**1050 Programming Logic**

Lab 09 – Collections and LINQ

Name: \_\_Russell Bauza

***Instructions:*** *Complete the following exercises. Push your code to github and share the URL to your repository by submitting it to Blackboard.*

1. Given the following code, output all the elements using a foreach loop. (2 points)

var fruits = new[] { "apple", "mango", "orange", "apricot", "cherry", ”grape”,” blueberry” };

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_9.\_1.\_2

{

class Program

{

static void Main()

{

string[] array = { "apple", "mango", "orange", "apricot", "cherry", "grape", "blueberry" };

Console.WriteLine("\nList of fruit: ");

foreach (var value in array)

{

Console.Write($" {value}");

}

}

}

}

1. Write a LINQ query to convert all fruit names in the above array to uppercase select fruit names that start with an “A”. Use a foreach loop to display the query results.(5 points)

using System;

using System.Linq;

using System.Collections.Generic;

class LINQWithListCollection

{

static void Main()

{

var items = new List<string>();

items.Add("apple");

items.Add("mango");

items.Add("orange");

items.Add("apricot");

items.Add("grape");

items.Add("cherry");

items.Add("blueberry");

Console.Write("item contains:");

foreach (var item in items)

{

Console.Write($" {item}");

}

Console.WriteLine();

var startsWithA =

from item in items

let uppercaseString = item.ToUpper()

where uppercaseString.StartsWith("A")

orderby uppercaseString

select uppercaseString;

Console.Write("results of query startsWithA:");

foreach (var item in startsWithA)

{

Console.Write($" {item}");

}

Console.WriteLine();

items.Add("avocado");

items.Add("artichoke");

Console.Write("Items contains:");

foreach (var item in items)

{

Console.Write($" {item}");

}

Console.WriteLine();

Console.Write("results of query startsWithA");

foreach (var item in startsWithA)

{

Console.Write($" {item}");

}

Console.WriteLine();

}

}

1. Create a 6-element List<T> collection to store the names of last six months of the year. Display the Count and Capacity of the creation. Use a for loop to display the last six months of the year. Insert the first six months of the year into this List in the right sequence. Use a for loop to display the twelve months of the year in the right sequence from the List. (4 points)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab9.\_3

{

class Program

{

static void Main(string[] args)

{

var items = new List<string>();

Console.WriteLine("Before adding to items: " + $"Count = {items.Count}; Capacity = {items.Capacity}");

items.Add("August");

items.Add("September");

items.Add("October");

items.Add("November");

items.Add("December");

items.Insert(0, "July");

Console.WriteLine("After adding six elements to items" + $"Count = {items.Count}; Capacity = {items.Capacity}");

Console.Write("\nDisplay list contents with counter-controlled loop:");

for (var i = 0; i < items.Count; i++)

{

Console.Write($" {items[i]}");

}

Console.Write("\nDisplay list contents with forech statement:");

foreach (var item in items)

{

Console.Write($" {item}");

}

}

}

}

1. Remove the last six months from the above List. Display the Count and Capacity of the List. Write a LINQ query to select all the months that end with the letters “ber”. Display the query results with a foreach loop.(4 points)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab9.\_4

{

class Program

{

static void Main(string[] args)

{

var items = new List<string>();

items.Add("January");

items.Add("February");

items.Add("March");

items.Add("April");

items.Add("May");

items.Add("June");

items.Add("July");

items.Add("August");

items.Add("September");

items.Add("October");

Console.Write("Items contain:");

foreach (var item in items)

{

Console.Write($" {item}");

}

Console.WriteLine();

var endsWithBer=

from item in items

let uppercaseString = item.ToUpper()

where uppercaseString.EndsWith("ber")

orderby uppercaseString

select uppercaseString;

Console.Write("results of query endsendswithber:");

foreach (var item in endsWithBer)

{

Console.Write($" {item}");

}

Console.WriteLine();

items.Add("November");

items.Add("December");

Console.Write("items contains:");

foreach (var item in items)

{

Console.Write($" {item}");

}

Console.WriteLine();

Console.Write("results of query endsWithBer");

foreach (var item in endsWithBer)

{

Console.Write($" {item}");

}

Console.WriteLine();

}

}

}

Or another option is:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab9.\_4.\_1

{

class Program

{

static void Main(string[] args)

{

var items = new List<string>();

Console.WriteLine("Before adding to items: " + $"Count = {items.Count}; Capacity = {items.Capacity}");

items.Add("January");

items.Add("February");

items.Add("March");

items.Add("April");

items.Add("May");

items.Add("June");

items.Add("August");

items.Add("September");

items.Add("October");

items.Add("November");

items.Add("December");

items.Insert(0, "July");

Console.WriteLine("After adding six elements to items" + $"Count = {items.Count}; Capacity = {items.Capacity}");

Console.Write("\nDisplay list contents with counter-controlled loop:");

for (var i = 0; i < items.Count; i++)

{

Console.Write($" {items[i]}");

}

Console.Write("\nDisplay list contents with forech statement:");

foreach (var item in items)

{

Console.Write($" {item}");

}

Console.WriteLine();

var endsWithBer =

from item in items

let uppercaseString = item.ToUpper()

where uppercaseString.EndsWith("ber")

orderby uppercaseString

select uppercaseString;

Console.Write("results of query endsendswithber:");

foreach (var item in endsWithBer)

{

Console.Write($" {item}");

}

Console.WriteLine();

Console.Write("items contains:");

foreach (var item in items)

{

Console.Write($" {item}");

}

Console.WriteLine();

Console.Write("results of query endsWithBer");

foreach (var item in endsWithBer)

{

Console.Write($" {item}");

}

Console.WriteLine();

}

}

}