Lab 3 – Collections and LINQ

# Purpose

* Create a console applications to perform the following tasks:
  1. Create a directory crawler
  2. Perform a number queries against the data gathered
* Familiarize yourself with .NET Collections and basic LINQ queries

# Due Date

* This lab is due before the beginning of the Lab period (3:59:59pm) on February 1

# Assessment

* This Lab is worth 2% of your total course mark.

# Estimated Time

* This Lab is estimated to take 2 hours.

This is only an estimate of the time required to complete this Lab. I would encourage you to work at your own pace and if at all possible obtain a laptop so that you can work on your assignments from anywhere

# Assigned Readings

* None this week

# Lab Supplies

To complete this lab you will require the following lab supplies:

* Visual Studio 2010
  + Available on MSDNAA
* Lecture notes
* Laptop or Caddy

# Summary of Tasks

1. Create an application to preform LINQ queries against a custom collection
2. Demo

READ EVERY WORD BEFORE YOU START, your life will be so much easier!

# Task 1

Create a C# .NET console application that will:

1. Crawl through the ‘Windows’ directory of your computer and store the FileInfo objects in a collection.
   * Feel free to create your own collection, or use a built-in .NET collection. If you are going to use a built in collection I recommend System.Collections.Generic.List<>. It contains most of the common built in functions used in the majority of common programming tasks.
2. The application must use LINQ to query your collection and display the following results. Please use a menu to allow the user to run the queries individually.
   * Count the number of files in your collection and display the result.
   * Find the largest file in your collection and display its Name, Path and Size.
   * Find all the files in your collection whose name is larger than 10 characters.
   * Order the files by Size, Name, Path and then display each files Size, Name and Path
   * File all the DLLs stored in your windows directory and use Distinct to display only the unique file names
   * Select all the Length properties from your collection and display them in ascending and descending order
     1. Do the same query but show the Max of the Length properties
     2. Do the same query but show the Min of the Length properties
     3. Do the same query but show the Average of the Length properties
     4. Do the same query but sum the total of the Length properties
   * Use First() to select the first object in your collection

Useful Namespace(s):

* System;
* System.IO;
* System.Collections;
* System.Collections.Generic;

Useful Classes(s):

* string
* Directory
  + Static method Directory.GetDirectories()
  + Static method Directory.GetFiles()
* FileInfo
  + Contains information about a file using its path
  + FileInfo.Length
  + FileInfo.Name
* DirectoryInfo
  + Contains information about a directory using its path
  + DirectoryInfo.Name
* List<>, Collection<>

NOTE: watch out for file system access issues while scanning directories and files:

# Task 2

Demo to your lab professor.