Lab 4

# Objectives

Practice C# code writing that involves object oriented programming concepts using generics.

# Instructions

* Install Visual Studio 2017 or Visual Studio 2019 to your computer.<https://visualstudio.microsoft.com/>
* For each question below submit your C # and also screenshots showing how your program compiles and executes (sample outputs)- You can upload your C# project but alternatively create a document with your C# code **text** copied and pasted and your screenshots pasted and then upload the document to Blackboard for submission. Prof. Aydin should be able to run your submitted code.
* Your name should appear on the screenshots for receiving full credit.
* **Academic Integrity:** If you are stuck when working in this lab you can collaborate with a couple of classmates. In that case, make sure to write/submit the name of your collaborators and any web site you used as a resource to understand the concepts and lab questions and to complete your code to prevent plagiarism and breach of academic integrity.
  + You are *not* allowed to directly copy code from the Internet, your friends, and other resources without spending any effort in completing the work. Make sure to review the academic integrity policy in the syllabus and ask for clarification, if needed.

1. Take a look at this sample Linked list code [chp5/GenericsSamples/LinkedListSample](https://drive.google.com/drive/folders/1nLBxPUxidq8TE_PMEqcMzBE5f-NRtJ9C?usp=sharing)

This program utilizes generic classes to create a linked list. Your job is to study this code and then modify it as follows.

* Class LinkedList:
  + add a new method called **AddFirst** to class LinkedList
* Main program: show that you can create a linked list for different type of objects
  + One linked list for int’s: use AddFirst method to add numbers 100, 200, 300 and display the list items on the screen
  + One linked list for string’s: use AddFirst method to add strings “one”, “two”, “three” and display the list items on the screen
  + One linked list for Person’s (code for class Person is provided in the same folder): use AddFirst method to add numbers persons with names “Alex Auburn”, “Bob Brown”, “Chuck Chimney” and display the list items on the screen
  + *Hint: when (int, string or Person) items are displayed on the screen, they should appear in the opposite order.*
  + Hint: Review [Week 5 - Chp 5 slides](https://drive.google.com/drive/folders/1-voMzBRitOdXiBNl8SrMm_WM-CbnhxQD?usp=sharing)

using System;

using System.Collections;

namespace Lab4

{

class Program

{

static void Main()

{

var list1 = new LinkedList();

list1.AddFirst(100);

list1.AddFirst(200);

list1.AddFirst(300);

foreach (int i in list1)

{

Console.WriteLine(i);

}

var list2 = new LinkedList();

list2.AddFirst("One");

list2.AddFirst("Two");

list2.AddFirst("Three");

foreach (String i in list2)

{

Console.WriteLine(i);

}

var list3 = new LinkedList();

Person p1 = new Person("Alex", "Auburn");

Person p2 = new Person("Bob", "Brown");

Person p3 = new Person("Chuck", "Chimney");

list3.AddFirst(p1);

list3.AddFirst(p2);

list3.AddFirst(p3);

foreach (Person i in list3)

{

Console.WriteLine(i);

}

}

}

public class LinkedList : IEnumerable

{

public LinkedListNode First { get; private set; }

public LinkedListNode Last { get; private set; }

public LinkedListNode AddLast(object node)

{

var newNode = new LinkedListNode(node);

if (First == null)

{

First = newNode;

Last = First;

}

else

{

newNode.Prev = Last;

Last.Next = newNode;

Last = newNode;

}

return newNode;

}

public LinkedListNode AddFirst(object node)

{

var newNode = new LinkedListNode(node);

if (Last == null)

{

Last = newNode;

First = Last;

}

else

{

newNode.Prev = First;

First.Next = newNode;

First = newNode;

}

return newNode;

}

public IEnumerator GetEnumerator()

{

LinkedListNode current = First;

while (current != null)

{

yield return current.Value;

current = current.Next;

}

}

}

public class LinkedListNode

{

public LinkedListNode(object value) => Value = value;

public object Value { get; }

public LinkedListNode Next { get; internal set; }

public LinkedListNode Prev { get; internal set; }

}

public class Person

{

public string FirstName { get; set; }

public string LastName { get; set; }

public Person(string f = "none", string l = "none")

{

FirstName = f;

LastName = l;

}

public override string ToString() => $"{FirstName} {LastName}";

}

}