Erinn Neidinger

**Overloading the – Operator**

**Function**: The – operator subtracts the instance in List One from the instance in List Two. The operator will find the value that is the same within the two lists and remove it.

**Parameters**: (List<T> listone, List<T> listtwo). Two lists will be passed through in order to compare the instances of both.

**Syntax**: List<int> result = listone - listtwo;

**Returns**: A List containing the remaining contents of List One. For the below example, there is a Custom List that was created to place the remaining contents.

**Example:**

CustomList<int> customlist = new CustomList<int>();

CustomList<int> listone = new CustomList<int>();

CustomList<int> listtwo = new CustomList<int>();

**Adding on to lists one and two:**

listone.Add(1);

listone.Add(2);

listone.Add(3);

listtwo.Add(2);

listtwo.Add(5);

listtwo.Add(6);

**What each list should look like:**

**List One**: 123 **List Two**: 256

List<int> result = listone - listtwo;

public static CustomList<T> operator -(CustomList<T> listone, CustomList<T> listtwo) //comparing lists one and two

{

CustomList<T> result = new CustomList<T>(); //creating a new list for the sum

for (int i = 0; i < listone.count; i++) //each character in listone will be tested for a match with listtwo.

{

bool NotEqual = true;

for (int j = 0; j < listtwo.count; j++)

{

if (listone[i].Equals(listtwo[j])) //if a match is found, the character will be subtracted or removed from the list.

{

NotEqual = false;

}

}

if (NotEqual == true) //if no match is found, the character will be added to the results list.

{

result.Add(listone[i]);

}

}

return result; //returns the sum

}

**Output**:

Result: 13