**Introduction to the Labs**

1. When creating the labs make sure that the box to create a subdirectory is ticked. This will allow you to move your work to another machine if required.
2. Name your labs Lab1\_1 for lab 1 Project 1. This will prevent confusion later on.
3. Some labs will be challenging and frustrating. Persevere!
4. Enjoy

**Lab 1**

**Project 1 – Hello world**

Start a new Console project called Lab1\_1.

1. *Add the following code* between the braces after “static void Main(string[] args)”:

Console.WriteLine("Hello, world!");  
Console.ReadLine();

Run your application by pressing CTRL + F5

**Project 2 – String Variables**

1. Add the following

string firstName = "John";  
string lastName = "Doe";

//Print the following line on the screen  
Console.WriteLine("Name: " + firstName + " " + lastName);

Console.WriteLine("Please enter a new first name:");  
//Read the user input that was typed in

firstName = Console.ReadLine();

//Write the new name to the screen  
Console.WriteLine("New name: " + firstName + " " + lastName);

//Wait for the user to input something in order to exit the application  
Console.ReadLine();

Questions

1. What is the purpose of the // (in green)?

**Project 3 – Numeric Variables**

1. Add the following:

int number1, number2;

Console.WriteLine("Please enter a number:");

number1 = int.Parse(Console.ReadLine());

Console.WriteLine("Thank you. One more:");

number2 = int.Parse(Console.ReadLine());

Console.WriteLine("Adding the two numbers: " + (number1 + number2));

Console.ReadLine();

Questions

1. What do you think the following line does?

number1 = int.Parse(Console.ReadLine());

1. What happens if you leave off a ; from a line?
2. Change your program so that it can divide the two numbers. Run the program. Do you get the correct results? If not, why?
3. For 3 above, change the program so that the value displayed can show decimal places.

**Project 4 – Variables and Scope**

So far, we have only used local variables, which are variables defined and used within the same method. In C#, a variable defined inside a method can't be used outside of this method - that's why it's called local. If you're familiar with other programming languages, you may also know about global variables, which can be accessed from more places, but C# doesn't support the concept of global variables. Instead, you can define a field on a class, which can be accessed from all the methods of this class.

1. *Add the following code* after the “namespace Lab1\_4:

{

class Program

{

private static string helloClass = "Hello, class!";

static void Main(string[] args)

{

string helloLocal = "Hello, local!";

Console.WriteLine(helloLocal);

Console.WriteLine(Program.helloClass);

DoStuff();

}

static void DoStuff()

{

Console.WriteLine("A message from DoStuff: " + Program.helloClass);

Console.WriteLine("Press any key to exit");

Console.ReadLine();

}

}

}

Notice the **helloClass** member, declared on the class scope instead of inside a method - this will allow us to access it from both our Main() method as well as our own DoStuff() method. That is not true for our **helloLocal** variable, which has been declared inside the Main() method and can therefore only be used inside of this specific method.