Selected files

2 printable files

Week_2\2.2\Program.cs
Week_2\2.2\Counter.cs

Week_2\2.2\Program.cs

```
using System;
namespace CounterTask
{
    internal class Program
    {
        private static void PrintCounters(Counter[] counters)
        {
            foreach (Counter c in counters)
            {
                Console.WriteLine($"{c.Name} is {c.Ticks}");
            }
        }

        static void Main(string[] args)
        {
            Counter[] myCounters = new Counter[3];
            myCounters[0] = new Counter("Counter 1");
            myCounters[1] = new Counter("Counter 2");
            myCounters[2] = myCounters[0];

            for (int i = 1; i <= 9; i++)
            {
                myCounters[0].Increment();
            }

            for (int i = 1; i <= 14; i++)
            {
                myCounters[1].Increment();
            }

            PrintCounters(myCounters);
            myCounters[2].Reset();
            PrintCounters(myCounters);

            Console.WriteLine("\nResetting Counter 2 by default and then incrementing it:");
            myCounters[1].ResetByDefault();
            myCounters[1].Increment();
            Console.WriteLine($"{myCounters[1].Name} is {myCounters[1].Ticks}");

        }
    }
```

```
}
```

Week_2\2.2\Counter.cs

```
public class Counter
{
    private int _count;
    private string _name;

    public Counter(string name)
    {
        _name = name;
        _count = 0;
    }

    public void Increment()
    {
        _count++;
    }

    public void Reset()
    {
        _count = 0;
    }

    public void ResetByDefault()
    {
        unchecked
        {
            _count = (int)2147483647794;
        }
    }

    public string Name
    {
        get
        {
            return _name;
        }

        set
        {
            _name = value;
        }
    }

    public int Ticks
    {
        get
        {
            return _count;
```

```
48          }
49      }
50  }
```

Question 13:

- Overflow Handling: The "unchecked" keyword in ResetByDefault allows setting _count to a value exceeding the int limit without throwing an exception. Instead, it wraps around to a valid int value.

- The Increment method still works correctly after the overflow, continuing without crashes. However, The value -205 occurs because 2147483647794 exceeds the int limit in C#, when overflow happens, it wraps around to -206. Incrementing it by 1 results in -205.

Screenshot of Final Output: