

## Lab Assignment 2

Jerry Belmonte

Keira Wong

---

# CODE (x2 files)

## IntegerSetLibrary.cs

```
using System;
using System.Text;

namespace IntegerSetLibrary
{
    public class IntegerSet
    {
        private const int SET_SIZE = 101;
        private bool[] mSet;

        // Parameterless Constructor
        public IntegerSet()
        {
            mSet = new bool[SET_SIZE];
        }

        // Parameterized Constructor
        public IntegerSet(int[] array) : this()
        { // "this()" : calls its parameterless constructor too
            for (int i = 0; i < array.Length; i++)
            {
                InsertElement(array[i]);
            }
        }

        // ValidEntry() is a helper method
        private bool ValidEntry(int integerValue)
        {
            // Check if the given integer value is in the range 0-100.
            return integerValue >= 0 && integerValue < SET_SIZE;
        }

        // InsertElement()
        public void InsertElement(int integerToInsert)
        {
            // if "integerToInsert" isValid
```

```

    if (ValidEntry(integerToInsert))
    {
        // add "integerToInsert" to "mSet"
        mSet[integerToInsert] = true;
    }
}

// DeleteElement()
public void DeleteElement(int integerToDelete)
{
    // if "integerToDelete" isValid
    if (ValidEntry(integerToDelete))
    {
        // remove "integerToDelete" from "mSet"
        mSet[integerToDelete] = false;
    }
}

// Union()
public IntegerSet Union(IntegerSet otherSet)
{
    IntegerSet unionSet = new IntegerSet();

    for (int i = 0; i < SET_SIZE; i++)
    {
        // If this mSet[i] is true or otherSet.mSet[i] is true,
        // then make the unionSet.mSet[i] to be true as well.
        unionSet.mSet[i] = (mSet[i] || otherSet.mSet[i]);
    }
    return unionSet;
}

// Intersection()
public IntegerSet Intersection(IntegerSet integerSet)
{
    IntegerSet intersection = new IntegerSet();

    for (int i = 0; i < SET_SIZE; i++)
    {
        // if this mSet[i] == integerSet's mSet[i] then they intersect @ i.
        intersection.mSet[i] = (mSet[i] && integerSet.mSet[i]);
    }
    return intersection;
}

// IsEqualTo()
public bool IsEqualTo(IntegerSet otherSet)
{
    bool bothEqual = false;

    for (int i = 0; i < SET_SIZE; i++)
    {
        bothEqual = mSet[i] == otherSet.mSet[i];

        if (!bothEqual)
            break;
    }
}

```

```

    }

    return bothEqual;
}

// ToString()
public override string ToString()
{
    StringBuilder toString = new StringBuilder();
    toString.Append("{");

    for (int i = 0; i < SET_SIZE; i++)
    {
        if (mSet[i])
        {
            toString.Append(" " + i);
        }
    }

    // Check if the set is empty.
    if (toString.Length.Equals(1))
        toString.Append("---");
    else
        toString.Append(" ");

    return toString.ToString();
}
}
}

```

## Program.cs

```

using System;
using IntegerSetLibrary;

namespace LabAssignment2
{
    class Program
    {
        // MAIN PROGRAM
        static void Main(string[] args)
        {
            // 1. Initialize 2 sets.
            Console.WriteLine("Input Set A");
            IntegerSet set1 = InputSet();
            Console.WriteLine("\nInput Set B");
            IntegerSet set2 = InputSet();

            IntegerSet union = set1.Union(set2);
            IntegerSet intersection = set1.Intersection(set2);

            // 2. Prepare output.
            Console.WriteLine("\nSet A contains elements:");

```

```

Console.WriteLine(set1.ToString());
Console.WriteLine("\nSet B contains elements:");
Console.WriteLine(set2.ToString());
Console.WriteLine(
"\nUnion of Set A and Set B contains elements:");
Console.WriteLine(union.ToString());
Console.WriteLine(
"\nIntersection of Set A and Set B contains elements:");
Console.WriteLine(intersection.ToString());

// test whether two sets are equal
if (set1.IsEqualTo(set2))
    Console.WriteLine("\nSet A is equal to set B");
else
    Console.WriteLine("\nSet A is not equal to set B");

// test insert and delete
Console.WriteLine("\nInserting 77 into set A...");
set1.InsertElement(77);
Console.WriteLine("\nSet A now contains elements:");
Console.WriteLine(set1.ToString());

Console.WriteLine("\nDeleting 77 from set A...");
set1.DeleteElement(77);
Console.WriteLine("\nSet A now contains elements:");
Console.WriteLine(set1.ToString());

// test constructor
int[] intArray = { 25, 67, 2, 9, 99, 105, 45, -5, 100, 1 };
IntegerSet set3 = new IntegerSet(intArray);

Console.WriteLine("\nNew Set contains elements:");
Console.WriteLine(set3.ToString());

} // END of MAIN

// MAIN FUNCTIONS
public static IntegerSet InputSet()
{
    IntegerSet temp = new IntegerSet();

    int number = 0;
    while (number != -1)
    {
        Console.Write("Enter a number (-1 to END): ");
        number = Convert.ToInt32(Console.ReadLine());
        temp.InsertElement(number);
    }

    return temp;
}
}

```

# OUTPUT

```
Microsoft Visual Studio Debug Console

Input Set A
Enter a number (-1 to END): 3
Enter a number (-1 to END): 5
Enter a number (-1 to END): 20
Enter a number (-1 to END): 35
Enter a number (-1 to END): 60
Enter a number (-1 to END): 99
Enter a number (-1 to END): -1

Input Set B
Enter a number (-1 to END): 5
Enter a number (-1 to END): 35
Enter a number (-1 to END): 99
Enter a number (-1 to END): 56
Enter a number (-1 to END): 82
Enter a number (-1 to END): 101
Enter a number (-1 to END): -1

Set A contains elements:
{ 3 5 20 35 60 99 }

Set B contains elements:
{ 5 35 56 82 99 }

Union of Set A and Set B contains elements:
{ 3 5 20 35 56 60 82 99 }

Intersection of Set A and Set B contains elements:
{ 5 35 99 }

Set A is not equal to set B

Inserting 77 into set A...

Set A now contains elements:
{ 3 5 20 35 60 77 99 }

Deleting 77 from set A...

Set A now contains elements:
{ 3 5 20 35 60 99 }

New Set contains elements:
{ 1 2 9 25 45 67 99 100 }

C:\Users\keira\source\repos\LabAssignment2\LabAssignment2\bin\Debug\netcoreapp3.1\LabAssignment2.exe (process 12412) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```