



National University
of Computer & Emerging Sciences

National University of Computer & Emerging Sciences (NUCES) Islamabad,

Department of Computer Science

Programming Fundamentals– FALL 2023



LAB 07

Conditional Statements

Learning Outcomes

In this lab you are expected to learn the following:

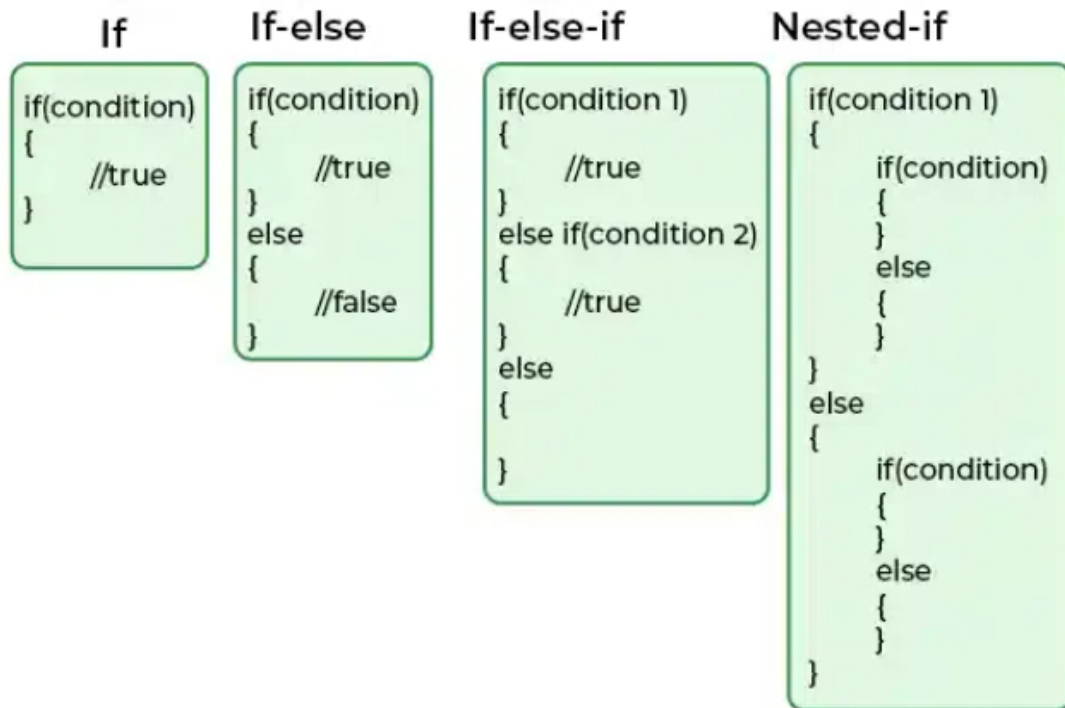
- If else statement
- Switch statement

If else Conditional Statement:

If-else conditional operators are divided into following four types:

1. If
2. If-else
3. If-else-if
4. Nested if

The syntax for each type is shown below:



1. If statement

The if_statement is the most simple decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e. if a certain condition is true then a block of statements is executed otherwise not.

// C++ program to illustrate If statement

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int i = 10;
```

```
    if (i > 15) {
```

```
        cout<<"10 is greater than 15";
```

```
    }
```

```
    Cout<<"I am Not in if";
```

```
    return 0;
```

```
}
```



Output

```
I am Not in if
```

2. If-else statement

The `if` statement executes the first block of code if the condition is true, and the `else` statement executes the second block of code if the condition is false.

```
// C++ program to illustrate if-else statement
#include <iostream>

using namespace std;

int main()
{
    int i = 20;

    if (i < 15)
        cout << "i is smaller than 15";
    else
        cout << "i is greater than 15";

    return 0;
}
```

Output

```
i is greater than 15
```

3. If-else-if statements

The `if-else if-else` ladder executes a specific statement depending on the first true condition encountered, or the `else` statement if all conditions are false.

```
// C++ program to illustrate if-else-if ladder
#include <iostream>

using namespace std;

int main()
{
    int i = 20;

    if (i == 10)
        cout << "i is 10";
    else if (i == 15)
        cout << "i is 15";
    else if (i == 20)
        cout << "i is 20";
    else
        cout << "i is not present";
}
```

Output

```
i is 20
```

4. Nested-if statements

The nested `if-else` structure executes based on `condition1` being true, and within that, it further executes based on `condition2`, with an `else` block for when `condition2` is false.

```
// C++ program to illustrate nested-if statement
#include <iostream>
using namespace std;

int main()
{
    int i = 10;

    if (i == 10) {
        // First if statement
        if (i < 15)
            cout << "i is smaller than 15\n";

        // Nested - if statement
        // Will only be executed if
        // statement above is true
        if (i < 12)
            cout << "i is smaller than 12 too\n";
        else
            cout << "i is greater than 15";
    }

    return 0;
}
```

Output

```
i is smaller than 15
i is smaller than 12 too
```

Switch Statement:

The `switch` case statement offers an alternative to the `if-else if` ladder for conditionally executing code based on a specified variable's value. The `switch` block contains cases corresponding to specific values of the switch variable, defining the code to be executed for each matched value.

The syntax for switch statement is shown below:

```
switch (expression) {  
    case value1:  
        statements;  
    case value2:  
        statements;  
    ....  
    ....  
    ....  
    default:  
        statements;  
}
```

// C++ Program to illustrate the use of switch statement

```
#include <iostream>  
  
using namespace std;  
  
// driver code  
  
int main()  
{  
  
    // variable to be used in switch statement  
  
    int var = 2;  
  
    // declaring switch cases  
  
    switch (var)  
    {  
  
        case 1:  
  
            cout << "Case 1 is executed";  
  
            break;  
  
        case 2:  
  
            cout << "Case 2 is executed";  
  
            break;  
  
        default:
```



```
cout << "Default Case is executed";  
  
break;  
  
}  
  
return 0;  
  
}
```

Output

Case 2 is executed

Note: The value you use in a `switch` statement must be an integer or a character. It can't be any other type of data, like a string or a floating-point number.

Why do we use switch when we have if else if ladder?

if-else-if	switch
1. if-else statement can test for equality as well as for logical expression.	1. switch statement only test for the equality.
2. It uses multiple expressions for multiple choices.	2. It uses single expression for multiple choices.
3. It can check a range of values.	3. It cannot check a range of values.
4. It is less compact than switch.	4. It is more compact than if-else-if.
5. It can evaluate int, float and other data types.	5. It can only evaluate character or integers.

Tasks

Task 1:

You are given a percentage score. Implement a program that classifies the grade based on the following criteria:

90% or above: 'A'

80-89%: 'B'

70-79%: 'C'

Below 70%: 'D'

Write a C++ program that takes the percentage score as input and prints the corresponding grade.



```
Input: 85.5
Output: Grade B
```

Task 2:

You are developing a ticket booking system for a cinema. The ticket price is determined based on the age of the customer. Write a C++ program that takes the age of the customer as input and calculates the ticket price using the following rules:

Children (age 0-12): \$10

Adults (age 13-64): \$20

Seniors (age 65+): \$15

```
Input: 45
Output: Ticket Price: $20
```

Also check for invalid input. **For Example:** if a user enters age as -10, invalid age message should be displayed.

Task 3:

Write a program that takes an integer as input and classifies it into one of the following categories: (even or odd), and (positive, negative, or zero).

Write a C++ program that takes an integer as input and prints the corresponding classification.

```
Input: -5
Output: Negative and Odd
```

Task 4 (Using Nested if):

You are tasked with creating a program that determines a student's eligibility for enrolling in a course based on certain criteria. Students are eligible to enroll if they meet the following conditions:

The student's age is between 18 and 30 years (inclusive).

The student has passed the prerequisite exam, obtaining a score of at least 60 out of 100.

If the student is below 21 years old, they must have parental consent.

Write a C++ program that takes the student's age, exam score, and whether they have parental consent as input. Based on the input, determine and print whether the student is eligible to enroll in the course.

```
Input: Age = 20, Exam Score = 75, Parental Consent = Yes
Output: Eligible to enroll in the course
```

Task 5 (Using Switch):

Implement a simple calculator that performs addition, subtraction, multiplication, or division based on user input.

Write a C++ program that takes two operands and an operator (+, -, *, /) as input and performs the corresponding operation.

```
Input: Operand1 = 10, Operand2 = 5, Operator = *
Output: Result = 50
```

Task 6 (Using Nested Switch):

You are tasked with creating a C++ program that simulates the game of "Rock, Paper, Scissors" between two players. Each player will choose one of the three options: rock (1), paper (2), or scissors (3). The program should determine the winner based on the game rules, where:

Paper (2) dominates rock (1)

Rock (1) dominates scissors (3)

Scissors (3) dominate paper (2)

Implement a C++ program using a switch statement that takes the choices of the first player and the second player as input (represented by 1, 2, or 3) and determines the winner of the game based on the choices made.

```
Input: First player = 3, Second player = 1
Output: 1st player wins
```

```
Input: First player = 1, Second player = 2
Output: 2nd player wins
```

Task 7 (Using Switch):

You are tasked with building a program that facilitates unit conversion between different units of measurement. The program should be able to convert units such as inches to centimeters and pounds to kilograms, allowing users to input the unit type as characters ('I' for inches, 'P' for pounds) and the value to convert.

Write a C++ program that takes the unit type as a character and the value as input and performs the necessary unit conversion, providing the converted value as output.

The program supports the following unit conversions:

Inches to Centimeters: Multiply the input value (in inches) by 2.54 to obtain the equivalent value in centimeters.

Pounds to Kilograms: Multiply the input value (in pounds) by 0.453592 to obtain the equivalent value in kilograms.

Ensure that the program handles invalid unit types gracefully and prompts the user to enter a valid unit type.

Your program should output the converted value along with the appropriate unit.

```
Input: Unit Type = 'I', Value = 12
Output: Converted Value = 30.48 cm
```

Submission Instructions:

1. Save all **.cpp** files with your roll no and task number

e.g. i22XXXX_Q1.cpp

i22XXXX_Q2.cpp

2. Now create a new folder with name *ROLLNO_LAB06* e.g. i22XXXX_LAB06

3. You need to display your **roll no** and **name** before the output of any question.

4. Move all of your **.cpp files** and the **screenshots** of the output of each **.cpp** file to this newly created directory and compress it into a **.zip file**.