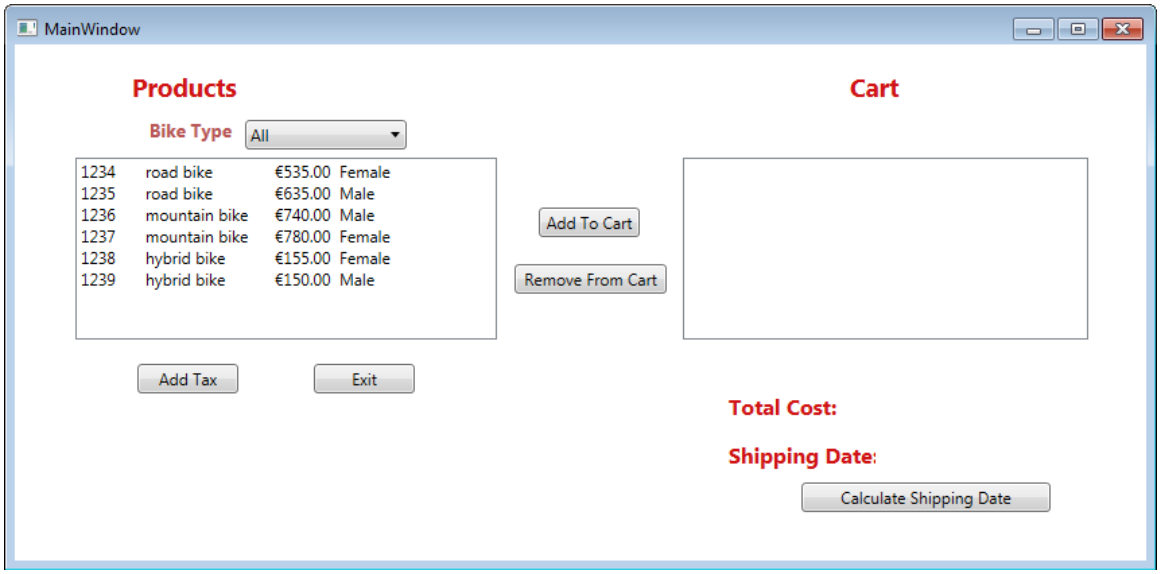
Week 10 – Exercise 1 - Shopping Cart Continued

In this exercise, you'll continue working on the Shopping Cart WPF application that was worked on in the Lecture.



Part 1

1. Update the XAML interface shown above to include the ComboBox and Selection Changed method for the filter

```
Name="comboBoxBikeType"
.
.
.
SelectionChanged="comboBoxBikeType_SelectionChanged"/>
```

2. Create a Variations class class with one auto-implemented methods Variation Type
3. In the MainWindow.xaml.cs file create 2 Variations Male and Female.
4. Update the Products class to include an auto implemented Variations property
5. Update the products to include 6 Products with a Variation now added to each
6. Store the 6 Products in a Product Array.
7. Add a third Array to hold filtered Products
8. Add 3 options to the Filter ComboBox in the OnWindowLoad()

```
string[] bikeTypes = { "All", "Male", "Female" };
comboBoxBikeType.ItemsSource = bikeTypes;
comboBoxBikeType.SelectedIndex = 0; //Set index to All
```

9. Add the comboBoxBikeType_SelectionChanged() method.
10. In the comboBoxBikeType_SelectionChanged() get the item selected in the Filter then go through the Variation of each product using a loop and if the Variation matches the Filter selected option then add this Product to the FilteredProductsArray. Display the FilteredProducts Array on the Filtered Products List Box

Part 2

11. For the Total Coast add a label with the content “Total Cost” and beside it add a blank label with the name `labelCalculatedCost`
12. Add a method `calculateTotalCost()` that loops through all products in Cart Array and gets the price and the tax for each product and adds them to a total.
13. Using the empty label `labelCalculatedCost` write the total onto this label using:
    `labelCalculatedCost.Content = total.ToString("C", CultureInfo.CurrentCulture);`
14. This method `calculateTotalCost()` is called from the end of the `btnAddToCart_Click()` method