C# Application Lab

Lab 1: Add Products.tsv File to Project

Get the labs/samples for this course.

Locate the **Products.tsv** file and add it to your project.

Lab 2: CheckFileSize() Method

Open the **Product.cs** file and add the following method to this class.

```
private void CheckFileSize(string fileName) {
  FileStream fs = null;

  try {
    fs = File.Open(fileName, FileMode.Open);
    if (fs.Length > 100000) {
      throw new FileTooLargeException($"The file '{fileName}' is too
large.", fs.Length);
    }
  }
  finally {
    fs?.Close();
  }
}
```

Lab 3: ReadProductFile() Method

Open the **Product.cs** file and add the following method to this class.

```
private string[] ReadProductFile(string fileName) {
  string[] ret;

  if (File.Exists(fileName)) {
    // Check to ensure file is not too large to open
    CheckFileSize(fileName);
    // Attempt to read all lines in the file
    ret = File.ReadAllLines(fileName);
  }
  else {
    throw new FileNotFoundException($"The file '{fileName}' does not 
exist.");
  }

  return ret;
}
```

Lab 4: ProcessProductLines() Method

Open the **Product.cs** file and add the following method to this class.

```
private Product[] ProcessProductLines(string[] lines) {
  Product[] ret;
  ArrayList products = new();

  foreach (string item in lines) {
    string[] entity = item.Split('\t');

    products.Add(new Product {
      ProductId = Convert.ToInt32(entity[0]),
      Name = entity[1],
      StandardCost = Convert.ToDecimal(entity[2]),
      ListPrice = Convert.ToDecimal(entity[3]),
      SellStartDate = Convert.ToDateTime(entity[4]),
      SellEndDate = string.IsNullOrEmpty(entity[5]) ? 
DateTime.MinValue : Convert.ToDateTime(entity[5])
    });
  }

  // Convert ArrayList to Array
  ret = (Product[])products.ToArray(typeof(Product));

  return ret;
}
```

Lab 5: GetProducts() Method

Open the **Product.cs** file and add the following method to this class.

```
public Product[] GetProducts(string fileName) {
  Product[] ret = null;
  string[] lines;

  try {
    lines = ReadProductFile(fileName);
    ret = ProcessProductLines(lines);
  }
  catch (FileNotFoundException ex) {
    Console.WriteLine(ex.ToString());
  }
  catch (FileTooLargeException ex) {
    Console.WriteLine(ex.ToString());
  }
  catch (Exception ex) {
    Console.WriteLine(ex.ToString());
  }

  return ret;
}
```

Lab 6: ToString() Method

Open the **Product.cs** file and override the ToString() method.

```
public override string ToString() {
  StringBuilder sb = new(1024);

  sb.AppendLine($"{Name} - ({ProductId})");
  sb.Append("Cost = " + StandardCost.ToString("c"));
  sb.Append("  Price = " + ListPrice.ToString("c"));
  sb.AppendLine("  Profit = " + Profit.ToString("c"));
  sb.Append("Start Selling = " + SellStartDate.ToString("d"));
  if (SellEndDate == DateTime.MinValue) {
    sb.AppendLine("  No Selling End Date");
  }
  else {
    sb.Append("  End Selling on = '" + SellEndDate.ToString("d") +
"'");
    sb.AppendLine("  Selling Days = " + GetNumberOfSellDays());
  }

  return sb.ToString();
}
```

Lab 7: Modify the Program.cs File

Replace the entire contents of the **Product.cs** file to the following code.

```
using CSharpSamples;
string fileName = @"D:\Samples\Products.tsv";

Product entity = new();
Product[] products = entity.GetProducts(fileName);

foreach (Product item in products) {
  Console.WriteLine(item);
}
```

Try It Out

Run the application and you view the output.