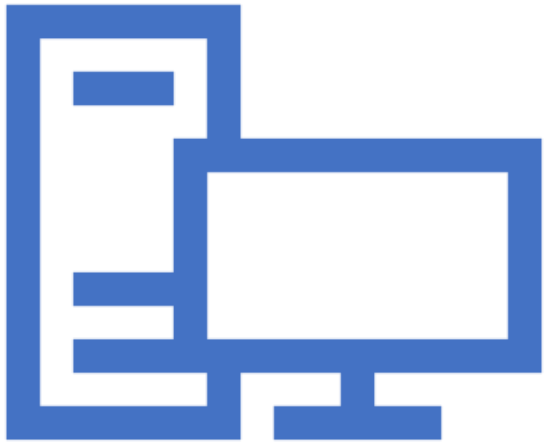
PROGRAMMING SKILLS II

~ ANURUDDHA ABEYSINGHE

Lecture 06

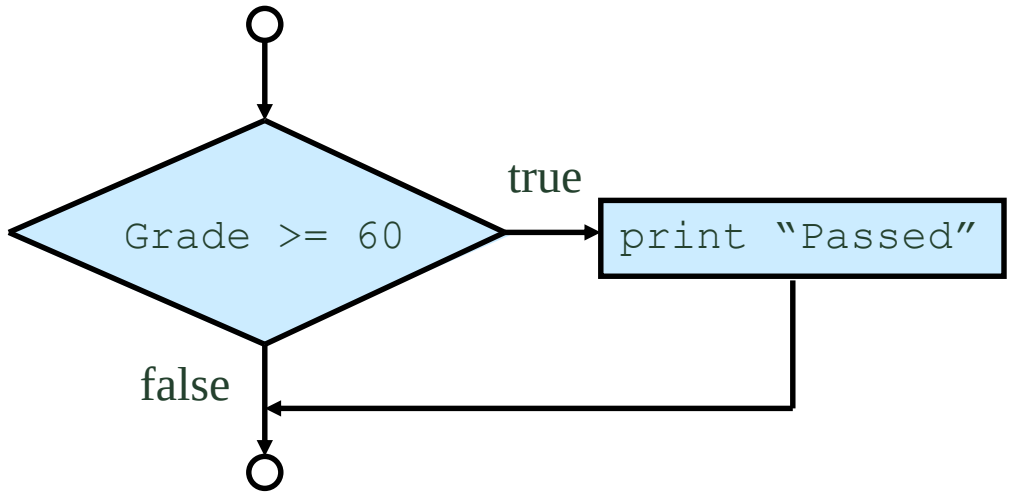
Foundation Certificate in IT – Curtin Batch



IF STATEMENT AND LOGICAL/BOOLEAN OPERATIONS

LECTURE 06

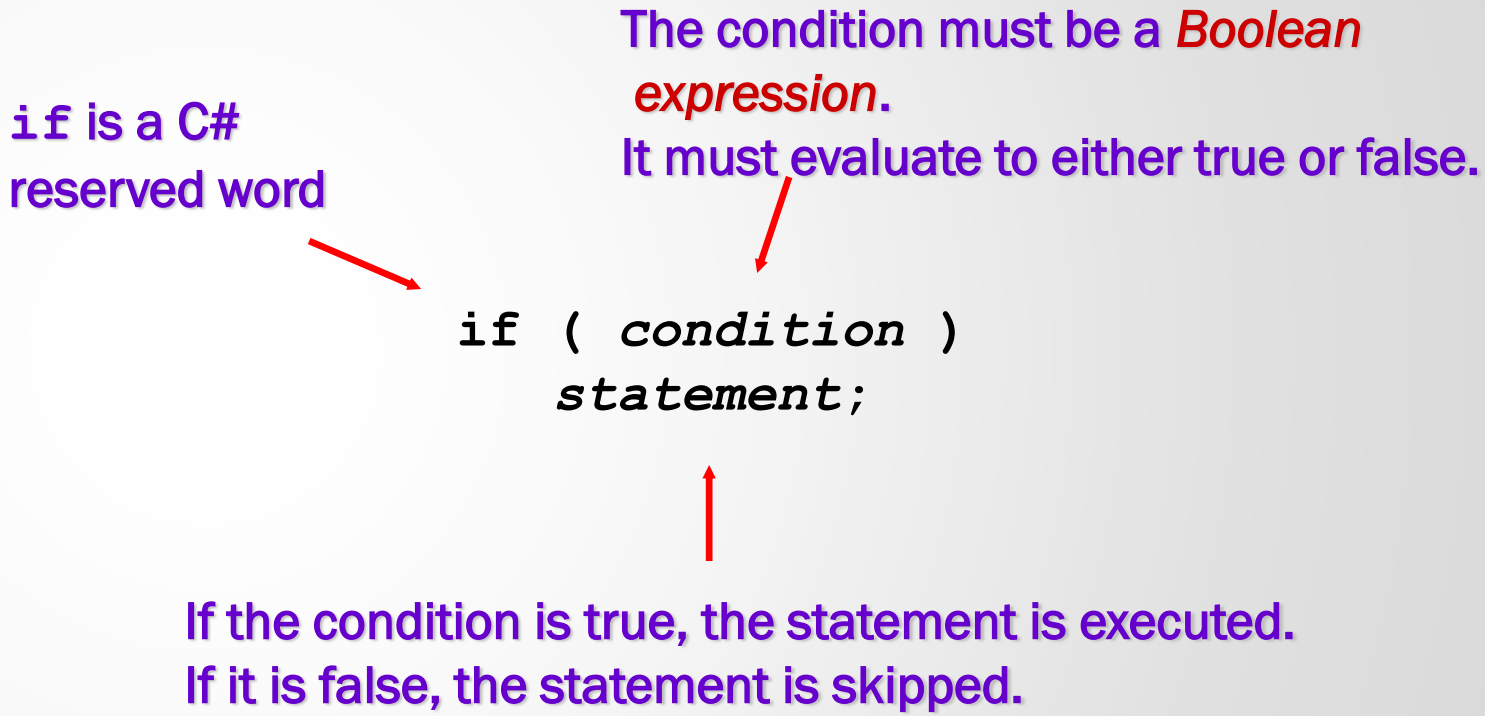
IF SELECTION STRUCTURE (CONT'D)




```
if (studentGrade >= 60)
  Console.WriteLine ("Passed");

// beginning of the next statement
```

THE `IF` STATEMENT

- The *`if` statement* has the following syntax:



BOOLEAN EXPRESSIONS: BASICS

- A condition often uses one of C#'s *equality operators (==, !=)* or *relational operators (<, >, <=, >=)*, which all return boolean results:

==	equal to
!=	not equal to
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to

EQUALITY AND RELATIONAL OPERATORS

Standard algebraic equality operator or relational operator	C# equality or relational operator	Example of C# condition	Meaning of C# condition
Equality operators			
=	==	x == y	x is equal to y
≠	!=	x != y	x is not equal to y
Relational operators			
>	>	x > y	x is greater than y
<	<	x < y	x is less than y
≥	>=	x >= y	x is greater than or equal to y
≤	<=	x <= y	x is less than or equal to y

Equality and relational operators.

Note the difference between the equality operator (==) and the assignment operator (=)

MORE COMPLEX (COMPOUND) BOOLEAN EXPRESSIONS: LOGICAL OPERATORS

- Boolean expressions can also use the following *logical and conditional operators*:

!	Logical NOT
&	Logical AND
|	Logical OR
^	Logical exclusive OR (XOR)
&&	**Conditional AND**
||	**Conditional OR**

- They all take boolean operands and produce boolean results

COMPARISON: LOGICAL AND CONDITIONAL OPERATORS

- Logical **AND (&)** and Logical **OR (|)**
  - Always evaluate both conditions
  - Ex 1: int a=5, b = 2;

    if ((a==5) | (a>b)) → T | T → T
  - Ex 2: int a=5, b = 2;

    if ((a==5) & (a>b)) → T & T → T

- Conditional AND (&&) and Conditional OR (||)
  - Would not evaluate the second condition if the result of the first condition would already decide the final outcome.
  - Ex 1: int a=5, b = 2;

    if ((a==5) || (a>b)) → T || ..... → T
  - Ex 2: int a=5, b = 2;

    if ((a==5) && (a>b)) → T && T → T

PROGRAMMING EXERCISE 1

- Create a program that prompts a user for input, accepts a letter, and then evaluates whether that input is a vowel.

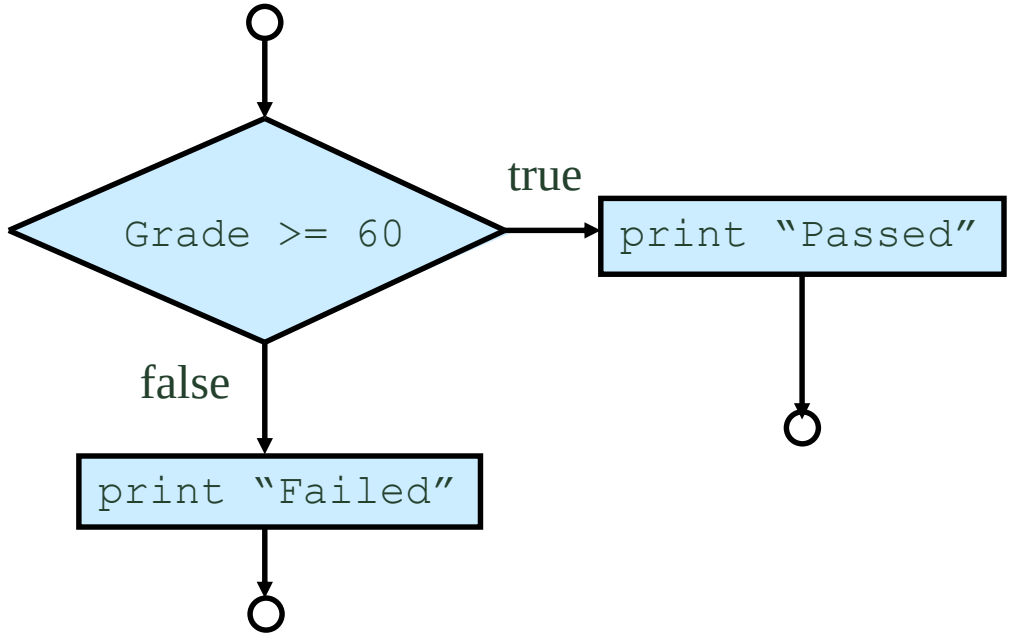
Ex:

```
Enter the letter:      a
a is a Vowel.
Enter the letter:      p
```

CODE – PROGRAMMING EXERCISE 1

```
using System;
class Test1{
    public static void Main(string[] args){
        char vowel;     //create a char variable to assign the single character
        Console.WriteLine("Enter the Letter :");
        vowel = Convert.ToChar(Console.ReadLine());     // get the user input and convert string to char

        if(vowel == 'a' || vowel == 'e' || vowel == 'i'||vowel == 'o'||vowel == 'u'){
//check the possibilities with vowel characters
            Console.WriteLine("{0} is a vowel", vowel);
        }

    }
}
```

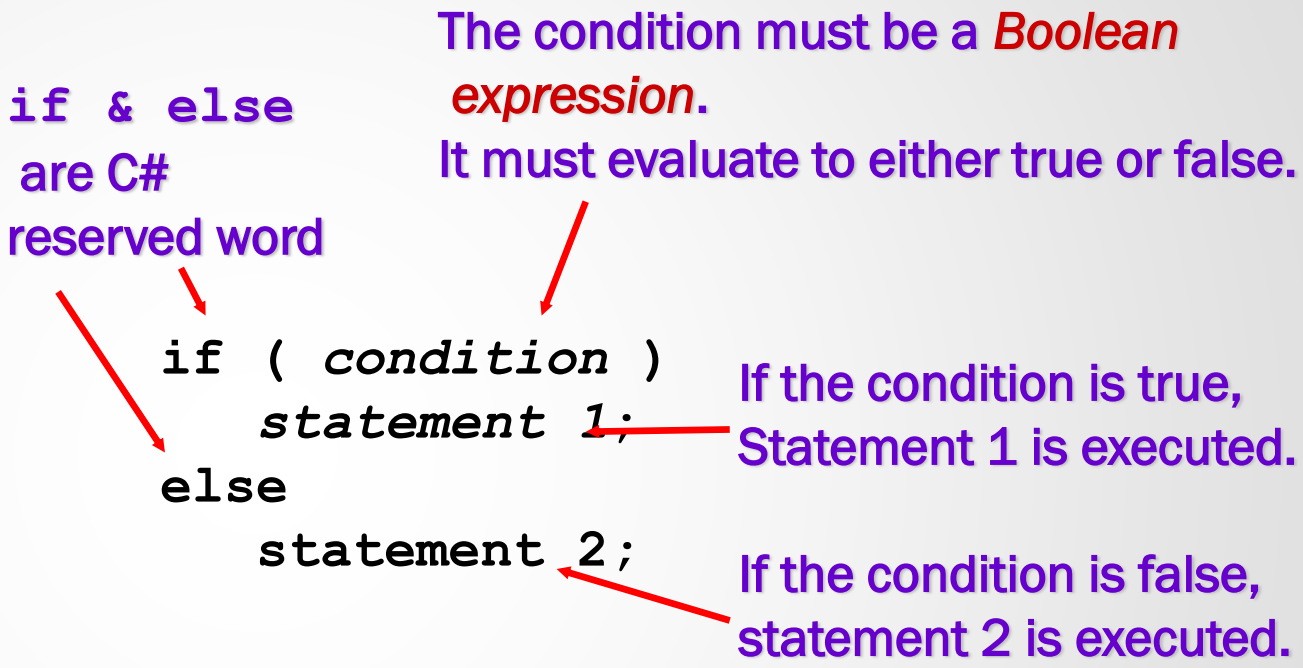
IF – ELSE **SELECTION STRUCTURE**




```
if (studentGrade >= 60)
    Console.WriteLine ("Passed");
else
    Console.WriteLine ("Failed");
```

THE IF- ELSE STATEMENT

- The *if statement* has the following syntax:



PROGRAMMING EXERCISE 2

- Write a program to check the eligibility of voting. The age limit is 18 years. If the age exceed the limit, print as “Eligible for voting”. If not print as “Not eligible for voting”

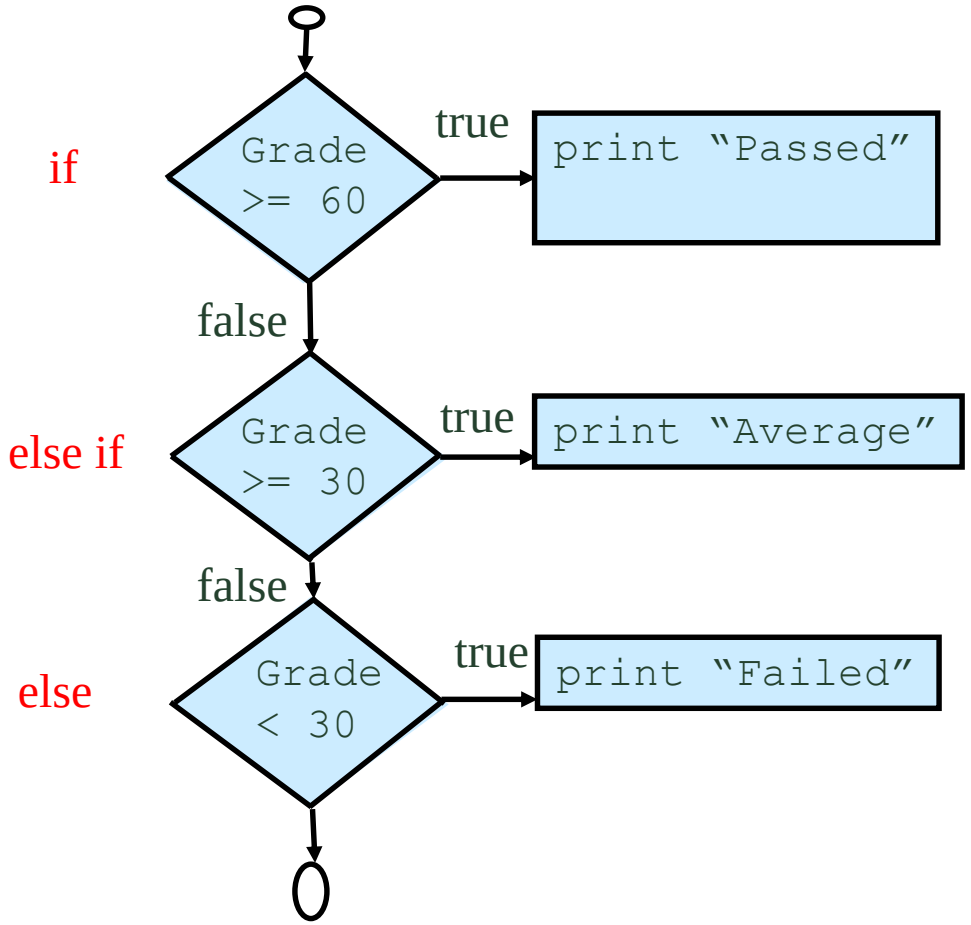
Ex:

```
Enter the Age: 20
Eligible for Voting.


Enter the Age: 15
Not Eligible for Voting.
```

CODE – PROGRAMMING EXERCISE 2

```
using System;
class Test1{
    public static void Main(string[] args){
        int age; //create a int variable to assign the age
        Console.Write("Enter the Age: ");
        age = int.Parse(Console.ReadLine()); // get the age and convert it to int
        if (age >= 18){
            Console.WriteLine("Eligible for voting");   //if the condition is true, this output will display
        }
        else {
            Console.WriteLine("Not Eligible for Voting"); //if the condition is false, this output will display

    }
}
```

IF – ELSE IF **SELECTION STRUCTURE**




```
if (studentGrade >= 60)
  Console.WriteLine ("Passed");
else if (studentGrade >= 30)
  Console.WriteLine ("Average");
else
  Console.WriteLine ("Failed");
```

THE `IF-ELSE IF` STATEMENT

- The *`if-else if` statement* has the following syntax:

The condition must be a *Boolean expression*.
It must evaluate to either true or false.

`else if` is a C# reserved word

```
if ( condition )
    statement;
else if (condition)
    statement2;
else if (condition)
    statement3;
else
    statement4;
```

Same theory ; If the condition is true, the statement is executed.
If it is false, the statement is skipped.

PROGRAMMING EXERCISE 3

- Body mass index (BMI) is a measure of body fat based on height and weight that applies to adult men and women.
- Write a program to identify the category of given BMI value and output the category.

Mark Range	Grade
<=18.5	Under Weight
18.5-24.9	Normal Weight
25 -29.9	Over Weight
>=30	Obesity

```
using System;
class DisplayGrade{
    public static void Main(string[] args){
        double value;   // Initialize values
        Console.WriteLine("Enter BMI value:");
        value = double.Parse(Console.ReadLine());

        if(value<=18.5){
            Console.WriteLine("Under Weight");
        }
        else if(value >= 18.5 && value <25){
            Console.WriteLine("Normal Weight");
        }
        else if(value >= 25 && value < 30){
            Console.WriteLine("Over Weight");
        }
        else{
            Console.WriteLine("Obesity");
        }
 }
}
```

// Get the user inputs

THANK YOU

SEE YOU NEXT WEEK