If Statements

- The = is used for assignment (set the value of a variable) and the == evaluates the expression (true or false). != checks if something is not equal to something else
- “if” statement evaluates conditions
- “else if” is to evaluate additional conditions; can use if but keeps it together and obvious; put in {} below else if
- “else” is for if not other conditions are satisfied do this block of code; put in {} below else
- All “if” statements must be followed curly braces {} to hold the code that goes along with it. You do not need the {} if it is followed by one single line of code

Code Samples

```
namespace Decisions
{
    class Program
    {
        static void Main(string[] args)
        {
        // Below is creating a string called userValue that will read what the user types
            Console.WriteLine("Please type something and press the Enter key");
            string userValue;
            userValue = Console.ReadLine();
            Console.WriteLine("You typed:" + userValue);
            Console.ReadLine();

        //The below is an “if” statement that will display a literal string based on
their response.  If the value is the variable string 1 then the next line of code will be
executed which is the display the message to the user.
            Console.WriteLine("Would you prefer what is behind door number 1, 2, 3?");
            string userValue = Console.ReadLine();
            if (userValue == "1")
            {
                Console.WriteLine("You won a new car!");
                Console.ReadLine();
            }
            else if (userValue == "2")
            {
                Console.WriteLine("You won a boat!");
                Console.ReadLine();
            }
            else if (userValue == "3")
            {
                Console.WriteLine("Sorry, try again next time.");
                Console.ReadLine();
            }
            else
            { Console.WriteLine("Sorry, we didn't understand");
            Console.ReadLine();
            }
        }
    }
}
```

ABOVE CAN BE SIMPLIFIED LIKE BELOW:
Created a new variable called message and set it to an empty string by doing “”. Then depending on what condition is true you change what the value of message is. Call the Console.ReadLine to pause the execution. **You can only omit the {} for if statement when it is followed by a SINGLE block of code.**

```
            Console.WriteLine("Would you prefer what is behind door number 1, 2, 3?");
            string userValue = Console.ReadLine();

            string message = "";

            if (userValue == "1")
                message = "You won a new car!";
            else if (userValue == "2")
                message = "You won a boat!";
            else if (userValue == "3")
                message = "Sorry, try again next time.";
            else
                message = "Sorry, we didn't understand.";

            Console.WriteLine(message);
            Console.ReadLine();

        }
    }
}
```

Could be simplified further by the below:
This is asking the question, then grabbing what the user inputs and evaluating it. Below allows us to use the string replacement syntax {0} (zero based number) that you can add to your literal string in Console.WriteLine to insert values into.

```
            Console.WriteLine("Would you prefer what is behind door number 1, 2, 3?");
            string userValue = Console.ReadLine();

            string message = (userValue == "1") ? "boat" : "strand of lint";
            Console.WriteLine("You won a {0}", message);

            Console.ReadLine();
        }
    }
}
```

All the code shown will do the same thing. Code that works without errors is considered correct and there are several ways to do the same thing. The more simplified the code is the more efficient it is but that does not mean that writing all your code out detailed like our first example is incorrect. Particularly when starting out it is often easier to write out more information to ensure we have everything we need.