**List Class Example**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

// AddElements();

// RemoveElements();

AddElements1();

Console.ReadLine();

}

public static void AddElements()

{

List<string> obj = new List<string>();

obj.Add("Mango");

obj.Add("Apple");

obj.Add("Banana");

obj.Add("Orange");

obj.Add("Grapes");

obj.Add("Apple");

obj.Add("Apple");

foreach (string s1 in obj)

{

Console.WriteLine(s1);

}

}

public static void RemoveElements()

{

IList<string> obj = new List<string>();

obj.Add("Mango");

obj.Add("Apple");

obj.Add("Banana");

obj.Add("Orange");

obj.Add("Grapes");

obj.Add("Apple");

obj.Add("Apple");

obj.Remove("Orange");

foreach (string s1 in obj)

{

Console.WriteLine(s1);

}

Console.WriteLine("------------------------");

obj.RemoveAt(0);

foreach (string s1 in obj) { Console.WriteLine(s1); }

}

public static void AddElements1()

{

List<string> obj = new List<string>() {"Mango","Apple","Banana","Orange","Apple" };

foreach (string s1 in obj)

{

Console.WriteLine(s1);

}

}

}

}

**LinkedList Example:**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

// AddElements();

RemoveElements();

Console.ReadLine();

}

public static void AddElements()

{

LinkedList<string> obj = new LinkedList<string>();

obj.AddFirst("MAngo");

obj.AddLast("GRapes");

obj.AddLast("Ornage");

obj.AddLast("Apple");

var newNode = obj.AddLast("Kiwi Fruit");

obj.AddBefore(newNode, "Jack Fruit");

var newNode1 = obj.AddLast("Ornage");

obj.AddAfter(newNode1, "Water Melon");

foreach (string s1 in obj)

{

Console.WriteLine(s1);

}

}

public static void RemoveElements()

{

string[] arr = {"Apple","Mango","Banana","Jack Fruit","Orange" };

IList<string> obj = new List<string>(arr); ;

obj.Remove("Orange");

foreach (string s1 in obj) { Console.WriteLine(s1); }

Console.WriteLine("------------------------");

obj.RemoveAt(0);

foreach (string s1 in obj) { Console.WriteLine(s1); }

}

}

}

**HashSet Example:**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

// AddElements();

RemoveElements();

Console.ReadLine();

}

public static void AddElements()

{

HashSet<string> obj = new HashSet<string>();

obj.Add("Apple");

obj.Add("Apple");

obj.Add("Mango");

obj.Add("Banana");

obj.Add("Orange");

obj.Add("Pineapple");

obj.Add("Apple");

foreach (string s1 in obj)

{

Console.WriteLine(s1);

}

}

public static void RemoveElements()

{

string[] arr = {"Apple","Mango","Banana","Jack Fruit","Orange" };

HashSet<string> obj = new HashSet<string>(arr);

obj.Remove("Orange");

foreach (string s1 in obj) { Console.WriteLine(s1); }

Console.WriteLine("------------------------");

}

}

}

**SortedSet Example:**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

// AddElements();

RemoveElements();

Console.ReadLine();

}

public static void AddElements()

{

SortedSet<string> obj = new SortedSet<string>();

obj.Add("Apple");

obj.Add("Water Melon");

obj.Add("Mango");

obj.Add("Banana");

obj.Add("Orange");

obj.Add("Pineapple");

obj.Add("Apple");

foreach (string s1 in obj)

{

Console.WriteLine(s1);

}

}

public static void RemoveElements()

{

string[] arr = {"Apple","Mango","Banana","Jack Fruit","Orange" ,"Apricot"};

SortedSet<string> obj = new SortedSet<string>(arr);

obj.Remove("Orange");

foreach (string s1 in obj) { Console.WriteLine(s1); }

Console.WriteLine("------------------------");

}

}

}

**Stack Example:**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

// StackDemo();

RemoveElement();

Console.ReadLine();

}

public static void StackDemo()

{

Stack<int> obj = new Stack<int>();

obj.Push(100);

obj.Push(200);

obj.Push(300);

obj.Push(400);

obj.Push(500);

obj.Push(600);

obj.Push(700);

foreach (int i in obj) { Console.WriteLine(i); }

}

public static void RemoveElement()

{

Stack<int> obj = new Stack<int>();

obj.Push(100);

obj.Push(200);

obj.Push(300);

obj.Push(400);

obj.Push(500);

obj.Push(600);

obj.Push(700);

int a =obj.Peek();

Console.WriteLine("Peek Eleemnt :" + a);

obj.Pop();

int b = obj.Peek();

Console.WriteLine("Peek Eleemnt :" + b);

}

}

}

**Queue Example:**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

// AddElements();

RemoveElements();

Console.ReadLine();

}

static void AddElements()

{

Queue<int> obj = new Queue<int>();

obj.Enqueue(100);

obj.Enqueue(200);

obj.Enqueue(300);

obj.Enqueue(400);

obj.Enqueue(500);

obj.Enqueue(600);

obj.Enqueue(700);

foreach (int i in obj) { Console.WriteLine(i); }

}

static void RemoveElements()

{

Queue<int> obj = new Queue<int>();

obj.Enqueue(100);

obj.Enqueue(200);

obj.Enqueue(300);

obj.Enqueue(400);

obj.Enqueue(500);

obj.Enqueue(600);

obj.Enqueue(700);

int a=obj.Peek();

Console.WriteLine("Peek Element :" + a);

obj.Dequeue();

int b = obj.Peek();

Console.WriteLine("Peek Element :" + b);

}

}

}

**SortedDictionary Example:**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

// AddElements();

RemoveElements();

Console.ReadLine();

}

static void AddElements()

{

Dictionary<string, string> obj = new Dictionary<string, string>();

obj.Add("lotus", "Lotus is a national flower of India");

obj.Add("camel", "Camel is ship of Desert");

obj.Add("mango", "Mango is a king of Fruits");

obj.Add("tiger", "Tiger is a national Animal");

obj.Add("peacock", "Peacock is a national Bird");

foreach (KeyValuePair<string,string> kv in obj)

{

Console.WriteLine(kv.Key + " ---->" + kv.Value);

}

}

static void RemoveElements()

{

Dictionary<string, string> obj = new Dictionary<string, string>();

obj.Add("lotus", "Lotus is a national flower of India");

obj.Add("camel", "Camel is ship of Desert");

obj.Add("mango", "Mango is a king of Fruits");

obj.Add("tiger", "Tiger is a national Animal");

obj.Add("peacock", "Peacock is a national Bird");

obj.Remove("mango");

obj.Remove("tiger");

foreach (KeyValuePair<string, string> kv in obj)

{

Console.WriteLine(kv.Key + " ---->" + kv.Value);

}

}

}

}

**SortedDictionary Example:**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

AddElements();

// RemoveElements();

Console.ReadLine();

}

static void AddElements()

{

SortedDictionary<string, string> obj = new SortedDictionary<string, string>();

obj.Add("lotus", "Lotus is a national flower of India");

obj.Add("camel", "Camel is ship of Desert");

obj.Add("mango", "Mango is a king of Fruits");

obj.Add("tiger", "Tiger is a national Animal");

obj.Add("peacock", "Peacock is a national Bird");

foreach (KeyValuePair<string,string> kv in obj)

{

Console.WriteLine(kv.Key + " ---->" + kv.Value);

}

}

static void RemoveElements()

{

SortedDictionary<string, string> obj = new SortedDictionary<string, string>();

obj.Add("lotus", "Lotus is a national flower of India");

obj.Add("camel", "Camel is ship of Desert");

obj.Add("mango", "Mango is a king of Fruits");

obj.Add("tiger", "Tiger is a national Animal");

obj.Add("peacock", "Peacock is a national Bird");

obj.Remove("mango");

obj.Remove("tiger");

foreach (KeyValuePair<string, string> kv in obj)

{

Console.WriteLine(kv.Key + " ---->" + kv.Value);

}

}

}

}

**SortedList Example:**

using CSharpTestingExamples.ExamplePrograms;

using OpenQA.Selenium;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium.Chrome;

using System.Threading;

using CSharpTestingExamples.seleniumtests;

namespace CSharpTestingExamples

{

class Program

{

static void Main(string[] args)

{

AddElements();

// RemoveElements();

Console.ReadLine();

}

static void AddElements()

{

SortedList<string, string> obj = new SortedList<string, string>();

obj.Add("lotus", "Lotus is a national flower of India");

obj.Add("camel", "Camel is ship of Desert");

obj.Add("mango", "Mango is a king of Fruits");

obj.Add("tiger", "Tiger is a national Animal");

obj.Add("peacock", "Peacock is a national Bird");

foreach (KeyValuePair<string,string> kv in obj)

{

Console.WriteLine(kv.Key + " ---->" + kv.Value);

}

}

static void RemoveElements()

{

SortedList<string, string> obj = new SortedList<string, string>();

obj.Add("lotus", "Lotus is a national flower of India");

obj.Add("camel", "Camel is ship of Desert");

obj.Add("mango", "Mango is a king of Fruits");

obj.Add("tiger", "Tiger is a national Animal");

obj.Add("peacock", "Peacock is a national Bird");

obj.Remove("mango");

obj.Remove("tiger");

foreach (KeyValuePair<string, string> kv in obj)

{

Console.WriteLine(kv.Key + " ---->" + kv.Value);

}

}

}

}