**Part 76 - Working with generic list class and ranges in c#**

Please watch [**Part 74**](http://csharp-video-tutorials.blogspot.com/2013/08/part-75-list-collection-class-in-c.html) & [**75**](http://csharp-video-tutorials.blogspot.com/2013/08/part-74-list-collection-class-in-c.html) before proceeding with this video. In thie video, we will discuss  
  
**1. AddRange() -**Add() method allows you to add one item at a time to the end of the list, where as AddRange() allows you to add another list of items, to the end of the list.

**2. GetRange() -**Using an item index, we can retrieve only one item at a time from the list, if you want to get a list of items from the list, then use GetRange() function. This function expects 2 parameters, i.e the start index in the list and the number of elements to return.  
  
**3. InsertRange() -**Insert() method allows you to insert a single item into the list at a specificed index, where as InsertRange() allows you, to insert another list of items to your list at the specified index.  
  
**4. RemoveRange() - Remove**() function removes only the first matching item from the list. **RemoveAt**() function, removes the item at the specified index in the list. **RemoveAll**() function removes all the items that matches the specified condition. **RemoveRange**() method removes a range of elements from the list. This function expects 2 parameters, i.e the start index in the list and the number of elements to remove. If you want to remove all the elements from the list without specifying any condition, then use **Clear**() function.

public class Program  
{  
    public static void Main()  
    {  
        // Create Customer Objects  
        Customer customer1 = new Customer()  
        {  
            ID = 101,  
            Name = "Mark",  
            Salary = 4000,  
            Type = "RetailCustomer"  
        };  
  
        Customer customer2 = new Customer()  
        {  
            ID = 102,  
            Name = "Pam",  
            Salary = 7000,  
            Type = "RetailCustomer"  
        };  
  
        Customer customer3 = new Customer()  
        {  
            ID = 103,  
            Name = "Rob",  
            Salary = 5500,  
            Type = "RetailCustomer"  
        };  
  
        Customer customer4 = new Customer()  
        {  
            ID = 104,  
            Name = "John",  
            Salary = 6500,  
            Type = "CorporateCustomer"  
        };  
  
        Customer customer5 = new Customer()  
        {  
            ID = 105,  
            Name = "Sam",  
            Salary = 3500,  
            Type = "CorporateCustomer"  
        };  
  
          
        List<Customer> listCustomers = new List<Customer>();  
        // Add() method allows you to add one at a time to the end of the list  
        listCustomers.Add(customer1);  
        listCustomers.Add(customer2);  
        listCustomers.Add(customer3);  
  
        List<Customer> listCorporateCustomers = new List<Customer>();  
        listCorporateCustomers.Add(customer4);  
        listCorporateCustomers.Add(customer5);  
  
        // AddRange() allows you to add another list of items, to the end of the list  
        listCustomers.AddRange(listCorporateCustomers);  
  
        foreach (Customer customer in listCustomers)  
        {  
            Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}, Type = {3}",  
                customer.ID, customer.Name, customer.Salary, customer.Type);  
        }  
        Console.WriteLine("------------------------------------------------------");  
  
        // GetRange() function returns a list of items from the list.

  List<Customer> corporateCustomers = listCustomers.GetRange(3, 2);  
        foreach (Customer customer in corporateCustomers)  
        {  
            Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}, Type = {3}",  
                customer.ID, customer.Name, customer.Salary, customer.Type);  
        }  
        Console.WriteLine("------------------------------------------------------");  
  
        // Remove() function removes only the first matching item from the list.  
        listCustomers.Remove(customer1);

  // RemoveAt() function, removes the item at the specified index in the list.  
        listCustomers.RemoveAt(0);  
  
        // RemoveAll() function removes all the items that matches the specified condition.  
        listCustomers.RemoveAll(x => x.Type == "RetailCustomer");  
  
        foreach (Customer customer in listCustomers)  
        {  
            Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}, Type = {3}",  
                customer.ID, customer.Name, customer.Salary, customer.Type);  
        }  
        Console.WriteLine("------------------------------------------------------");  
  
        // RemoveRange() method removes a range of elements from the list.

        // This function expects 2 parameters, i.e the start index in the   
        // list and the number of elements to remove.  
        listCustomers.RemoveRange(0, 2);  
  
        // Insert() method allows you to insert a single item at a time into   
        // the list at a specificed index  
        listCustomers.Insert(0, customer1);  
        listCustomers.Insert(1, customer2);  
        listCustomers.Insert(2, customer3);  
  
        // InsertRange() allows you, to insert another list of items to your list at the specified index  
        listCustomers.InsertRange(0, listCorporateCustomers);  
  
        foreach (Customer customer in listCustomers)  
        {  
            Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}, Type = {3}",  
                customer.ID, customer.Name, customer.Salary, customer.Type);  
        }  
        Console.WriteLine("------------------------------------------------------");  
  
        // If you want to remove all the elements from the list without specifying   
        // any condition, then use Clear() function.  
        listCustomers.Clear();  
  
        Console.WriteLine(" Total Items in the List = " + listCustomers.Count);  
    }  
}  
  
public class Customer  
{  
    public int ID { get; set; }  
    public string Name { get; set; }  
    public int Salary { get; set; }  
    public string Type { get; set; }  
}