Structs

This lessons introduced structs and explains how to declare and use them in C#

WE'LL COVER THE FOLLOWING

Structs VS Classes

Structs are similar to *classes* but have subtle differences.

They are used as lightweight versions of *classes* that can help reduce *memory management* efforts when working with small data structures. In most situations, however, using a standard class is a better choice.

> The principal difference between **structs** and **classes** is that *instances* of **structs** are values whereas *instances* of **classes** are *references*.

Thus when you pass a **struct** to a *function* by **value** you get a *copy* of the *object*, so changes to it are not reflected in the original because there are now **two** distinct *objects* but if you pass an *instance* of a *class* by **reference** then there is only **one** *instance*.

Declaring a struct

Structures, or **structs**, are *defined* with

- `struct` *keyword*
- followed by an *identifier* to *name* the *structure*.

```
//Example 1
public struct Vector
{
 public int X;
 public int Y;
 public int Z;
}

//Example 2
public struct Point
{
 public decimal x, y;

 public Point(decimal pointX, decimal pointY)
 {
   x = pointX;
   y = pointY;
 }
}
```

Remarks

- `struct` *instance* fields can be set via a *parametrized constructor* or individually after *struct* construction.

> **Note:** A `struct` cannot declare a *parameterless constructor*.

- **Private** *members* can only be *initialized* by the *constructor*.
- **Structs** cannot *inherit* from any other type, but they can implement interfaces.
- A `struct` cannot be `null`, although it can be used as a **nullable** type.
- **Structs** can be instantiated **with** or **without** using the `new` operator.

```
//Both of these are acceptable
//with new operator
Vector v1 = new Vector();
v1.X = 1;
v1.Y = 2;
v1.Z = 3;

//without new operator
Vector v2;
```

```
v2.X = 1;
v2.Y = 2;
v2.Z = 3;
```

Example

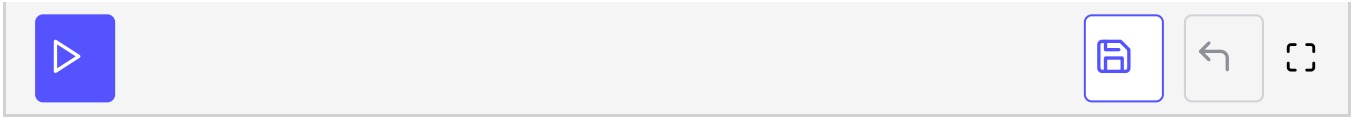
Let's take a look at an example implementing the use of `struct`.

With Constructor

Let's start by looking at the example of `struct` using *constructor*.

```
using System;

public struct Vector //declaring struct called Vector
{
  //no constructor defined
  //three public varibles defined
  public int X;
  public int Y;
  public int Z;
}

public struct Point //declaring struct called Point
{
  public double x, y;

  public Point(double pointX, double pointY) //constructor with parameters defined
  {
   x = pointX;
   y = pointY;
 }
}

class StructExample1
{
    static void Main()
    {
      Vector v1 = new Vector(); //making a struct insatnce v1 using new operator
      //setting values of variables defined in struct
      v1.X = 1;
      v1.Y = 2;
      v1.Z = 3;

      // Output X=1,Y=2,Z=3
      Console.WriteLine("X = {0}, Y = {1}, Z = {2}",v1.X,v1.Y,v1.Z);

      Vector v2 = new Vector();

      //v1.X is not assigned
      v2.Y = 2;
      v2.Z = 3;

      // Output X=0,Y=2,Z=3
      Console.WriteLine("X = {0}, Y = {1}, Z = {2}",v2.X,v2.Y,v2.Z);

      Point point1 = new Point();
```

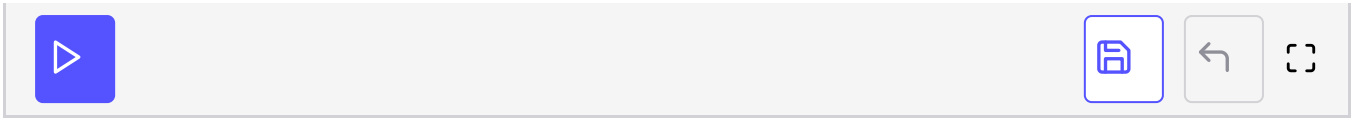
```
      point1.x = 0.5;
      point1.y = 0.6;
      Console.WriteLine("X = {0}, Y = {1}",point1.x,point1.y);

      //making instance of Point using constructor with parameter
      Point point2 = new Point(0.5, 0.6);
      Console.WriteLine("x = {0}, y = {1}",point2.x,point2.y);
    }
}
```



Without Constructor

Now let's look at the example for implementing `struct` without *constructor*.

```
using System;

public struct Vector //declaring struct called Vector
{
  //no constructor defined
  //three public varibles defined
  public int X;
  public int Y;
  public int Z;
}

public struct Point //declaring struct called Point
{
  public double x, y;

  public Point(double pointX, double pointY) //constructor with parameters defined
  {
    x = pointX;
    y = pointY;
  }
}

class StructExample2 {
  static void Main() {

    Vector v1; //declaring an instance of struct Vector without a constructor
    //setting values of variables in the struct instance v1
    v1.X = 5;
    v1.Y = 2;
    v1.Z = 3;

    Console.WriteLine("X = {0}, Y = {1}, Z = {2}",v1.X,v1.Y, v1.Z);

    Point point1; //declaring an instance of struct Point without a constructor
    //setting values of variables in the struct instance point1
    point1.x = 0.5;
    point1.y = 0.6;
    Console.WriteLine("x = {0}, y = {1}",point1.x,point1.y);
  }
}
```



This marks the end of our chapter. In the next chapter, we will discuss *inheritance* in *classes*. Read on to find out more!