```
using System;

namespace Methods
{
    class MethodClass
    {
        //we use return to return a method value
        //we define input parameters
        //careful that you pass the exact parameter type or else the program
        //will crash
        static double GetRectangleArea(double width, double height)
        {
            double area = width * height;
            return area;
        }
        //we can return multiple values
        static int CompareTo(int number1, int number2)
        {
            if (number1 > number2)
            {
                return 1;
            }
            else if (number1 == number2)
            {
                return 0;
            }
            else
            {
                return -1;
            }
        }
        //we can return nothing but still give an output
        static void PrintLogo()
        {
            Console.WriteLine("TheSchaub");
            Console.WriteLine("www.theschaub.ca");
        }
        //we can pass in arrays
        static void PrintTotalAmount(double[] prices)
        {
            double totalAmount = 0;
            foreach (double totalPrice in prices)
            {
                totalAmount += totalPrice;
            }
            Console.WriteLine("The total amount for all books is: " +
            totalAmount);
        }
        //we can have dependant outputs
        static void PrintSign(int number)
        {
            if (number > 0)
            {
                Console.WriteLine("Positive");
            }
            else if (number < 0)
            {
                Console.WriteLine("Negative");
            }
            else
            {
                Console.WriteLine("Zero");
            }
        }
        //we can redefine and use passed in parameters
        static void PrintMax(float number1, float number2)
        {
            float max = number1;
            if (number2 > max)
            {
                max = number2;
            }
            Console.WriteLine("Biggest number is: " + max);
        }
        //if we do not know how many parameters we are passing into the method
        //we can use the "params" argument
        //in our prices example it was asumed that the array with a SET number of
elements
        //had already been created... just as we did...
        //below the array will not be premade
        static long CalcSum(params int[] elements)
        {
            long sum = 0;
            foreach (int element in elements)
            {
                sum += element;
            }
            return sum;
        }
        //we can have optional parameters by redefining the optional ones
        static void SomeMethod(int x, int y = 5, int z = 7)
        {
            Console.WriteLine("x-value: "+ x +" y-value: "+y+" z-value: "+z);
        }
        //we can overload so methods so it does not matter what the user inputs
        static void PrintContent(string str)
        {
            Console.WriteLine(str);
        }
        static void PrintContent(int number)
        {
            Console.WriteLine(number);
        }
        static void PrintContent(float number)
        {
            Console.WriteLine(number); //can abuse variables as they only exist in
method
        }

        static void Main(string[] args)
        {
            Console.WriteLine(GetRectangleArea(3,4));
            double areaGotten = GetRectangleArea(5, 6);
            Console.WriteLine(areaGotten);
            Console.WriteLine(CompareTo(3, 4));
            PrintLogo();
            double[] foodPrice = { 4.50, 3.80, 7.40 };
            PrintTotalAmount(foodPrice);
            //PrintTotalAmount(4, 5, 6, 7); DOES NOT TAKE 4 ARGUMENTS
            PrintSign(-1);
            PrintSign(1);
            PrintMax(4, 5);
            Console.WriteLine("The sum is: " + CalcSum(4, 5, 6, 7));
            Console.WriteLine("The sum is: " + CalcSum()); // can even handle no
args
            SomeMethod(123);
            SomeMethod(1,2);
            SomeMethod(1, 2,3);
            PrintContent("Banana");
            PrintContent(5);
        }

    }

}
```