**Ref Keyword:**

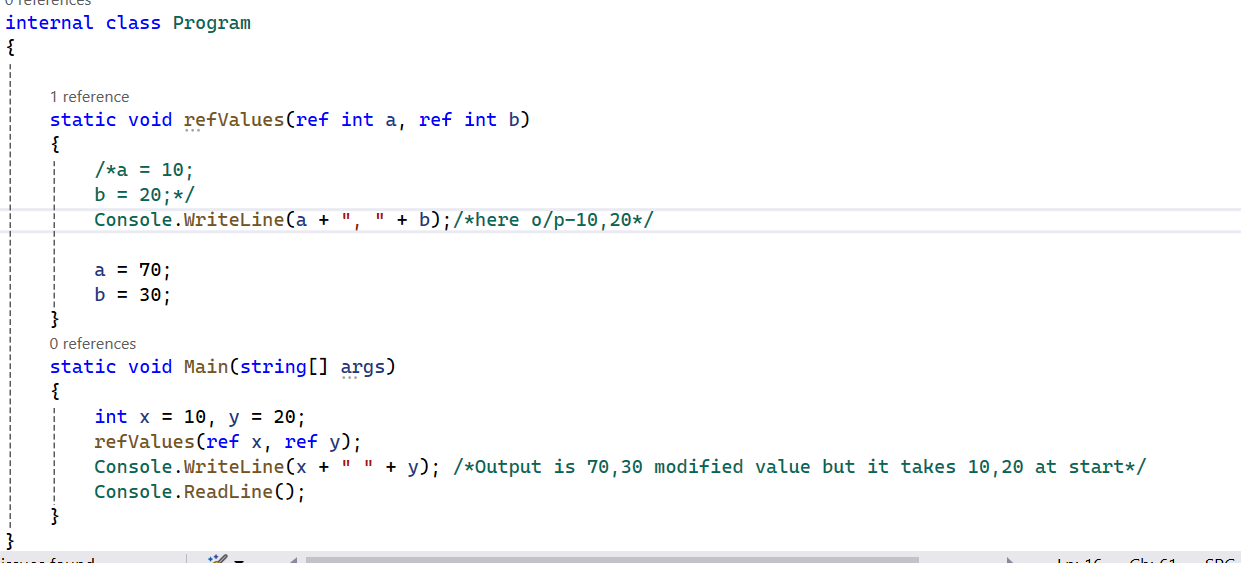
Used for Both Input and Output:

ref is used to indicate that a parameter is passed by reference.

It means that the method can modify the value of the parameter, and changes made inside the method will be reflected outside the method.

Initialization Required:

When using ref, the variable must be initialized before passing it to the method (opposite to out functionality).



**Out keyword:**

Primarily Used for Output:

out is used to indicate that a parameter is an output parameter.

It means that the method is expected to assign a value to the parameter, and the initial value of the parameter is not considered.

A computer screen shot of a person

Description automatically generated

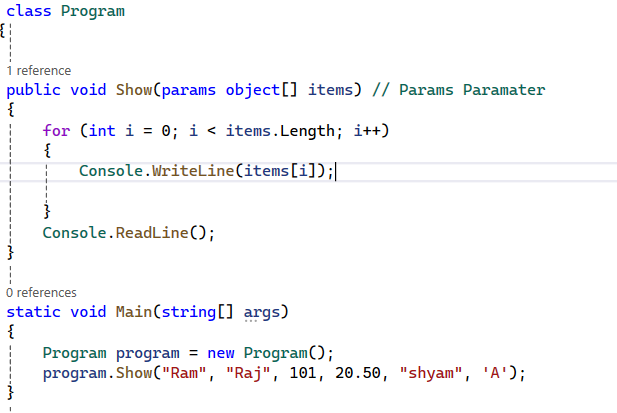
**Partial keyword:**

The partial keyword in C# is a way to split the definition of a class, methods ,struct, or interface into multiple files, if there is any additional info we can add to other classes,methods,etc ,and they all must be in same namespace bcz to identify they are on same entity making it easier to manage and organize code, especially in scenarios involving code generation or collaboration among multiple developers.

//doubt

**Params parameter:**

Params is a keyword which is used to specify a parameter that takes variable number of arguments. It is useful when we don't know the number of arguments prior. Only one params keyword is allowed and no additional parameter is permitted after params keyword in a function declaration.



**Named Parameters:**

Named parameters allow you to specify the values for parameters by providing the parameter names along with the values, rather than relying on the order of parameters in the method signature.

Advantages: This can enhance code readability, especially when a method has many parameters.

Optional parameters allow you to specify default values for certain parameters in a method. If a value is not provided when calling the method, the default value is used.

Advantages:It is used when there is a large no.of parameters we can make some default

A screenshot of a computer program

Description automatically generated

**Iterators &Yield keyword:**

Iterators are implemented using the yield keyword. They allow you to create a sequence of values without having to generate the entire sequence at once. An iterator allows you to iterate over a collection of data one item at a time without loading the entire collection into memory.

are commonly used with foreach loops to iterate over a sequence of values .

Iterators are useful when dealing with large datasets or when you want to generate values on-the-fly without consuming excessive memory.

Yield Keyword is used mainly in collections.

**Async &Await**

Allows you to write asynchronous code more easily and efficiently.Asynchronous means it allows the program to perform multiple tasks concurrently without waiting for each one to complete before moving on to the next.

Asynchronous programming is used in tasks which take much time to complete like the I/0 operations, network requests, etc.

The async keyword is used to declare a method as asynchronous.

It indicates that the method contains an asynchronous operation, and it can be paused and resumed.

public async Task MyAsyncMethod()

{

// Asynchronous code here

}

An asynchronous method typically returns a Task or Task<T>.

Task : ongoing operation

T : represents an operation that produces a result of type <T>

The await keyword is applied to an asynchronous operation (typically a method returning Task or Task<T>

It is used inside an asynchronous method to pause the execution until the awaited task is finished.

The expression after await must be a task or a task-like object.

