Programming Fundamentals

Learning Outcomes:

- Students should be able to think logically and develop problem-solving skills
- Students should be able to define the need for conditional statements.
- Students should be able to implement the learned concepts of Multiple and Nested If Statement
- Students should be able to define problem-solving through programming concepts

Nested IF vs Using Logical Operators

Introduction

In our daily life, we encounter many situations where we have to decide from multiple options. For example, whether to buy a dress or not? Or Whether to go to university today or not?

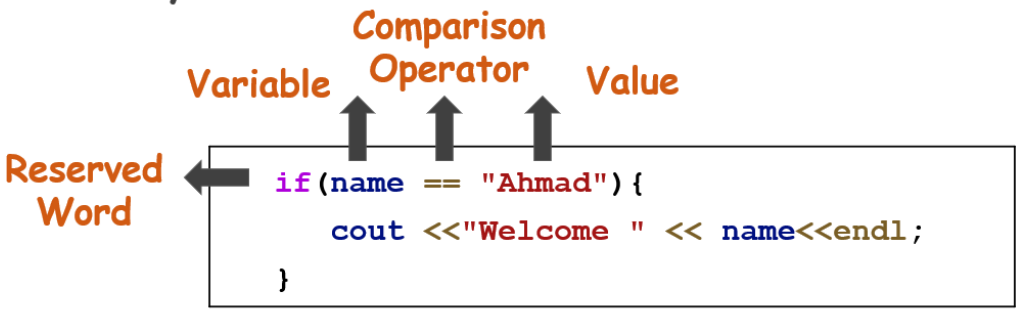
Even these simple phenomena pose us with these optional solutions that can be opted depending upon different conditions. For example, **IF** the cost of the dress is more than 1500 **Then** you will buy that dress. OR **IF** your friends are going to university **Then** you will go to the university.

Similarly, we are faced with similar problems in programming as well. Where we, as programmers, have to write programming code to solve such problems. Therefore, for such problems, we make use of the **IF Statement**.

We have learned about solving such issues in the last class through various examples. However, in this class, we shall increase the complexity of the problems and then try to solve these issues using the previously learned concepts.

But First, Recall these concepts that you were taught in the earlier class.

- IF statement
- Body of IF statement



Using Nested IF

For example, let's introduce a little bit of complexity in the previous examples.

Example #1:

IF it is an MTJ kurta and **IF** the cost of the dress is less than 1500 **Then** you will buy that dress.

Solution:	The code produces the following output.

```
#include<iostream>

using namespace std;

int main()
{

int cost;
string brand;
cout<< "Enter the cost of the dress: ";
cin >> cost;
cout<< "Enter the brand of the dress: ";
cin >> brand;

if (cost < 1500)
  if( brand == "MTJ" ){
        cout<< "Buy the dress";
}
}
```

```
D:\Codes>c++ test.cpp -o test.exe

D:\Codes>test.exe
Enter the cost of the dress: 1300
Enter the brand of the dress: MTJ
Buy the dress
D:\Codes>
```

Example #2:

IF you have class today and **IF** your friends are going to university **Then** you will go to the university.

Solution:	The code produces the following output.

```cpp
#include<iostream>

using namespace std;

int main()
{

char going;
char lecture;
cout<< "Do you have class today (press Y for yes): ";
cin >> going;
cout<< "Are your friends going? (press Y for yes): ";
cin >> lecture;

if ( going == 'y' )
  if( lecture == 'y' ){
        cout<< "You are also going";
}
}
```

```
D:\Codes>c++ test.cpp -o test.exe

D:\Codes>test.exe
Do you have class today (press Y for yes): y
Are your friends going? (press Y for yes): y
You are also going
D:\Codes>
```

Example #3:

Consider that Ali is a student, who recently got admission to UET. We need to write a program to congratulate **IF** his roll number is 501 and **IF** he is Ali.

Solution:

```cpp
#include<iostream>

using namespace std;

int main()
{

string name;
int number;

cout<< "Please Enter your name: ";
cin >> name;
cout<< "Please Enter your roll no: ";
cin >> number;

if (name == "ali")
  if(number == 501){
        cout<< "Welcome "<< name<< endl;
}
}
```

The code produces the following output.

```
D:\Codes>c++ test.cpp -o test.exe

D:\Codes>test.exe
Please Enter your name: ali
Please Enter your roll no: 501
Welcome ali

D:\Codes>
```

Congratulations, you have performed all the tasks using the Nested IF Statements.

Using Logical operators

Just like, in the last class, we learned how the same operations implemented by Multiple IF statements can also be implemented through IF ELSE Block. Similarly, we can also perform the above-mentioned operations by using the comparison operators.

Example #1:

IF it is an MTJ kurta and **IF** the cost of the dress is less than 1500 **Then** you will buy that dress.

Solution:	The code produces the following output.

`#include<iostream>`
 	
`using namespace std;`
 	
`int main()`
`{`
 	
`int cost;`
`string brand;`
`cout<< "Enter the cost of the dress: ";`
`cin >> cost;`
`cout<< "Enter the brand of the dress: ";`
`cin >> brand;`
 	
`if (cost < 1500 && brand == "MTJ"  ){`
`cout<< "Buy the dress";`
`}`
`}`
	`D:\Codes>c++ test.cpp -o test.exe`
	
	`D:\Codes>test.exe`
	`Enter the cost of the dress: 1300`
	`Enter the brand of the dress: MTJ`
	`Buy the dress`
	`D:\Codes>`

Example #2:

IF you have class today and **IF** your friends are going to university **Then** you will go to the university.

Solution:	The code produces the following output.

`#include<iostream>`
 	
`using namespace std;`
 	
`int main()`
`{`
 	
`char going;`
`char lecture;`
`cout<< "Do you have class today (press Y for yes): ";`
`cin >> going;`
`cout<< "Are your friends going? (press Y for yes): ";`
`cin >> lecture;`
 	
`if ( going == 'y' && lecture == 'y' ){`
`cout<< "You are also going";`
`}`
`}`
	`D:\Codes>c++ test.cpp -o test.exe`
	
	`D:\Codes>test.exe`
	`Do you have class today (press Y for yes): y`
	`Are your friends going? (press Y for yes): y`
	`You are also going`
	`D:\Codes>`

Example #3:

Consider that Ali is a student, who recently got admission to UET. We need to write a program to congratulate **IF** his roll number is 501 and **IF** he is Ali.

Solution:	The code produces the following output.

```
#include<iostream>

using namespace std;

int main()
{

string name;
int number;

cout<< "Please Enter your name: ";
cin >> name;
cout<< "Please Enter your roll no: ";
cin >> number;

if (name == "ali" && number == 501) {
        cout<< "Welcome "<< name<< endl;
}
}
```

```
D:\Codes>c++ test.cpp -o test.exe

D:\Codes>test.exe
Please Enter your name: ali
Please Enter your roll no: 501
Welcome ali

D:\Codes>
```

Congratulations, you have practiced and learned the fundamental concepts of Nested IF and Comparison Operators.

Let's start the challenges now.

Challenge 1:

A Store has announced to give the 10% discount on the 5000 or more purchase amount on every Sunday. Write a program that takes Day, total purchase amount, and output the payable amount.

Challenge 2:

A Store has announced to give the 10% discount on the total purchase greater than 5000 amount of every Sunday and 5% on every other day. Write a program that takes Day, total purchase amount and outputs the payable.

Challenge 3:

A Store has announced to give the 10% discount on the total purchase amount of every Sunday of Month October and March Write a program that takes Day, Month, total purchase amount and outputs the payable (amount-discount).

Challenge 4:

A Store has announced to give the 10% discount on the total purchase amount of every sunday of Month October, March and August (amount-discount) and 5% discount on the total purchase amount of every monday of November and December.Write a program that take Day, Month and total purchase amount and output the payable (amount-discount).

Challenge 5:

A Store has announced to give the 10% discount on the total purchase amount of every Sunday of Month October and 10% discount on the total purchase amount of every Thursday of November.

Challenge 6:

You are appointed as a customs officer at the airport. Your task is to impose tax on the luggage of passengers. There are two categories of ticket business class(weight: 40 kg, price=15000) and economy class(weight: 25 kg , price=8000).If the passenger wants to take luggage more than mentioned weight ,he/she will charge an extra amount (1000 per kg for business class, 500 per kg for economy class). Program should display a total amount that will be applicable to passengers.

Challenge 7:

Create a C++ program that takes three integer numbers as input and displays the amount of integers which are of equal value. Your program must return 0, 2 or 3.

Test Cases:

Input: 3, 4, 3 Output → 2
Input: 1, 1, 1 Output → 3

Input: 3, 4, 1 Output → 0

Challenge 8:
Create a C++ program which takes two strings (which represent player 1 and 2) as input from the user and returns a string stating the winner in a game of *Rock, Paper, Scissors*.

Each input will contain a single string: "Rock", "Paper", or "Scissors". Display the winner according to the following rules:

- **Rock** beats **Scissors**
- **Scissors** beats **Paper**
- **Paper** beats **Rock**

If p1 wins, Display the string "The winner is p1". If p2 wins, display the string "The winner is p2" and if p1 and p2 are the same, return "It's a draw".

Test Cases:

Input: "Rock", "Paper" Output → "The winner is p2"

Input: "Scissors", "Paper" Output → "The winner is p1"

Input: "Paper", "Paper" Output → "It's a draw"