**Part 75 - List collection class in c# continued**

This is continuation to [Part 74](http://csharp-video-tutorials.blogspot.com/2013/08/part-74-list-collection-class-in-c.html). So, please watch [Part 74](http://csharp-video-tutorials.blogspot.com/2013/08/part-74-list-collection-class-in-c.html), before proceeding. In this video, we will discuss  
**1. Contains() function -** Use this function to check if an item exists in the list. This method returns true if the items exists, else false.

**2. Exists() function -**Use this function, to check if an item exists in the list based on a condition. This method returns true if the items exists, else false.  
  
**3. Find() function -**This method searches for an element that matches the conditions defined by the specified lambda expression and returns the first matching item from the list.  
  
**4. FindLast() function -**This method searches for an element that matches the conditions defined by the specified lambda expression and returns the Last matching item from the list.  
  
**5. FindAll() function -**This method returns all the items from the list that match the conditions specified by the lambda expression.

**6. FindIndex() function -**This method returns the index of the first item, that matches the condition specified by the lambda expression. There are 2 other overloads of this method which allows us to specify the range of elements to search, with in the list.  
  
**7. FindLastIndex() function -**This method returns the index of the last item, that matches the condition specified by the lambda expression. There are 2 other overloads of this method which allows us to specify the range of elements to search, with in the list.

**8. Convert an array to a List -**Use ToList() method  
  
**9. Convert a list to an array -**Use ToArray() method  
  
**10. Convert a List to a Dictionary -**Use ToDictionary() method

public class Program  
{  
    public static void Main()  
    {  
        // Create Customer Objects  
        Customer customer1 = new Customer()  
        {  
            ID = 101,  
            Name = "Mark",  
            Salary = 4000  
        };  
  
        Customer customer2 = new Customer()  
        {  
            ID = 102,  
            Name = "Pam",  
            Salary = 7000  
        };  
  
        Customer customer3 = new Customer()  
        {  
            ID = 104,  
            Name = "Rob",  
            Salary = 5500  
        };  
  
        Customer[] arrayCustomers = new Customer[3];  
        arrayCustomers[0] = customer1;  
        arrayCustomers[1] = customer2;  
        arrayCustomers[2] = customer3;  
  
        // To convert an array to a List, use ToList() method  
        List<Customer> listCustomers = arrayCustomers.ToList();  
        foreach (Customer c in listCustomers)  
        {  
            Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}", c.ID, c.Name, c.Salary);  
        }  
        Console.WriteLine("------------------------------------------------------");  
  
        // To convert a List to an array, use ToLArray() method

Customer[] arrayAllCustomers = listCustomers.ToArray();  
        foreach (Customer c in arrayAllCustomers)  
        {  
            Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}", c.ID, c.Name, c.Salary);  
        }  
        Console.WriteLine("------------------------------------------------------");  
  
        // To convert a List to a Dictionary use ToDictionary() method  
        Dictionary<int, Customer> dictionaryCustomers = listCustomers.ToDictionary(x => x.ID);  
        foreach (KeyValuePair<int, Customer> keyValuePairCustomers in dictionaryCustomers)  
        {  
            Console.WriteLine("Key = {0}", keyValuePairCustomers.Key);  
            Customer c = keyValuePairCustomers.Value;  
            Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}", c.ID, c.Name, c.Salary);  
        }  
        Console.WriteLine("------------------------------------------------------");  
  
        // To check if an item exists in the list use Contains() function  
        // This method returns true if the items exists, else false  
        if (listCustomers.Contains(customer2))  
        {  
            Console.WriteLine("Customer2 object exists in the list");  
        }  
        else  
        {  
            Console.WriteLine("Customer2 object does not exist in the list");  
        }  
        Console.WriteLine("------------------------------------------------------");  
  
        // To check if an item exists in the list based on a condition, then use Exists() function

        // This method returns true if the items exists, else false  
        if (listCustomers.Exists(x => x.Name.StartsWith("M")))  
        {  
            Console.WriteLine("List contains customer whose name starts with M");  
        }  
        else  
        {  
            Console.WriteLine("List does not contain a customer whose name starts with M");  
        }  
        Console.WriteLine("------------------------------------------------------");  
  
        // Find() method searches for an element that matches the conditions defined by   
        // the specified lambda expression and returns the first matching item from the list  
        Customer cust = listCustomers.Find(customer => customer.Salary > 5000);  
        Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}", cust.ID, cust.Name, cust.Salary);

Console.WriteLine("------------------------------------------------------");  
  
        // FindLast() method searches for an element that matches the conditions defined  
        // by the specified lambda expression and returns the Last matching item from the list  
        Customer lastMatch = listCustomers.FindLast(customer => customer.Salary > 5000);  
        Console WriteLine("ID = {0}, Name = {1}, Salary = {2}", lastMatch.ID, lastMatch.Name, lastMatch.Salary);  
        Console.WriteLine("------------------------------------------------------");  
  
        // FindAll() method returns all the items from the list that  
        // match the conditions specified by the lambda expression  
        List<Customer> filteredCustomers = listCustomers.FindAll(customer => customer.Salary > 5000);  
        foreach (Customer cstmr in filteredCustomers)  
        {  
            Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}", cstmr.ID, cstmr.Name, cstmr.Salary);  
        }  
        Console.WriteLine("------------------------------------------------------");  
  
        // FindIndex() method returns the index of the first item, that matches the

// condition specified by the lambda expression. There are 2 other overloads  
        // of this method which allows us to specify the range of elements to   
        // search, with in the list.  
        Console.WriteLine("Index of the first matching customer object whose salary is greater 5000 =" +  
            listCustomers.FindIndex(customer => customer.Salary > 5000));  
        Console.WriteLine("------------------------------------------------------");  
  
        // FindLastIndex() method returns the index of the last item,   
        // that matches the condition specified by the lambda expression.   
        // There are 2 other overloads of this method which allows us to specify   
        // the range of elements to search, with in the list.  
        Console.WriteLine("Index of the Last matching customer object whose salary is greater 5000 = " +  
            listCustomers.FindLastIndex(customer => customer.Salary > 5000));  
        Console.WriteLine("------------------------------------------------------");  
    }  
}  
  
public class Customer  
{  
    public int ID { get; set; }  
    public string Name { get; set; }  
    public int Salary { get; set; }  
}