## IF() STATEMENTS

The if() statement is a **conditional control structure**, also called a **decision structure**, which executes a set of statements when a condition is true.

The if() statement takes the form:

```
if (<logical expression>)
{
   //this code executes if the logical expression is true
}
```

**EXAMPLE**

```
if (guess == 7)
{
   lblMessage.text = "You guessed it!";
}
```

The double equal sign is used as a relational operator to determine if the value of guess is equal to 7. If equal, then Text property of lblMessage is changed. If the expression is **false**, the code in the curly braces is skipped, and the program continues on the next line below the ending curling brace.

If you do not use any curly braces, the if statement will only execute the very first statement.

## COMPARISON OPERATORS

The condition of an if…istatement is a **Boolean expression**, which evaluates to either True or False. *Comparison operators* can be used to form Boolean expressions. There are six comparison operators:

| Operator | Meaning |
|---|---|
| == | equal to |
| < | Less than |
| <= | Less than or equal to |
| > | greater than |
| >= | greater than or equal to |
| != | not equal to |

## IF(), ELSEIF, ELSE STATEMENTS

The if() statement can include an optional Else clause that is executed when the if condition evaluates to **False**.

The if(), else statement takes the form:

```
if (<logical expression>)
{
   //this code executes if the logical expression is true
}
else
{
   //this code executes if the logical expression is false
}
```

**EXAMPLE**

```
if (guess == secret)
{
   lblMessage.text = "You guessed it!";
}
else
{
   lblMessage.text = "Try Again";
}
```

You can use the **else if** keywords to evaluate an expression only if the previous **if** expression was **false**. You may add a final **else** keyword as well. This **else** statement will not use the logical expression. Instead, it will designate statements to be executed if none of the prior **if** or **else if** statements are **true**.

```
if (<1st Logical expression>)
{
   //this code executes if the logical expression is true
}
else if (<2nd Logical expression>)
{
   //this code executes if the 1st logical expression is false
   //AND if the 2nd logical expression is true
}
else
{
   //this executes if all the above logical expressions are false
}
```

**EXAMPLE**

```
if (guess == secret)
{
   lblMessage.text = "You guessed it!";
}
else if (guess > secret)
{
   lblMessage.text = "Too High";
}
else
{
   lblMessage.text = "Try Again";
}
```

## NESTED IF..THEN..ELSE STATEMENTS

An If…Then…Else statement can contain another If..Then…Else or If..Then statement, which is said to be nested. Nested statements execute only when the branch it is in is executed.

```
if (guess == secret)
{
   lblMessage.text = "You guessed it!";
}
else
{
    if (guess > secret)
    {
        lblMessage.text = "Too High";
    }
    else
    {
        lblMessage.text = "Too Low";
    }
}
```

The logic used in developing the If…Then…Elseif statement is important. For example, when testing a range of numbers, if conditions must be properly ordered because ONLY the code associated with the FIRST true condition is executed. The rest of the code is skipped over.

## LOGICAL OPERATORS

| LOGICAL OPERATOR | C# SYMBOL | EXAMPLE | DESCRIPTION |
|---|---|---|---|
| AND | && | A && B | Returns **TRUE** if both “A” and “B” are **TRUE**, or **FALSE** otherwise |
| OR | \|\| | A \|\| B | Returns **TRUE** if either “A” or “B” are **TRUE**, or **FALSE** otherwise |
| EXCLUSIVE OR | ^ | A ^ B | Returns **TRUE** if either “A” is **TRUE** and “B” is **FALSE**, or vice-versa |
| NOT | ! | ! A | Returns **TRUE** if either “A” is **FALSE** or **FALSE** if “A” is **TRUE** |

```
(age > 10  && Name = "Bob")

(age == 10  || age == 11)
```