C# Variables with Examples

In c#, **Variables** will represent storage locations, and each variable has a particular type that determines what type of values can be stored in the variable.

C# is a **Strongly Typed** programming language. Before we perform any operation on variables, it's mandatory to define a variable with the required data type (/tutorial/csharp/csharp-data-types-with-examples) to indicate what type of data that variable can hold in our application.

Syntax of C# Variables Declaration

Following is the syntax of declaring and initializing variables in the c# programming language.

```
[Data Type] [Variable Name];
[Data Type] [Variable Name] = [Value];
[Access Specifier] [Data Type] [Variable Name] = [Value];
```

If you observe the above syntax, we added a required data type before the variable name to tell the compiler about what type of data the variable can hold or which data type the variable belongs to.

- **[Data Type]** - It's a type of data the variable can hold, such as integer, string, decimal, etc.
- **[Variable Name]** - It's the name of the variable to hold the values in our application.
- **[Value]** - Assigning a required value to the variable.
- **[Access Specifier]** - It is used to define access permissions for the variable.

Now we will see how to define variables in our c# applications with examples.

C# Variables Declaration Example

Following is the example of using the variables in the c# programming language.

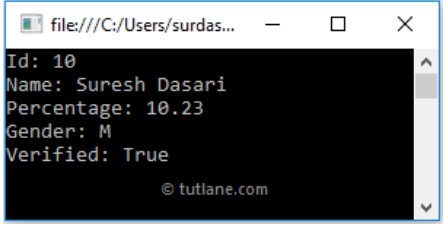
```
using System;

namespace Tutlane
{
      class Program
      {
           static void Main(string[] args)
           {
                int number = 10;
                string name = "Suresh Dasari";
                double percentage = 10.23;
                char gender = 'M';
                bool isVerified = true;
                Console.WriteLine("Id: " + number);
                Console.WriteLine("Name: " + name);
                Console.WriteLine("Percentage: " + percentage); Console.WriteLine("Gender: " + gender);
                Console.WriteLine("Verified: " + isVerified);
                Console.ReadLine();
           }
     }
}
```

If you observe the above c# variables example, we defined multiple variables with different data types and assigned values based on our requirements.

Output of C# Variables Declaration Example

When you execute the above program by pressing `Ctrl + F5` or clicking on the **Start** option in the menu bar, you will get the result shown below.



If you observe the above result, we are able to print the variables in our c# application based on our requirements.

Rules to Declare C# Variables

Before we declare and define variables in the c# programming language, we need to follow particular rules.

- You can define a variable name with a combination of alphabets, numbers, and underscore.
- A variable name must always start with either alphabet or underscore but not with numbers.
- While defining the variable, no white space is allowed within the variable name.
- Don't use any reserved keywords such as int, float, char, etc., for a variable name.
- In c#, once the variable is declared with a particular data type (/tutorial/csharp/csharp-data-types-with-examples), it cannot be re-declared with a new type, and we shouldn't assign a value that is not compatible with the declared type.

The following are some **valid** ways to define the variable names in the c# programming language.

```
int abc;
float a2b;
char _abc;
```

The following are some of the **Invalid** ways of defining the variable names in the c# programming language.

```
int a b c;
float 2abc;
char &abc;
double int;
```

C# Multiple and Multi-Line Variables Declaration

In c#, we can declare and initialize multiple variables of the same data type in a single line by separating with a comma.

Following is the example of defining the multiple variables of the same data type in a single line by separating with a comma in the c# programming language.

```
int a, b, c;
float x, y, z = 10.5;
```

While declaring the multiple variables of the same data type, we can arrange them in multiple lines to make them more readable. The compiler will treat it as a single statement until it encounters a **semicolon** (;).

Following is the simple of defining the multiple variables of the same data type in multiple lines in c# programming language.

```
int a,
    b,
    c;
float x,y,
      z = 10.5;
```

C# Variables Assignment

In c#, once we declare and assign a value to the variable that can be assigned to another variable of the same data type.

Following is the example of assigning a value of one variable to another variable of the same type in c# programming language.

```
int a = 123;
int b = a;
string name = "suresh";
string firstname = name;
```

In c#, it's mandatory to assign a value to the variable before we use it; otherwise, we will get a compile-time error.

If we try to assign a value of **string** data type to an **integer** data type or vice versa, as shown below, we will get an error like "**cannot implicitly convert type int to string**".

```
int a = 123;
string name = a;
```

This is how we can use variables in the c# programming language based on our requirements.

CONTACT US

Address: No.1-93, Pochamma Colony, Manikonda, Hyderabad, Telangana - 500089

Email: support@tutlane.com (mailto:support@tutlane.com)