider the variable grade. It is defined as a character data type and the case statements have character arguments such as 'B'. This seems to contradict what we said above, namely that the switch statement uses the value of integer expressions to determine branching. However, this apparent contradiction is resolved by the compiler automatically converting character data into the integer data type. Finally, notice the role of the default statement. The default branch is followed if none of the case expressions match the given switch expression.

## Logical operators:

By using relational operators C++ programmers can create relational expressions.

Programmers can also combine truth values into a single expression by using logical operators. For example, instead of a statement such as “if it is sunny, then we will go outside,” one may use a statement such as “if it is sunny and it is warm, then we will go outside.” Note that this statement has two smaller statements “it is sunny” and “it is warm” joined by the AND logical operator. To evaluate to true, both the sunny and warm requirements must be met.

The NOT operator negates a single statement. For example, “it is sunny” can be negated by “it is not sunny.”

The OR operator is similar to the AND in that it connects two statements.

However, there is an ambiguity about the meaning of the word or in English. In the statement “tonight at 8:00 I will go to the concert in the park or I will go to the stadium to see the ball game,” the word or is exclusive. That is, I can go to the concert or to the game, but not both. However, in the statement “I need to draw an ace or a king to have a good poker hand,” the word or is inclusive. In other words, In can draw a king, an ace, or even both, and I will have a good hand. So we have a choice to make. Let A and B be two statements. A OR B could mean A or B but not both. It could also mean A or B or both. In computer science we use the second meaning of the word or. For example, in the statement “if it is sunny or it is warm, then I will go outside,” there are three scenarios where I will go outside: if it is sunny but not warm, if it is warm but not sunny, or if it is sunny and warm.

The syntax used by C++ for logical operators is the following: AND &&

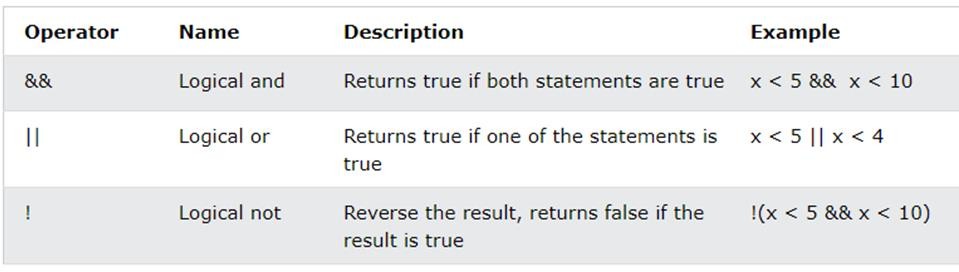
OR | | NOT !

Consider the following:

if (dollars <= 0 || !(accountActive) )

cout << " You may not withdraw money from the bank";

It is good programming practice to enclose the operand after the (!) operator in parentheses. Unexpected things can happen in complicated expressions if you do not. When will this code execute the cout statement? What type of variable do you think accountActive is?



## Tasks:

### Exercise 1:

1. Write a C++ program that takes two integers as input from user and perform the following operations. Note that only one operation can be performed at a time. Ask the user for his choice and perform operation accordingly.

* Choice –1 Bitwise and operator
* Choice –2 Bitwise or operator
* Choice –3 Bitwise XOR operator
* Choice –4 Shift left operator
* Choice –5 Shift right operator
* Choice –6 Addition
* Choice –7 Subtraction
* Choice –8 Multiplication
* Choice –9 Division

1. Write a C++ program for the above scenario using switch statement.

### Exercise 2:

1. Write a C++ program to find the largest number among the four integers. Get the input from user. Use nested if-else statements to solve this question.
2. Write a C++ program to sort the 5 numbers in an ascending order. Get the input of numbers by users. Use if-else-if statements and logical operators.

### Exercise 3:

1. Write a program in C++ that takes a character input and tell that the given character is vowel or consonant. You have solve this by using both if-else statement and switch statement respectively.