Republic of the Philippines
CEBU TECHNOLOGICAL UNIVERSITY
MAIN CAMPUS
M.J. Cuenco Ave. Cor. R. Palma Street, Cebu City, Philippines
Website: http://www.ctu.edu.ph

COLLEGE OF COMPUTER, INFORMATION AND COMMUNICATIONS TECHNOLOGY

Microsoft

ASP.NET

Module 5

Conditional Statement

I. Preparations

At the end of this module students will:

- construct simple condition and compound condition
- construct solution using conditional statements;

II. Presentation

Conditional statements in C# programming are used to make decisions based on the conditions. Conditional statements execute sequentially when there is no condition around the statements. If you put some condition for a block of statements, the execution flow may change based on the result evaluated by the condition. This process is called decision making in 'C.'

There are four (4) structure of branching

- if () statement
- if-else statement
- if-elseif statement
- switch statement

The *if()* statement

The *if()* statement is simplest form of condition, it is also known as *one-way branching*. It specifies a block of C# code to be executed if the result of the condition is TRUE.

Syntax:

```
if (condition) {
        code to be executed; true block
}
```

The *if() else* Statement

The *if() else* statement is also known as *two-way branching*. When the result of the condition is false, the *else* block of C# code will be executed.

Syntax:

```
if (condition) {
        code to be executed; true block
}
else{
        code to be executed; false block
}
```

The *if() elseif()* Statement

The *if() elseif()* statement is also known as *ladder-if*. When there are two(2) more than conditions, this statement is utilized. If the result of the conditions is *false*, an *else* statement can be added at the bottom for the false statement.

Syntax:

```
if (condition1) {
        code to be executed; true block
}
else if (condition2) {
        code to be executed; true block
}
else if (condition3) {
        code to be executed; true block
}
else if (condition4) {
        code to be executed; true block
}
else if (conditionN) {
        code to be executed; true block
}
else {
        code to be executed; true block
}
```

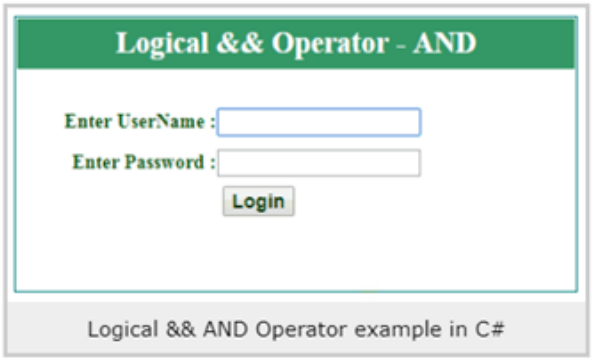
The *switch()* statement

In C#, a switch statement is a control flow statement that allows you to compare a single value against multiple possible values and execute different code depending on the result.

```
switch (expression)
{
   case value1:
      // code to execute if expression == value1
      break;
   case value2:
      // code to execute if expression == value2
      break;
         // additional case statements as needed
   default:
         // code to execute if expression does not match any of
         the case values
      break;
}
```

Logical AND (&&) operator

Example



Here, **&&** operator is use in **if condition** to check both username and password are right or wrong. Above c# .net example first input **username=Welco** and **password=123** then result

is **Welcome to System** and try **username=Welco** and **password=234** that time result will be **Invalid Credential** because password value is wrong.

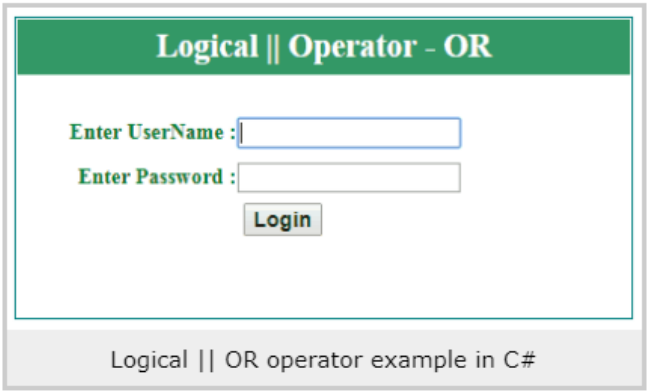
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Opertorss : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void btnlogin_Click(object sender, EventArgs e)
    {
        if (txtuname.Text == "Welco" && txtupass.Text == "123")
        {
            lblmsg.Text = "Welcome to System";
        }
        else
        {
            lblmsg.Text = "Invalid Credential";
        }
    }
}
```

AND Operator

Logical OR (||) Operator

Unlike in && operator that both conditions must be **TRUE** but in || operator it needs only **one true condition** to perform the true statements.

Example:



Logical || OR operator example in C#

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Opertorss : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void btnlogin_Click(object sender, EventArgs e)
    {
        if (txtuname.Text == "admin" || txtupass.Text ==
"123")
        {
            lblmsg.Text = "Welcome to System";
        }
        else
        {
            lblmsg.Text = "Invalid Credential";
        }
    }
}
```

Logical NOT (!)

In NOT (!) operator, if the condition is **TRUE** then logical not operator returns **FALSE**.

Assume int variable **X=10** and variable **Y=5** then

```
if (X != Y)
{
result = “X is not equal to Y”;
}
else
{
result = “X is equal to Y”;
}
```

Here, output is **result = "X is not equal to Y"** because value of X and Y not same so return true block.

III. Practice

Create a C# webforms that will accept three (3) grades (prelim, midterm, and temporary final). Display the average grade as the final grade and the corresponding equivalent description.

Grade	Description
Below 60	Failed
60 – 69	Poor
70 – 79	Average
80 – 89	Good
90 and above	Excellent

IV. Performance

Create a C# webforms that will mimic a Payroll system. The regular working hours is 120. Regular rate per day is 490.88 if the employee is (R) Regular, 420.30 for (P) Probationary, 380.56 for (C) Casual and 300.10 for (PT) Part Timers. The rate per overtime hours is 1 ½ of its regular rate per hour. The withholding tax is 15.75% of the gross earnings if the status is (S) Single, 10.12% for "M" Married, 12.35% for "W" Widow, otherwise 12.60%. The SSS deduction is 11.5% if the gross earning is greater or equal to 12,000.00 and 9.16% if lesser than 9,500.00, otherwise 10.5%. PhilHealth deduction is 420 if gross earnings is greater or equal to 12,000.00 and 290 if lesser than 9,500, otherwise 380. Lastly, PagIbig is 3.75% of the gross earning for married, 2.75%, Single, else 2.55%.

Sample Output: