

C# Class

C# Class

A class is a blueprint that defines what properties and behavior, the objects of this class should have.

class keyword is used to declare a class.

Example – C# Class

In the following example, we defined a class named `Book` . Ofcourse there is another class named `Program` where our main function is. But, we will focus on the class `Book` and make our discussions on it.

Program.cs

```
using System;

namespace CSharpExamples {
    class Book{
        public string author;
        public string title;

        public Book(string Title, string Author){
            title = Title;
            author = Author;
        }

        public void printBookDetails(){
            Console.WriteLine("\n---Book Details---\n=====");
            Console.WriteLine("Title : "+title);
            Console.WriteLine("Author : "+author);
        }
    }
    class Program {
        static void Main(string[] args) {
            Book book1 = new Book("C# Tutorial", "TutorialKart");
            book1.printBookDetails();

            Book book2 = new Book("Java Tutorial", "Author1");
            book1.printBookDetails();
        }
    }
}
```

Output

```
---Book Details---
=====
Title : C# Tutorial
Author : TutorialKart

---Book Details---
=====
Title : C# Tutorial
Author : TutorialKart
```

Class

We used `class` keyword followed by class name `Book` . Then the definition of the class is enclosed between the curly braces.

Properties

We declared two public variables: `title` and `author` of datatype string. These are called **properties** of class.

Constructor

`public Book(string Title, string Author){}` is the constructor in which we assigned the arguments to the class properties. You can also have an empty constructor with no arguments passed.

Methods

We defined one method named `printBookDetails()` with access modifier `public` . The method accepts no arguments, hence empty parenthesis. And returns nothing, hence `void` .

Create Class Object

We can create a new class object using `new` keyword.

In the above example, we created a new object of class type `Book`, using the following line of code.

```
Book book1 = new Book("C# Tutorial", "TutorialKart");
```

`Book book1` declares that `book1` is of type `Book` . `new Book()` creates a new object and assigns the object to `book1` . `Book("C# Tutorial", "TutorialKart")` calls the constructor with the arguments of type `(string, string)` .

Call a method on the Class Object

Call a method on the Class Object

To call any method of the class object, you can use the class object reference, dot operator and then the method name with arguments if any.

```
book1.printBookDetails();
```

Information about a C# class

- A class can be defined inside another class.
- A class can have an access modifier.
- A class can have multiple properties of different access types and datatypes.
- A class can have multiple methods that return or not return any values and accept or do not accept arguments.

C# Tutorial

- ◆ [C# Tutorial](#)
- ◆ [C# Basic Example](#)
- ◆ [C# Comments](#)
- ◆ [C# Variables](#)
- ◆ [C# Constants](#)
- ◆ [C# if, if-else](#)
- ◆ [C# switch](#)
- ◆ [C# while loop](#)
- ◆ [C# for loop](#)
- ◆ [C# foreach](#)
- ◆ [C# break](#)
- ◆ [C# continue](#)
- ◆ [C# struct](#)
- ◆ [C# enum](#)
- ◆ [C# String](#)
- ◆ [C# Array](#)
- ◆ [C# Command Line Arguments](#)

C# Console Operations

◆ C# Write to Console

◆ C# Read from Console

C# Object Oriented Programming Concepts

⇒ C# Class & Constructors

◆ C# Encapsulation

◆ C# Polymorphism

◆ C# Method Overloading

◆ C# Interfaces

C# String Operations

◆ C# String Length

◆ C# Substring

C# File Operations

◆ C# Read Text File

◆ C# Write to File

◆ C# Delete File

◆ C# Copy File

C# Exception Handling

◆ C# try-catch

◆ C# finally

◆ C# throw

◆ C# Custom Exception

◆ C# SystemException

◆ C# DivideByZeroException

◆ C# NullReferenceException

◆ C# InvalidCastException

◆ C# IOException

◆ C# FieldAccessException

C# Collections

◆ C# List

◆ C# SortedList

◆ C# HashSet

◆ C# SortedSet

◆ C# Stack

◆ C# Queue

◆ C# LinkedList

◆ C# Dictionary

◆ C# SortedDictionary

C# Errors [Solved]

◆ C# Error: Class does not contain a constructor that takes arguments