**ScratchList<T> -operator overload**

For a given ScratchList<T> ‘A’, it removes the earliest incident of each item in a second ScratchList<T> B provided that item appears in both ScratchList.

**Syntax:**

(ScratchList<T> A) – ( ScratchList<T> B)

**Parameters:**

ScratchList<T> A

The initial list, values will be removed from this list

ScratchList<T> B

For each item in this list, if it also appears in A, it will be removed from A.

Type T must be the same data type for both ScratchLists.

**Returns:**

ScratchList<T>

T will be of same type as both ScrachList A and ScratchList B

**Examples:**

namespace ListFromScratch

{

class Program

{

static void Main(string[] args)

{

//instantiate ScratchList A {1, 2, 3, 2, 4, 5, 2, 6}

ScratchList<int> A = new ScratchList<int>();

A.Add(1);

A.Add(2);

A.Add(3);

A.Add(2);

A.Add(4);

A.Add(5);

A.Add(2);

A.Add(6);

//write contents of list to console

Console.WriteLine("ScratchList A: " + A.ToString());

//instantiate ScratchList B {2, 4, 6, 8, 2}

ScratchList<int> B = new ScratchList<int>();

B.Add(2);

B.Add(4);

B.Add(6);

B.Add(8);

B.Add(2);

//write contents of list to console

Console.WriteLine("ScratchList B: " + B.ToString());

//declare new, resultant ScratchList

ScratchList<int> C;

//subtract ScratchLists, set to C

C = A - B;

Console.WriteLine("Resulting ScratchList: " + C.ToString());

//Output:

//ScratchList A: 12324526

//ScratchList B: 24682

//Resulting ScratchList: 1352

//

//\*\*note

//2, 2, 4, 6 were removed from SCratchList A

//the first 2 in ScratchList B triggered removing the first 2 in ScratchList A

//the second 2 in ScratchList B triggered removing the second 2 in ScratchList A

//the third 2 in ScratchList A remains

//the 8 in ScratchList B was ignored since it was not in ScratchList A

}