**Paul Stuart – C# WEB 000389223 – Lab 16 Research**

1. **How is the LinkedList Class in .NET implemented?**

The List is a doubly Linked list class in the .NET framework.

1. **What methods are available to call and does it differ from the implementations of the list?**

**Add After** - Adds the specified new node after the specified existing node. There is also a constructor for specifying a value after the existing node.

**Add Before** – Same as Add After but for before rather than after.

**Add First** – Adds the specified Node at the start of the LinkedList. Also, a constructor for a specified value.

**Add Last** – Same as above but last rather than first

**Clear** – Removes all functions from the list

**Contains** – Search Function for a value in the LinkedList

**Copy To** – Copies the entire LinkedList to a one-dimensional Array

**Equals** – Checks Object equality in the list

**Find** – Finds first node that contains specified value

**Find Last** – As above but for the last item in the Linked list

**Get Enumerator** – Returns an enumerator that iterates through the List

**Get Hash Code** – Serves as default hash function

**Get Object Data** - Implements the I-Serializable interface and returns the data needed to serialize the LinkedList<T> instance.

**Get Type** – Inherited from Object, it collects the type of the current instance

**Member wise Clone** – Creates a shadow copy of the current Object

**On Deserialization** - Implements the I-Serializable interface and raises the deserialization event when the deserialization is complete.

**Remove** – Function for both linked list node and for an occurrence.

**Remove First** – Removes the first node in the list

**Remove Last** – Removes the last node in the list

**To String** – outputs String representation of the current Object

1. What is the main advantage of using Doubly Linked Lists over Single Linked options?

A DLL can be traversed in both forward and backward direction. The delete operation in DLL is more efficient if pointer to the node to be deleted is given. We can also quickly insert a new node before a given node.

In singly linked list, to delete a node, pointer to the previous node is needed. To get this previous node, sometimes the list is traversed. In DLL, we can get the previous node using previous pointer.