Activity 1

NeDL Transition Academy

Programming Activity 1: Console application to parse contacts file (CSV)

Part 1

Step 1. Create a new console application in Visual Studio.

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

Step 2. Add using statement.

```
using System.IO;
```

Step 3. Open the text file Contacts.csv and read all the lines

```
var lines = File.ReadAllLines("path/to/Contacts.csv");
```

Step 4: Loop through all lines of the file.

```
            foreach (var line in lines)
            {
                var splits = line.Split(',');
                Console.WriteLine(splits[1] + " " + splits[2]);
            }
```

Step 5: Parse the line using the string split method.

```
var splits = line.Split(',');
```

Step 6: Output each name to the console.

```
Console.WriteLine(splits[1] + " " + splits[2]);
```

Solution

```
    class Program
    {
        static void Main(string[] args)
        {
            var lines = File.ReadAllLines("path/to/Contacts.csv");

            foreach (var line in lines)
            {
                var splits = line.Split(',');
                Console.WriteLine(splits[1] + " " + splits[2]);
            }
        }
    }
```

Stretch your thinking

What would happen if line.Split returns no splits?

How could we protect ourselves from line.Split returning no splits? How could we prevent that exception?

Part 2

Change the solution to use a *StreamReader* to loop through the file one line at a time. Use File.OpenRead.

Hints

- File.OpenRead("path/to/file/Contacts.csv");
- new StreamReader(stream)
- ReadLine() method to read a line of text

Stretch your thinking

Compare the solutions from part 1 to part 2. What are the advantages of part 1, what are the advantages of part 2?