

# Code Repository Exercises

## C#

**Directory Name:** C-Sharp

**Required Pages/Components:**

1. Download the accompanying zip file, extract it, and place the files in the following locations:
  - a. *Product.cs*: Create an App\_Code folder inside the root directory of the repository (**not** inside the C-Sharp directory created for this exercise). The App\_Code directory can be created by right-clicking on the root directory in the Solution Explorer Window, selecting 'Add ASP.NET Folder..' and then selecting 'App\_Code'. Place the *Product.cs* file in the App\_Code directory.
  - b. *microwave.jpg*, *mixer.jpg*, *toaster.jpg*: Place these files in the C-Sharp directory created for this exercise
2. *TaxableProduct.cs*: A new C# class placed in the App\_Code directory. This class inherits from the *Product* class downloaded in the previous step. Add the following to this class (the first three elements are shown on pages 69-70 of the textbook):
  - a. *taxRate*: a private member variable of type decimal. Initialized to 1.15M
  - b. *TotalPrice*: a public property of type decimal. *Get* returns the product of the Price and the TaxRate
  - c. A constructor that accepts the same parameters as the *Product* class constructor. This constructor should invoke the *Product* class constructor using the *base* keyword.
  - d. *public override string GetHtml()*: a public method that overrides the *GetHtml()* method of the *Product* class. This method should look like the *Product* version of *GetHtml* method except that it should output the **public properties**, *Name*, *TotalPrice*, and *ImageUrl*.
3. *ProductCatalog.cs*: A new C# class placed in the App\_Code directory. Add the following components to this class
  - a. *products*: a private generic list (not an ArrayList) of *TaxableProduct* objects (see the bottom of page 75 for an example).
  - b. *public void AddProduct(string name, decimal price, string imageUrl)*: a public method, that creates a new *TaxableProduct* object and adds it to the *products* list.
  - c. *public Product GetHighPricedProduct()*: a public method that finds the product in the *products* list that has the highest price. Returns the product as a *Product* object.
  - d. *public string GetCatalogHtml()*: a public method that invokes the *GetHtml()* method of each product in the *products* list, stores the result in a string variable, and returns this string variable.
4. *ProductCatalogTest.aspx*: A new ASP.NET Web Form placed in the C-Sharp directory. In the *Page\_Load* method of this page, do the following:
  - a. Create a new instance of the *ProductCatalog* class

- b. Add three *TaxableProduct* objects to the *products* list of the new product catalog by using the *AddProduct* method. Use the three images you downloaded for the product images.
- c. Invoke the *GetCatalogHtml* method of the product catalog within the *Response.Write* method. This will print the catalog products on the page with the necessary html code
- d. Invoke the *GetHighPricedProduct* method of the product catalog and store the result in a *TaxableProduct* object (a cast will be necessary). Use *Response.Write* to print an html paragraph that identifies the *Name* and *TotalPrice* of the product.

**Header on Default.aspx: C#**

**Pages linked from Default.aspx: ProductCatalogTest.aspx**