CSE 1321L – C#

Introduction to C# and Visual Studio

Lab 1

**Goals:**

* Become familiar with Visual Studio Integrated Development Environment (IDE).
* Explore how Console C# applications work and understand the Main method.

For each lab program you develop, make sure to include the following header - **replace the dots with your section #, semester, your full name, your instructor’s name, and lab #:**

Class: CSE 1321L

Section: ...

Term: ...

Instructor: ...

Name: ...

Lab#: ...

#### *Make sure to put the correct comment character(s) before the above lines in each file. C# & Java use // for line comments. Use /\* \*/ for block comments (put comments between the characters).*

**Background:**

* Open Visual Studio.
* Click on File > New Project
* In the dialog box, name your file with your last name first. (e.g. Smith\_Sam\_Lab\_One
* As a review, in the Visual Studio IDE, look at the Solution Explorer. (View > Solution Explorer if it's not open). In it you will see the structure of the solution.
* Notice the structure of this solution corresponds 1-to-1 to the file system structure. The References folder includes information of which external libraries are required to support this application.

We can examine some of the files in our project (double click on the files to open them in the editor).

* The files with the .sln extension are called solution files. They describe the structures of project files.
* The files with the .csproj extension are called project files. They describe the structures of source code (.cs files) and the building method.
* A solution file is the starting point of a project; open this file when starting development of an application.
* Once the solution file is open, press CTRL + F5 to build and run.

There are some important parts of the default/blank Console C# code to note: Main is where the program begins. All code thereafter is called from this function.

* **Main()** - A program starts with Main(). Any application will always require one Main() function. If there are two or more, a build error will occur.
* **AssemblyInfo.cs** - This file contains auto-generated meta data about the application. You should leave this alone and not edit this file.
* References - This folder contains references to other code we might use; working with code someone else wrote is quite helpful

**Directions**

Create your own basic C# console application, as instructed below.

1. Open Visual Studio IDE and begin a new solution for a blank Console Application.
2. Name the console application: MyFirstProgram
3. Replace the Console.WriteLine(“Hello World!”); line with
   1. Console.WriteLine ("This is my first C# program!");
   2. Console.WriteLine ("This program does not do much!");
4. Save the file and Run it (F5).
5. If you get any syntax errors, correct them and make sure the code works.
6. Then complete the programs below using steps 1 through 5 replacing step 2 with the corresponding name and steps 3a and 3b with the code below:

================================== Program CountDown.cs ===================================

using System;

// Program CountDown.cs

// Demonstrate the difference between Write and WriteLine methods.   
  
public class CountDown   
{   
 // Prints two lines of output representing a rocket countdown.   
 public static void Main (String[] args)   
 {   
 Console.Write ("Three... ");   
 Console.Write ("Two... ");   
 Console.Write ("One... ");   
 Console.Write ("Zero... ");   
 Console.WriteLine ("Liftoff!"); // appears on first line   
 Console.WriteLine ("Houston, we have a problem.");   
 }   
}

================================== Program Addition.cs ====================================

using System;

// Program Addition.cs

// Demonstrate the difference between the addition and string concatenation operators.   
  
public class Addition   
{   
 // Concatenates and adds two numbers and prints the results.   
 public static void Main (String[] args)   
 {   
 Console.WriteLine ("24 and 45 concatenated: " + 24 + 45);  
 Console.WriteLine ("24 and 45 added: " + (24 + 45));   
 }   
}

==================================== Program Echo.cs ======================================

using System;

// Program Echo.cs

// Demonstrate reading a string from the user.   
  
public class Echo   
{   
 // Reads a character string from the user and prints it.   
 public static void Main (String[] args)   
 {   
 String message;   
 Console.Write("Enter a line of text:");   
 message = Console.ReadLine();   
 Console.WriteLine("You entered: \"" + message + "\"");   
 }   
}

=================================== Program GasMileage.cs =================================

using System;

// Program GasMileage.cs

public class GasMileage   
{   
 // Calculates fuel efficiency based on values entered by the user.   
 public static void Main (String[] args)   
 {   
 int miles;   
 double gallons, mpg;   
 Console.Write ("Enter the number of miles: ");   
 miles = Convert.ToInt32(Console.ReadLine());   
   
 Console.Write ("Enter the gallons of fuel used: ");   
 gallons = Convert.ToInt32(Console.ReadLine());   
   
 mpg = miles / gallons;   
 Console.WriteLine ("Miles Per Gallon: " + mpg);   
 }   
}

==================================== Program Facts.cs =====================================

using System;

// Program Facts.cs

// Demonstrate string concatenation operator conversion of an integer to a string.   
public class Facts   
{   
 // Prints various facts.   
 public static void Main (String[] args)   
 {   
 // Strings can be concatenated into one long string   
 Console.WriteLine ("We present the following facts for your "   
 + "extracurricular edification:");   
 Console.WriteLine ();   
   
 // A string can contain numeric digits   
 Console.WriteLine ("Letters in the Hawaiian alphabet: 12");   
   
 // A numeric value can be concatenated to a string   
 Console.WriteLine ("Dialing code for Antarctica: " + 672);   
 Console.WriteLine ("Year in which Leonardo da Vinci invented "   
 + "the parachute: " + 1515);   
 Console.WriteLine ("Speed of ketchup: " + 40 + " km per year");   
 }   
}

**Save all programs in one folder (call it Lab1) for future reference.**

**Instructions:**

1. All practice programs must be completed and working correctly.

2. The programs must be checked by the end of the designated lab session by Lab Instructor or TA.

3. Source code files (.cs) must be uploaded to D2L by due date