SLIIT ACADEMY

Higher Diploma in Information Technology
Year 1, Semester 1

Introduction to Programming(C++)

Lecture 04 : Selection Statements in C++

Intended Learning Outcomes

On the Completion of this lecture student will be able to learn ,

LO1 : Understand the selection statements in C++

LO2: Identify the different selection statements in C++.

SLIIT ACADEMY

Selection Statements in C++

- Selection control structure illustrates a choice between two or more actions, depending on whether a condition is true or false.
- C++ language is supported by two types of selection statements: **if and switch.**
- The condition in the IF statement is based on a comparison of two items and is usually expressed with relational operators.

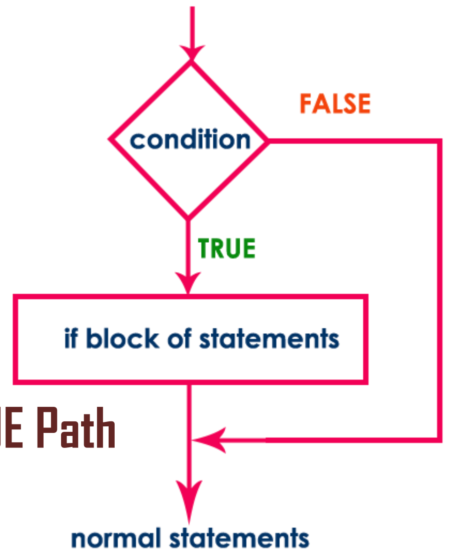
Simple Selection

- The null ELSE statement is used when a task is performed only when a particular condition is true.
- The statement is executed if the logical expression is ***true*** and not executed if the logical expression is ***false***.
- A compound statement can be used for the statement part of the *if* statement.

Syntax

```
if ( condition )
{
    ....
    block of statements;
    ....
}
```

Execution flow diagram



No ELSE , Always evaluates the TRUE Path

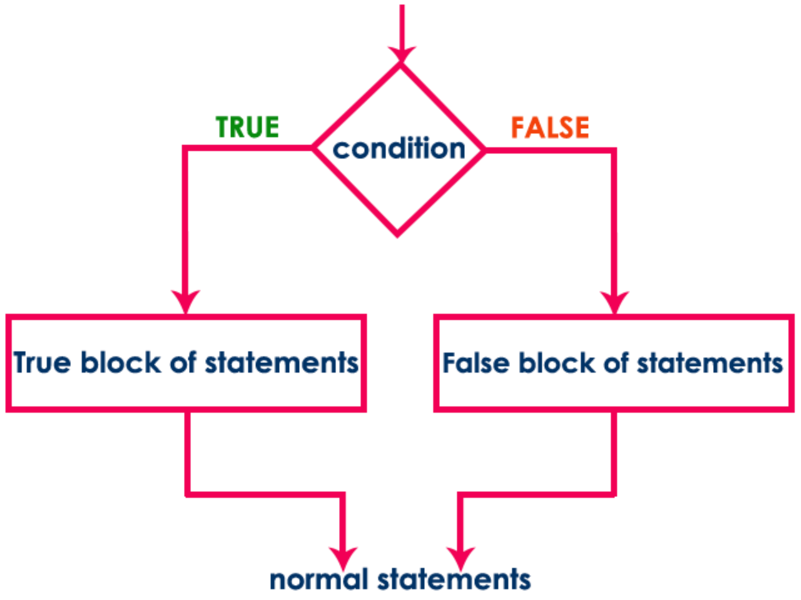
Simple Selection :(Simple IF Statement)

Simple selection occurs when a choice is made between two alternative paths, depending on the result of a condition being true or false.

Syntax

```
if ( condition )
{
    ....
    True block of statements;
    ....
}
else
{
    ....
    False block of statements;
    ....
}
```

Execution flow diagram



Only One Condition can check with Simple IF .

A compound statement can be used for the statement part of the if statement.

Warnings about syntax

- The else must follow immediately after the ***if*** clause.
- You must use a compound statement/block for any clause (***if*** and/or ***else***) when they contains more than one statement.

Combined Selection (Combined IF Statement)

- A combined IF statement is one that contains multiple conditions, each connected with the logical operators AND or OR .
- IF the conditions are combined using the connector AND, both conditions must be true for the combined condition to be true .

```
IF(condition 1){
        statement block;
}
IF(condition 2){
        statement block;
}
IF(condition 3){
        statement block;
}
```

Multiple Conditions can check independently.

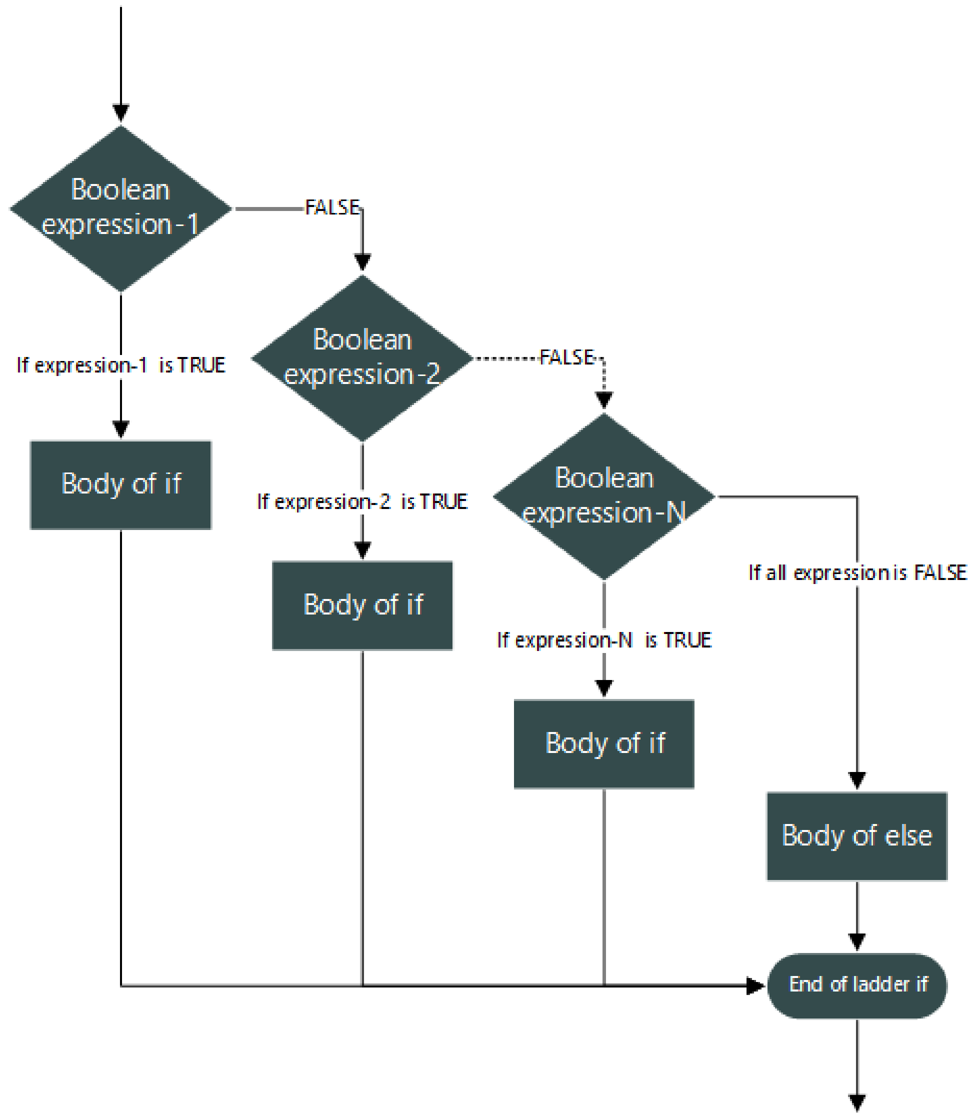
Nested Selection (Nested IF Statement):

- Nested selection occurs when the word IF appears more than once within an IF statement.
- The linear nested IF statement is used when a field is being tested for various values and a different action is to be taken for each value.
- This form of nested IF is called linear because each ELSE immediately follows the IF condition to which it corresponds.

SLIIT ACADEMY

Nested Selection (Nested IF Statement):

An ***if*** or ***if-else*** statement may appear as a statement in the ***if*** or ***else*** clause. In this case the entire ***if*** statement with its ***if*** and ***else*** clause is considered to be one statement.



C++ Switch Statements

- The ***switch*** statement is an alternative to the nested ***if-else*** statement provided the expressions can be written as:

  (variable == value)

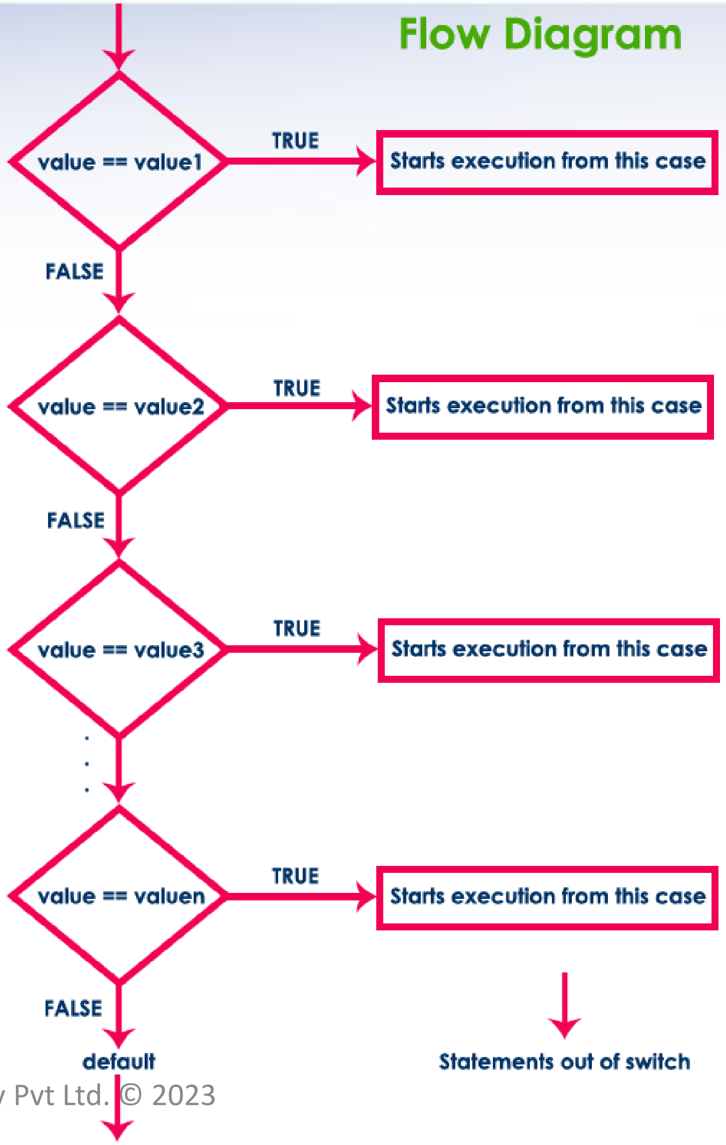
- The ***Switch*** statement allows for execution of multiple statements for a given condition, using the ***break*** statement to terminate execution for the condition.

C++ Switch Statements

Syntax

```
switch ( expression or value )
{
        case  value1:  set of statements;
                        ....
        case  value2:  set of statements;
                        ....
        case  value3:  set of statements;
                        ....
        case  value4:  set of statements;
                        ....
        case  value5:  set of statements;
                        ....
        .
        .
        default: set of statements;
}
```

Flow Diagram


value == value1
TRUE
Starts execution from this case
FALSE
value == value2
TRUE
Starts execution from this case
FALSE
value == value3
TRUE
Starts execution from this case
value == valuen
TRUE
Starts execution from this case
FALSE
default
Statements out of switch

SLIIT ACADEMY

The break statement

- The ***break*** statement causes the ***switch*** statement to terminate and begin execution with the statement after the ***switch*** statement.
- If a ***break*** statement does not appear at the end of a group of statement for a ***case***, processing continues sequentially even though the statements may be specified for another ***case***.
- A break can save a lot of execution time because it "ignores" the execution of all the rest of the code in the switch block.

Summary

- Introduction to selection statements in C++
- Simple selection
- Combined selection
- Nested selection
- Case structure