Appendix: Code Samples (used if not running sequentially through the lab)

This section contains the starting blocks of code required if skipping sections. All code written in this lab is contained in one file: *program.cs* Each exercise in this lab is structured to build upon the code written in the previous exercises. Compiled in this appendix are the complete sets of code that are generated by each exercise (the entire *program.cs* file). For example: If you wish to start at exercise 4, then you may start with the code provided for you in the exercise 4 below.

Exercise 1:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace NewLanguageFeatures
{
    public class Customer
    {
        public int CustomerID { get; private set; }
        public string Name { get; set; }
        public string City { get; set; }
        public Customer(int ID)
        {
            CustomerID = ID;
        }
        public override string ToString()
        {
            return Name + "\t" + City + "\t" + CustomerID;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            List<Customer> customers = CreateCustomers();
            Console.WriteLine("Customers:\n");
            foreach (Customer c in customers)
                Console.WriteLine(c);
        }
        static List<Customer> CreateCustomers()
        {
           return new List<Customer>
           {
              new Customer(1) { Name = "Maria Anders",     City = "Berlin"    },
              new Customer(2) { Name = "Laurence Lebihan", City = "Marseille" },
```

```
                    new Customer(3) { Name = "Elizabeth Brown",  City = "London"    },
                    new Customer(4) { Name = "Ann Devon",        City = "London"    },
                    new Customer(5) { Name = "Paolo Accorti",    City = "Torino"    },
                    new Customer(6) { Name = "Fran Wilson",     City = "Portland"  },
                    new Customer(7) { Name = "Simon Crowther",  City = "London"    },
                    new Customer(8) { Name = "Liz Nixon",       City = "Portland"  }
                };
            }
        }
    }
```

Exercise 2:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace NewLanguageFeatures
{
    public class Customer
    {
        public int CustomerId { get; private set; }
        public string Name { get; set; }
        public string City { get; set; }
        public Customer(int Id)
        {
            CustomerId = Id;
        }
        public override string ToString()
        {
            return Name + "\t" + City + "\t" + CustomerId;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Customer c = new Customer(1);
            c.Name = "Alex Roland";
            c.City = "Berlin";
            Console.WriteLine(c);
        }
    }
}
```

Exercise 3:

```
    using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace NewLanguageFeatures
{
    public class Customer
    {
        public int CustomerID { get; private set; }
        public string Name { get; set; }
        public string City { get; set; }
        public Customer(int ID)
        {
            CustomerID = ID;
        }
        public override string ToString()
        {
            return Name + "\t" + City + "\t" + CustomerID;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            List<Customer> customers = CreateCustomers();
            Console.WriteLine("Customers:\n");
            foreach (Customer c in customers)
                Console.WriteLine(c);
        }
        static List<Customer> CreateCustomers()
        {
           return new List<Customer>
           {
              new Customer(1) { Name = "Maria Anders",      City = "Berlin"    },
              new Customer(2) { Name = "Laurence Lebihan", City = "Marseille" },
              new Customer(3) { Name = "Elizabeth Brown",  City = "London"    },
              new Customer(4) { Name = "Ann Devon",     City = "London"     },
              new Customer(5) { Name = "Paolo Accorti", City = "Torino"     },
              new Customer(6) { Name = "Fran Wilson",    City = "Portland"  },
              new Customer(7) { Name = "Simon Crowther",City = "London"     },
              new Customer(8) { Name = "Liz Nixon",     City = "Portland"  }
           };
       }
    }
}
```

Exercise 4:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace NewLanguageFeatures
{
    public class Customer
    {
        public int CustomerID { get; private set; }
        public string Name { get; set; }
        public string City { get; set; }
        public Customer(int ID)
        {
            CustomerID = ID;
        }
        public override string ToString()
        {
            return Name + "\t" + City + "\t" + CustomerID;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Customer c = new Customer(1);
            c.Name = "Maria Anders";
            c.City = "Berlin";
            Console.WriteLine(c);
        }
    }
}
```

Exercise 5:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Linq.Expressions;
namespace NewLanguageFeatures
{
    public static class Extensions
    {
        public static List<T> Append<T>(this List<T> a, List<T> b)
        {
```

```
            var newList = new List<T>(a);
            newList.AddRange(b);
            return newList;
        }
        public static bool Compare(this Customer customer1, Customer customer2)
        {
            if (customer1.CustomerID == customer2.CustomerID &&
                customer1.Name == customer2.Name &&
                customer1.City == customer2.City)
            {
                return true;
            }
            return false;
        }
    }
    public class Store
    {
        public string Name { get; set; }
        public string City { get; set; }
        public override string ToString()
        {
            return Name + "\t" + City;
        }
    }
    public class Customer
    {
        public int CustomerID { get; private set; }
        public string Name { get; set; }
        public string City { get; set; }
        public Customer(int ID)
        {
            CustomerID = ID;
        }
        public override string ToString()
        {
            return Name + "\t" + City + "\t" + CustomerID;
        }
    }
    class Program
    {
        static void Query()
        {
             var stores = CreateStores();
             var numLondon = stores.Count(s => s.City == "London");
             Console.WriteLine("There are {0} stores in London. ", numLondon);
        }
        static void Main(string[] args)
```

```
        {
            Query();
        }
        public static List<Customer> FindCustomersByCity(
            List<Customer> customers,
            string city)
        {
            return customers.FindAll(c => c.City == city);
        }
        static List<Store> CreateStores()
        {
            return new List<Store>
            {
                new Store { Name = "Jim's Hardware",   City = "Berlin"    },
                new Store { Name = "John's Books",     City = "London"    },
                new Store { Name = "Lisa's Flowers",   City = "Torino"    },
                new Store { Name = "Dana's Hardware",  City = "London"    },
                new Store { Name = "Tim's Pets",       City = "Portland"  },
                new Store { Name = "Scott's Books",    City = "London"    },
                new Store { Name = "Paula's Cafe",     City = "Marseille" }
            };
        }
        static List<Customer> CreateCustomers()
        {
            return new List<Customer>
            {
                new Customer(1) { Name = "Maria Anders",      City = "Berlin"    },
                new Customer(2) { Name = "Laurence Lebihan",  City = "Marseille" },
                new Customer(3) { Name = "Elizabeth Brown",   City = "London"    },
                new Customer(4) { Name = "Ann Devon",         City = "London"    },
                new Customer(5) { Name = "Paolo Accorti",     City = "Torino"    },
                new Customer(6) { Name = "Fran Wilson",       City = "Portland"  },
                new Customer(7) { Name = "Simon Crowther",    City = "London"    },
                new Customer(8) { Name = "Liz Nixon",         City = "Portland"  }
            };
        }
    }
}
```