

Difference Between Class and Structure in C#

www.differencebetween.com

Key Difference - Class vs Structure in C#

C# is a modern, general-purpose and [high-level programming language](#) developed by Microsoft. It is a programming language that has a similar resemblance to [Java](#). It contains strong programming features such as delegates, indexers and Language Integrated Query. The main advantage of C# is that it supports [Object Oriented Programming](#) (OOP). OOP is useful to model a program or a complex [software](#) using [objects](#). Objects are created using classes. The structure is also a concept in C#. In programming, a variable is a storage area that can store data. Each variable has a specific data type to store. Two data types are a value type and reference type. In value type, a value can be assigned directly. Some examples are int, char, float. Reference type data types do not store the actual data, but they contain a reference to the variable. Classes and Structures hold data. This article discusses the difference between Class and Structure in C#. The **key difference** between class and structure in C# is, **a class is a reference type data type while a structure is a value type data type.**

What is Class in C#?

In OOP, everything is considered as an object. In a program or software, there are any numbers of objects. These objects do not exist in isolation. Each object interacts with other objects. This communication occurs using actions. An object consists of attributes and methods. Attributes define the characteristics of the object and methods define the actions or behaviours of the object. Objects are created using a class. A class is similar to a description or a blueprint to create an object. The class has attributes and method that should be contained in the object. Creating an object using a class is known as instantiation. To create an object of type Student, there should be a class called Student. To define a class in C#, 'class' keyword is used.

A class contains a constructor. It is a special function that is executed whenever creating a new object of the class. A constructor has the same name as the class. It does not have a return type. The default constructor does not have any parameters. The parameterized constructor contains parameters. A class can also contain a destructor. It is a special function that executes when an object of the class goes out of scope. Destructor has the same name as the class name and starts with ~. It does not return any value and does not require parameters.

The class Student contains attributes and methods that should exist in the created objects.

```
using System;
```

```
namespace Application1{
```

```

class Student {
int studentID;
string name;
public Student (int sid, string sname) {
studentID = sid;
name = sname;
}
~Student();
public void displayDetails () {
Console.WriteLine (“Student ID is {0}”, studentID);
Console.WriteLine(“Student name is {0}”, name);
}
}
}

```

```

class StudentInformation {
static void Main (string [] args) {
Student s = new Student (1, “Ann”);
s.displayDetails();
Console.ReadLine();
}
}
}

```

According to the above program, the namespace is called Application 1. The namespace contains a related set of classes. Class Student has two properties student ID and name.

In the main program, Student object is created. That object is'. When creating the object, the constructor sets the values. Then, the displayDetails method is invoked. That will print student id and name.



Figure 01: C#

The classes can also implement [inheritance](#). Inheritance is a feature in Object Oriented Programming. It is to use the attributes and methods of an already existing class. Therefore, a class can inherit another class.

What is Structure in C#?

The structure is a value type data type in C#. It is a single variable that holds multiple data types. A Structure can represent a certain record. A structure contains attributes. A student has attributes such as id, name and age. A structure is defined using 'struct' keyword. A program with structure is as follows.

```
using System;
```

```
struct Student {  
  
public int id;  
  
public string name;  
  
};
```

```
public class Student () {  
  
public static void main (string [] args) {  
  
Student s;  
  
s.id = 1;  
  
s.name = "Ann";  
  
Console.WriteLine("Student id {0}", s.id);  
  
Console.WriteLine("Student name {0}", s.name);  
  
Console.ReadLine();  
  
}  
  
}
```

According to the above program, the structure contains two properties. They are id and name. In the main program, Student s statement declares s of type Student. Then, the id and name were given values. Finally, those are printed on the screen.

It is not possible to change the default constructor of a structure because the default constructor is automatically defined. The structure does not contain destructors. Unlike in class, a structure cannot inherit another class or another structure.

What are the Similarities Between Class and Structure in C#?

- Both have attributes and methods.
- Both Class and Structure can implement one or more interfaces.

What is the Difference Between Class and Structure in C#?

Class vs Structure in C#	
A class is a blueprint that is used to create an object.	A structure is a value type data type that can hold related data of various data types.
Data Type	
A class is a reference type data type.	A structure is a value type data type.
Inheritance	
A Class can inherit from other classes or structures.	A Structure cannot inherit other classes or structures.
Instantiation	
A class instantiates an object using the new keyword.	A structure instantiates an object without using the new keyword.
Default Constructor	
Can change the default constructor of a class.	Cannot change the default constructor of a structure. The default constructor is automatically defined.
Destructor	
Class can have a destructor.	Structure does not have a destructor.

Summary - Class vs Structure in C#

C# is a programming language created by Microsoft. In programming, the data should be stored. Data is stored in memory locations, and they are known as a variable. Each variable has a specific type of data to be stored. There can be value type data types and reference type data types. Value type data types store the actual value. The reference type data types store a reference to the variable. Class and Structure have different data types. The difference between class and structure in C# is, the class is a reference type data type while a structure is a value type data type.

Reference:

- 1.tutorialspoint.com. "C# Classes." [The Point, Available here](#)
- 2.tutorialspoint.com. "C# Structures." [The Point, Available here](#)

Image Courtesy:

1.'Logo C Sharp'By [Microsoft](#), (Public Domain) via [Commons Wikimedia](#)

How to Cite this Article?

APA: Difference Between Class and Structure in C#.(2018 February 02). Retrieved (date), from <http://differencebetween.com/difference-between-class-and-vs-structure-in-c/>

MLA: "Difference Between Class and Structure in C#" Difference Between.Com. 02 February 2018. Web.

Chicago: "Difference Between Class and Structure in C#." Difference Between.Com. <http://differencebetween.com/difference-between-class-and-vs-structure-in-c/> accessed (accessed [date]).



Copyright © 2010-2018 Difference Between. All rights reserved