

## Module 4 - Class Method Assignment - Model Answer

```
//MathMethods.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CallingMethodsAssignment
{
    public class MathMethods
    {
        //Create three methods that take one integer parameter and return an integer
        public static int Heal(int magic)
        {
            int hitpoints = magic + 10;
            return hitpoints;
        }
        public static int Defense(int defense)
        {
            int enemy_str = 7;
            int weapon_mt = 11;
            int triangle_bonus = -1;
            int crit_coeff = 1;
            int damage = (enemy_str + (weapon_mt + triangle_bonus) - defense) * crit_coeff;
            return damage;
        }
        public static int Attack(int strength)
        {
            int enemy_def = 4;
            int weapon_mt = 8;
            int triangle_bonus = 1;
            int crit_coeff = 3;
            int damage = (strength + (weapon_mt + triangle_bonus) - enemy_def) * crit_coeff;
            return damage;
        }
    }
}
```

```
using System;
```

```
namespace ClassMethodAssignment  
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            //In the Main() method, instantiate the class
```

```
            VoidMethod math = new VoidMethod();
```

```
            //Call the void method
```

```
            Console.WriteLine("Input a number:");
```

```
            int dividend = Convert.ToInt32(Console.ReadLine());
```

```
            math.MathOp(dividend);
```

```
            Console.WriteLine(math.Value);
```

```
            Console.ReadLine();
```

```
            //Call the method with output parameters
```

```
            Console.WriteLine("Input a phrase:");
```

```
            string input = Console.ReadLine();
```

```
            string phrase = math.StringCounter(input, out int count);
```

```
            Console.WriteLine(phrase);
```

```
            Console.WriteLine("Doubled would be: " + (count * 2));
```

```
            Console.ReadLine();
```

```
            //Call the overloaded method
```

```
            Console.WriteLine("Input another phrase:");
```

```
            string input2 = Console.ReadLine();
```

```
            int phrase2_len = math.StringCounter(input2);
```

```
            Console.WriteLine("This phrase is " + phrase2_len + " characters long");
```

```
            Console.ReadLine();
```

```
            //Call the method of the static class
```

```
            Console.WriteLine("Input another phrase:");
```

```
            string input3 = Console.ReadLine();
```

```
            int i_count = VoidMethod.IFinder(input3);
```

```
            Console.WriteLine("This phrase has " + i_count + " i's in it.");
```

```
            Console.ReadLine();
```

```
        }
```

```
    }
```

```
}
```