Hack Club

Meeting 2

ST MARYS HACK CLUB

C#

Lets get it!

Task - C#

Let’s begin programming:

https://www.onlinegdb.com/online_csharp_compiler

Program.cs

```
using System;

namespace HelloWorld
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Hello World!");
    }
  }
}
```

Result:

```
Hello World!
```

C# - Variables and Data Types

Syntax

```
type variableName = value;
```

```
int myNum = 5;
double myDoubleNum = 5.99D;
char myLetter = 'D';
bool myBool = true;
string myText = "Hello";
```

C# - Data Types - Test

```
int myNum = 9;
double myDoubleNum = 8.99;
char myLetter = 'A';
bool myBoolean = false;
string myText = "Hello World";
```

C# - User Input

Similar to input() in Python, **Console.ReadLine()** is used to get user input.

An example of this is:

```
// Type your username and press enter
Console.WriteLine("Enter username:");

// Create a string variable and get user input from the keyboard and store it in the variable
string userName = Console.ReadLine();

// Print the value of the variable (userName), which will display the input value
Console.WriteLine("Username is: " + userName);
```

C# - Operators

Operators are used to perform operations on variables and values, such as arithmetic operators (math)

Operator	Name	Description	Example
+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	x++
--	Decrement	Decreases the value of a variable by 1	x--

In the example below, we use the + operator to add together two values:

```
int sum1 = 100 + 50;         // 150 (100 + 50)
int sum2 = sum1 + 250;       // 400 (150 + 250)
int sum3 = sum2 + sum2;      // 800 (400 + 400)
```

C# - Booleans

Very often, in programming, you will need a data type that can only have one of two values, like:

- YES / NO
- ON / OFF
- TRUE / FALSE

For this, C# has a bool data type, which can take the values true or false.

```
bool isCSharpFun = true;
bool isFishTasty = false;
Console.WriteLine(isCSharpFun);   // Outputs True
Console.WriteLine(isFishTasty);   // Outputs False
```

C# - If and Else statements

Syntax

```
if (condition1)
{
  // block of code to be executed if condition1 is True
}
else if (condition2)
{
  // block of code to be executed if the condition1 is false and condition2 is True
}
else
{
  // block of code to be executed if the condition1 is false and condition2 is False
}
```

Example

```
int time = 22;
if (time < 10)
{
  Console.WriteLine("Good morning.");
}
else if (time < 20)
{
  Console.WriteLine("Good day.");
}
else
{
  Console.WriteLine("Good evening.");
}
// Outputs "Good evening."
```

C# - While Loops

Loops can execute a block of code as long as a specified condition is reached.

Loops are handy because they save time, reduce errors, and they make code more readable

The while loop loops through a block of code as long as a specified condition is True:

```
int i = 0;
while (i < 5)
{
  Console.WriteLine(i);
  i++;
}
```

C# - Arrays

Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

```
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
Console.WriteLine(cars[0]);
// Outputs Volvo
```

See you next week!

ST MARYS HACK CLUB