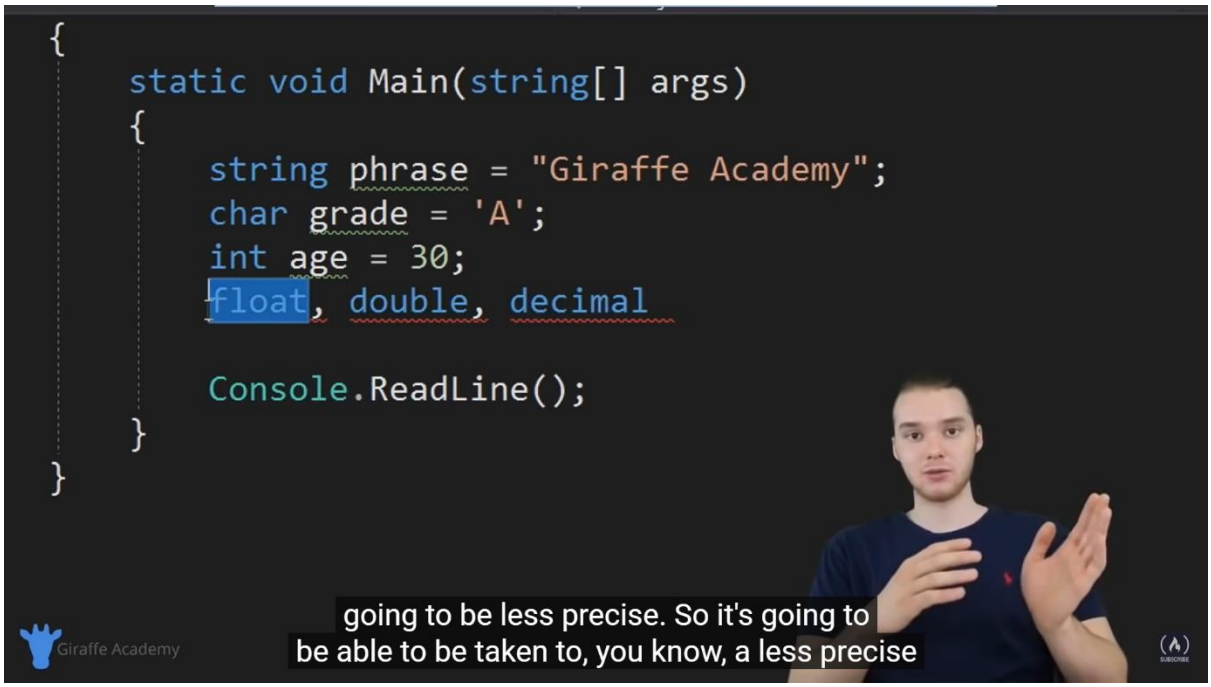
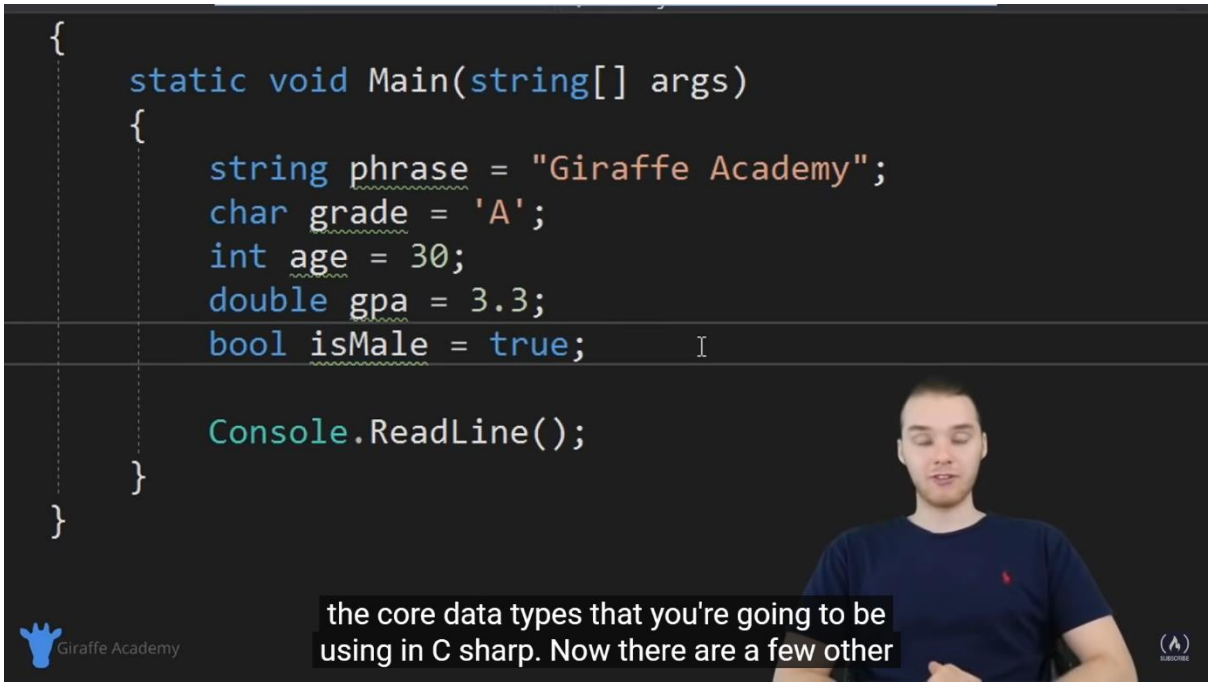
Data Types


{
static void Main(string[] args)
{
string phrase = "Giraffe Academy";
char grade = 'A';
int age = 30;
float, double, decimal
Console.ReadLine();
}
}
Giraffe Academy
going to be less precise. So it's going to
be able to be taken to, you know, a less precise


{
static void Main(string[] args)
{
string phrase = "Giraffe Academy";
char grade = 'A';
int age = 30;
double gpa = 3.3;
bool isMale = true;
Console.ReadLine();
}
}
Giraffe Academy
the core data types that you're going to be
using in C sharp. Now there are a few other

String Class Properties and Methods

String Class Properties

Properties	Description
Chars	Gets the Char object at a specified position in the current String object..
Length	Gets the number of characters in the current String object.

Common String Class Methods

Properties	Description
Clone()	Make clone of string.
CompareTo()	Compares two specified String objects and returns an integer that indicates their relative position in the sort order.
Contains()	Returns a value indicating whether a specified substring occurs within this string.
EndsWith()	Determines whether the end of this string instance matches the specified string.
Equals()	Determines whether this instance and another specified String object have the same value
IndexOf(String)	Reports the zero-based index of the first occurrence of the specified string in this instance.
ToLower()	Returns a copy of this string converted to lowercase.
ToUpper()	Returns a copy of this string converted to uppercase.
Insert()	Returns a new string in which a specified string is inserted at a specified index position in this instance.
LastIndexOf(String)	Reports the zero-based index position of the last occurrence of a specified string within this instance.
Remove()	This method deletes all the characters from beginning to specified index position.
Replace()	This method helps to replace the character.
Split()	This method splits the string based on specified value.
StartsWith(String)	Determines whether the beginning of this string instance matches the specified string.
Substring()	Retrieves a substring from this instance. The substring starts at a specified character position and continues to the end of the string.
Trim()	It removes extra whitespaces from beginning and ending of string.

Getting Input and Working with string

```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        //Getting Input
        Console.WriteLine("Welcome");
        Console.Write("Enter Your name: ");
        string name = Console.ReadLine();
        Console.Write("Enter Your age: ");
        string age = Console.ReadLine();
        Console.WriteLine("Hello " + name + " you are " + age);

        Console.ReadLine();
    }
}
```

```
C:\Users\Admin\Desktop\CS_
Welcome
Enter Your name: Rohit
Enter Your age: 20
Hello Rohit you are 20
```

```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        String CharcterName = " Hello, \n Rohit Tekchandani \" Nice to meet you \" ";
        Console.WriteLine(CharcterName);

        Console.WriteLine(" " + "four".Length); // Methods
        Console.WriteLine(" " + "four".ToUpper()); // Methods
        Console.WriteLine(" " + "four".ToLower()); // Methods
        Console.WriteLine(" " + "four".Contains("five")); // Methods
        Console.WriteLine(" " + "four five six seven".Split(" ")[2]); // Methods

        Console.WriteLine(" " + "four"[2]); // Methods
                        //Index 0123 --> u
        Console.WriteLine(" " + "four five six".IndexOf("five")); // Methods
                        //Index 0123456789 --> 5
        Console.WriteLine(" " + "four five six".IndexOf("seven")); // Methods
        Console.WriteLine(" " + "four five six".Substring(5,4)); // (Index, number of charecter)

        Console.Read();
    }
}
```

```
C:\Users\Admin\Desktop\CS_syntex\HelloWorld\
Hello,
Rohit Tekchandani " Nice to meet you "
4
FOUR
four
False
six
u
5
-1
five
```

```
        Console.WriteLine(Math.);
        Console.Read();
    }
}
```

★ Min
★ Max
★ Clamp
Abs
Acos
Acosh
Asin
Asinh
Atan

int Math.Min(int val1, int val2) (+ 10 overloads)
Returns the smaller of two 32-bit signed integers.
★ IntelliCode suggestion based on this context

```
2   0
ebug
(CoreCLR: clrhost): Loaded 'C:\Program F                \Microsoft.NETCore.App\6.0.12\System.Runtime.InteropServices.d
(CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\6.0.12\System.Security.Claims.dll'. Ski
(CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\6.0.12\Microsoft.Win32.Primitives.dll'.
```

Working With Numbers

```
class Program
{
    0 references
    static void Main(string[] args)
    {
        //Working with numbers
        Console.WriteLine(5 + 2 * 3); // Bodmas
        Console.WriteLine((5 + 2) * 3); // Pernthisis
        Console.WriteLine(5/2); // int/int
        Console.WriteLine(5/2.0); // int/decimal

        int num = 10;
        Console.WriteLine(num);
        num++;
        Console.WriteLine(num);
        num--;
        Console.WriteLine(num);

        Console.WriteLine(Math.Pow(num,2));
        Console.WriteLine(Math.Sqrt(num));
        Console.WriteLine(Math.Round(5.3269,3)); // 3 digits after decimal point
        Console.WriteLine(Math.Max(2.56,3)); // only for 2 digits
        // ERROR: Console.WriteLine(Math.Max(2,3,4));

        Console.Read();
```

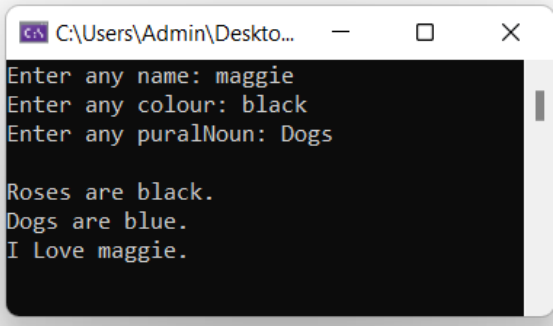
```
C:\Users\Admin\Deskt
11
21
2
2.5
10
11
10
100
3.1622776601683795
5.327
3
```

```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        //Calculater app
        //Console.WriteLine(Convert.ToInt32("45") + 6); // string to int
        Console.Write("Enter a number: ");
        decimal num1 = Convert.ToDecimal( Console.ReadLine() );

        Console.Write("Enter another number: ");
        decimal num2 = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine(num1 + num2);
        Console.WriteLine(num1 - num2);
        Console.WriteLine(num1 * num2);
        Console.WriteLine(num1 / num2);
        Console.WriteLine(num1 % num2);

        Console.ReadLine();
    }
```

```
C:\Users\Admin\Desktop\CS_syntex\Hello
Enter a number: 2.3
Enter another number: 45
47.3
-42.7
103.5
0.051111111111111111111111111
2.3
```

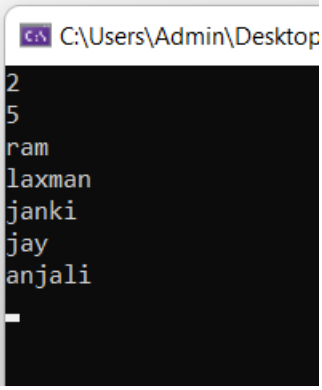
Exercise: Mad Libs

```
static void Main(string[] args)
{
    //Mad Libs - random string in paragraphs to change it's context

    String colour, puralNoun, name;

    Console.Write("Enter any name: ");
    name = Console.ReadLine();
    Console.Write("Enter any colour: ");
    colour = Console.ReadLine();
    Console.Write("Enter any puralNoun: ");
    puralNoun = Console.ReadLine();

    Console.WriteLine();

    Console.WriteLine("Roses are "+ colour +".");
    Console.WriteLine(puralNoun +" are blue.");
    Console.WriteLine("I Love " + name + ".");

    Console.ReadLine();
}
```



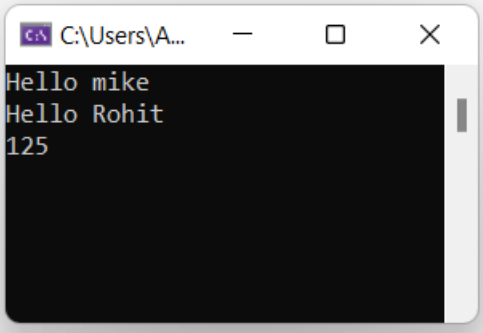
Arrays

```
static void Main(string[] args)
{
    //Arrays
    int[] numbers = { 1, 2, 3, 4};
    Console.WriteLine(numbers[1]);
    numbers[1] = 5; //chinging the value
    Console.WriteLine(numbers[1]);

    string[] friends = new string[6]; // number of elements
    friends[0] = "ram";
    friends[1] = "laxman";
    friends[2] = "janki";
    friends[3] = "jay";
    friends[4] = "anjali";
    friends[5] = "kirti";
    Console.WriteLine(friends[0]);
    Console.WriteLine(friends[1]);
    Console.WriteLine(friends[2]);
    Console.WriteLine(friends[3]);
    Console.WriteLine(friends[4]);
    Console.ReadLine();
}
```



Method

```
class Program
{
    0 references
    static void Main(string[] args)
    {
        //Methods
        string user = "mike";
        SayHi(user); // calling the methods
        SayHi("Rohit"); // calling the methods
        Console.WriteLine(Cube(5)); // calling the methods
        Console.ReadLine();
    }
    // Method defination
    // static returnType Name(prarmeters)
    2 references
    static void SayHi(string name)
    {
        Console.WriteLine("Hello " + name);
    }
    1 reference
    static int Cube(int num)
    {
        int result = num * num * num;
        return result;
    }
}
```



If else

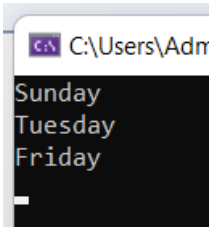
```
class Program
{
    static void Main(string[] args)
    {

        bool isMale = true;
        bool isTall = true;

        if (isMale || isTall)

        if (isMale && isTall)
        {
            Console.WriteLine("You are a tall male");
        } else if (isMale && !isTall){
            Console.WriteLine("You are a short male");
        } else if (!isMale && isTall) {
            Console.WriteLine("You are not a male but you are tall");
        } else
        {
            Console.WriteLine("You are not male and not tall");
        }


        Console.ReadLine();
    }
```

Operators

> , < , <= , >=, !, !=, ==, &&, ||, =, ….

Switch statement

```
static void Main(string[] args)
        {
            Console.WriteLine(GetDay(0));
            Console.WriteLine(GetDay(2));
            Console.WriteLine(GetDay(5));
            Console.ReadLine();
        }
static string GetDay(int dayNum)
        {
            string dayName;
            switch (dayNum)
            {
                case 0:
                    dayName = "Sunday";
                    break;
                case 1:
                    dayName = "Monday";
                    break;
                case 2:
                    dayName = "Tuesday";
                    break;
                case 3:
                    dayName = "Wednesday";
                    break;
                case 4:
                    dayName = "Thursday";
                    break;
                case 5:
                    dayName = "Friday";
                    break;
                case 6:
                    dayName = "Saturday";
                    break;
                default:
                    dayName = "INVALID DAY NUMBER";
                    break;
            }
            return dayName;
        }
```



while, do while and for

```
int index = 6;
while (index <= 5)
{
    Console.WriteLine(index);
    index++;
}
```

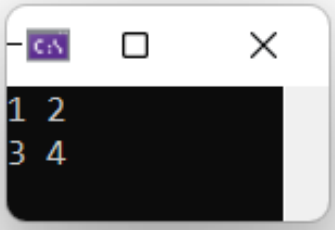
(local variable) int index

```
int index = 6;
do
{
    Console.WriteLine(index);
    index++;
} while (index <= 5);
```

```
for(int i = 1; i <= 5; i++)
{
    Console.WriteLine(i);
}
```

2D Array

```
class Program
{
    0 references
    static void Main(string[] args)
    {
        // 2D array
        int[,] myArray = new int[2, 2];
        int[,] numberGrid =
        {
            {1,2},
            {3,4}
        };
        myArray = numberGrid;

        for(int i = 0;i< 2; i++)
        {
            for(int j = 0;j< 2; j++)
            {
                Console.Write(numberGrid[i,j]+" ");
            }
            Console.WriteLine();
        }
        Console.ReadLine();
    }

}
```

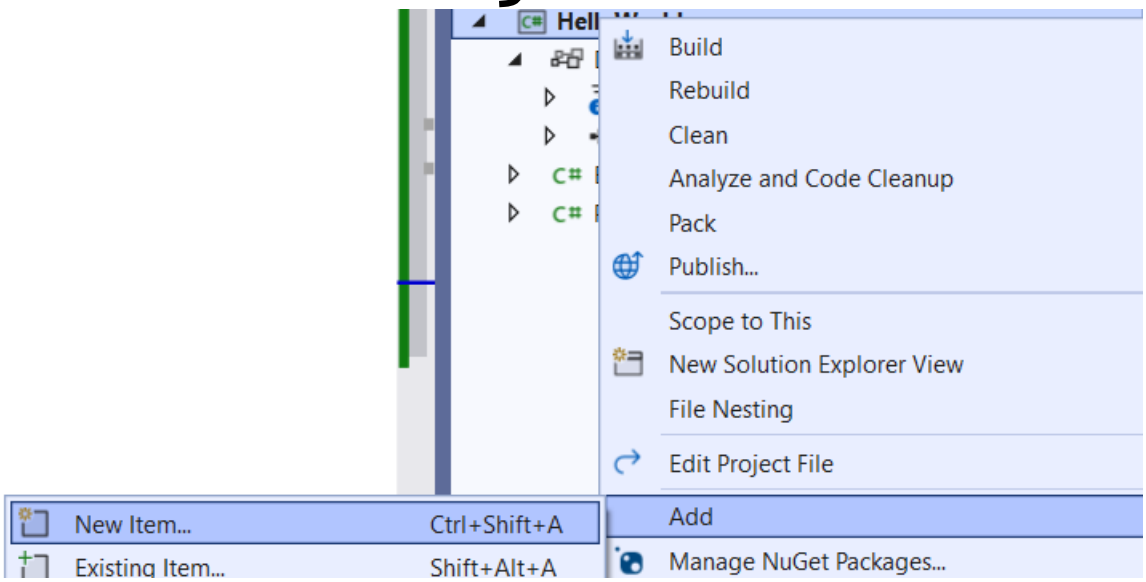


Comments -> /* … */ , // …

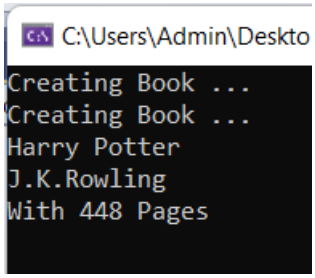
Exception Handling

```
static void Main(string[] args)
        {
            // Exception handling
            try
            {
                Console.WriteLine("Enter a number: ");
                int num1 = Convert.ToInt32(Console.ReadLine());

                Console.WriteLine("Enter another number: ");
                int num2 = Convert.ToInt32(Console.ReadLine());

                Console.WriteLine(num1 / num2);
            }
            catch(DivideByZeroException e)
            {
                Console.WriteLine(e.Message);
            }
            catch (FormatException e)
            {
                Console.WriteLine(e.Message);
            }

            catch (Exception e)
            {
                Console.WriteLine(e.ToString());
                Console.WriteLine(e.Message);
            }
            finally // Compulsory
            {
                Console.ReadLine();
            }
        }
```

Classes and Objects



```
class Program
{
    static void Main(string[] args)
    {
        // Classes and Objects
        Book book1 = new Book();
        book1.title = "Harry Potter";
        book1.author = "J.K.Rowling";
        book1.pages = 448;

        Book book2 = new Book();
        book2.title = "Load Of The Ring";
        book2.author = "Tolkein";
        book2.pages = 742;

        Console.WriteLine(book1.title + "\n" +book1.author);
        Console.WriteLine("With " + book1.pages + " Pages");

        Console.ReadLine();
    }

 }

class Book
{
    public string title;
    public string author;
    public int pages;
}
```

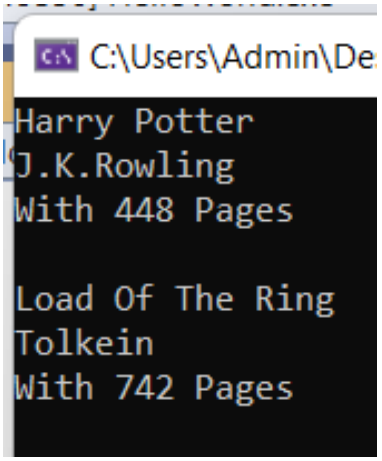


Constructors

```
class Book
{
    public string title;
    public string author;
    public int pages;

    public Book() // Constructor
    {
        Console.WriteLine("Creating Book ...");
    }
}
```

-> Constructors helps to initialise classes

```
    class Program
    {
        static void Main(string[] args)
        {
            // Classes and Objects
            Book book1 = new Book("Harry Potter", "J.K.Rowling", 448);
            Book book2 = new Book("Load Of The Ring", "Tolkein", 742);


            Console.WriteLine(book1.title + "\n" +book1.author);
            Console.WriteLine("With " + book1.pages + " Pages");
            Console.WriteLine("");
            Console.WriteLine(book2.title + "\n" + book2.author);
            Console.WriteLine("With " + book2.pages + " Pages");

            Console.ReadLine();
        }
    }

    class Book
    {
        public string title;
        public string author;
        public int pages;

        public Book(string aTitle, string aAuthor, int aPages)
// Constructor // 'a' stands for argument
        {
            title = aTitle;
            author = aAuthor;
            pages = aPages;
        }
    }
```



-> To update class we can use .= method

```
Book book1 = new Book("Harry Potter", "JK Rowling",
Book book2 = new Book("Lord Of the Rings", "Tolkein"

book2.title = "The hobbit";
```

Object Methods

```
class Program
{
    static void Main(string[] args)
    {
        // Object Methods
        Student s1 = new Student("Rohit","CS",5.0);
        Student s2 = new Student("Jim","EL",2.75);
        Student s3 = new Student("Sonu","EC",3.10);

        Console.WriteLine(s1.hasHoners());
        Console.WriteLine(s2.hasHoners());
        Console.WriteLine(s3.hasHoners());

        Console.ReadLine();
    }
}

class Student
{
    public string name;
    public string major;
    public double gpa;

    public Student(string name, string major, double gpa)
    {
        this.name = name;
        this.major = major;
        this.gpa = gpa;
    }

    public bool hasHoners()
    {
        if (gpa > 3.0) { return true; }
        return false;
    }
}
```

OUTPUT:

True

False

True

Getter and Setter

-> To make some parametes more secure with specific conditions.

```
class Program
{
    static void Main(string[] args)
    {
        // Object Methods
        Student s1 = new Student("Rohit","CS",5.0);
        Student s2 = new Student("Jim","EL",2.75);
        Student s3 = new Student("Sonu","EC",3.10);

        s3.GPA = 33;

        Console.ReadLine();
    }
}

class Student
{
    public string name;
    public string major;
    private double gpa;

    public Student(string name, string major, double gpa)
    {
        this.name = name;
        this.major = major;
        this.gpa = GPA;
    }

    public double GPA
    { get { return gpa; }
      set
        {
            if (value >= 0 && value <= 5)
            {
                gpa = value;
            }
            else
            {
                Console.WriteLine("ERROR IN GPA ENTRY");
            }
        }
    }

}
```

OUTPUT:

ERROR IN GPA ENTRY

Static attribute - visehshtah

```
class Program
    {
        static void Main(string[] args)
        {
            // Object Methods
            Student s1 = new Student("Rohit","CS",5.0);
            Console.WriteLine(Student.stcount);
            Student s2 = new Student("Jim","EL",2.75);
            Console.WriteLine(Student.stcount);
            Student s3 = new Student("Sonu","EC",3.10);
            Console.WriteLine(Student.stcount);

            Console.ReadLine();
        }
    }

class Student
    {
        public string name;
        public string major;
        public double gpa;
        public static int stcount = 0;

        public Student(string name, string major, double gpa)
        {
            this.name = name;
            this.major = major;
            this.gpa = gpa;
            stcount++;
        }

    }
```

One perameter per class. Same for all objects of one class.

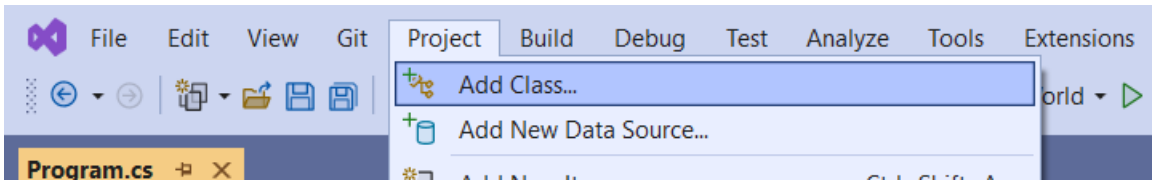
Output :

1

2

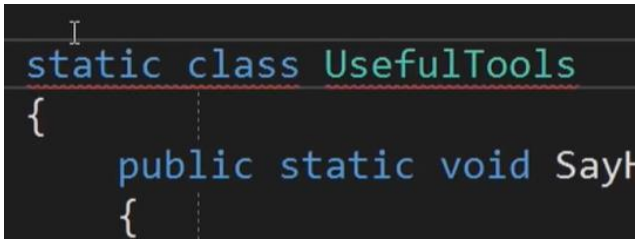
3

Static Methods:




```
// static Methods
// Student s1 = new Student("Rohit","CS",5.0);
// no need to create an instance
// same as - Console.WriteLine(Math.Sqrt(144));
```

```
class Program
{
    static void Main(string[] args)
    {
        Usefultools.SayHi("Ram");
        Console.ReadLine();
    }
}

class Usefultools
{
    public static void SayHi(string name)
    {
        Console.WriteLine("Hello, " + name + ".");
    }
}
```



```
static class UsefulTools
{
    public static void SayH
    {
```

```
UsefulTools tools = new UsefulTools();
UsefulTools.SayHi("Mike");
```

class Giraffe.UsefulTools

Cannot create an instance of the static class 'UsefulTools'

Inheritance

```
class Program
{

    static void Main(string[] args)
    {

        Chef chef = new Chef();
        chef.MakeSpecialDish();

        ItalianChef italianChef = new ItalianChef();
        italianChef.MakeSpecialDish();

        Console.ReadLine();
    }
}
```

```
class Chef
{
    public void MakeChicken()
    {
        Console.WriteLine("The Chef makes chicken");
    }

    public void MakeSalad()
    {
        Console.WriteLine("The Chef makes salad");
    }

    public virtual void MakeSpecialDish()
    {
        Console.WriteLine("The Chef makes bbq ribs");
    }
}
```

```
class ItalianChef : Chef
{
    public override void MakeSpecialDish()
    {
        Console.WriteLine("The Chef makes chicken parm");
    }

    public void MakePasta()
    {
        Console.Write("The Chef makes pasta");
    }
}
```