Assignment #3
Automated Porting of Code from C to C#

Background

In this assignment, you will create a tool that can take normal C code source files for console applications, and have these files transformed into working C# console applications.

You'll find that this type of solution is referred to as a "porting wizard", a tool that automates the portability process from one environment to another. Some port wizards transform source code from one compiler to another. Others from one GUI environment to another. Some from one language to another, as is the case here.

This is a fun exercise, in that this will be your first cut at trying to create a tool that you may find useful in the future. **You can work with a partner on this assignment if you wish**.

Your Task

Your job here is to create a tool that can filter a C source file for a complete console application, and transform it into an equivalent C#.NET console application, that can be compiled by Visual Studio.

It can be assumed for the purposes of this assignment that the C source code will use

- Only certain standard library support, such as printf() and gets() – to show support for console input/output
  - Note that in some cases, you are required to output variable values using printf() and using *format placeholders*
- And use FILE objects for support of file input/output (fopen(), fclose(), fgets(), fprintf())
- Please see the samples provided in this Assignment in D2L (assignment-3-Samples.zip) for more detail and for samples to use in your ***PortWizard*** testing

To run your filter, it will have a command line interface such as this:

```
portwizard -i C_source_file -o C#_output_file
```

Due Date : Feb 28, 2014 by 11:59pm
(in the D2L Dropbox)

For example:

```
                portwizard -i hello.c -o hello.cs
```

- The output file should appear similar to hello.cs file found in the assignment-3-Samples.zip file

** . ***Ensure that your output can be successfully compiled in Visual Studio*** **

Extra Notes and Food For Thought

- Remember that you're not trying to port every conceivable C programming situation
  - Instead, concentrate on a small subset of capabilities
  - Analyze the mechanisms of doing manual porting of software, in terms of how you as a human recognize that a printf() becomes a System.Console.WriteLine() and so forth
  - After you tabulate the mechanics behind performing a port, can you write a solution that automates that mechanical process …
- Now imagine the long term potential of a tool like this. Consider how many companies have invested in creating C or C++ applications, and now wish they could move to C#.NET
  - A tool like this – the ***PortWizard*** could handle that task. What a market you could own!
- Note that you might find the use of a **regular expression parsing library** quite useful in an assignment like this
- Note as well that this solution can be written in ***any development language you choose***
  - Since this tool is meant to facilitate portability, you will not be graded on the portability of the tool source code itself
  - However, you must ensure maintainability and quality of your solution (this means good naming scheme, commenting, etc.)
- When it comes to marking this assignment, I will simply run your portWizard against the source code I gave you originally as well as some blind test files (constructed from these sample files)