



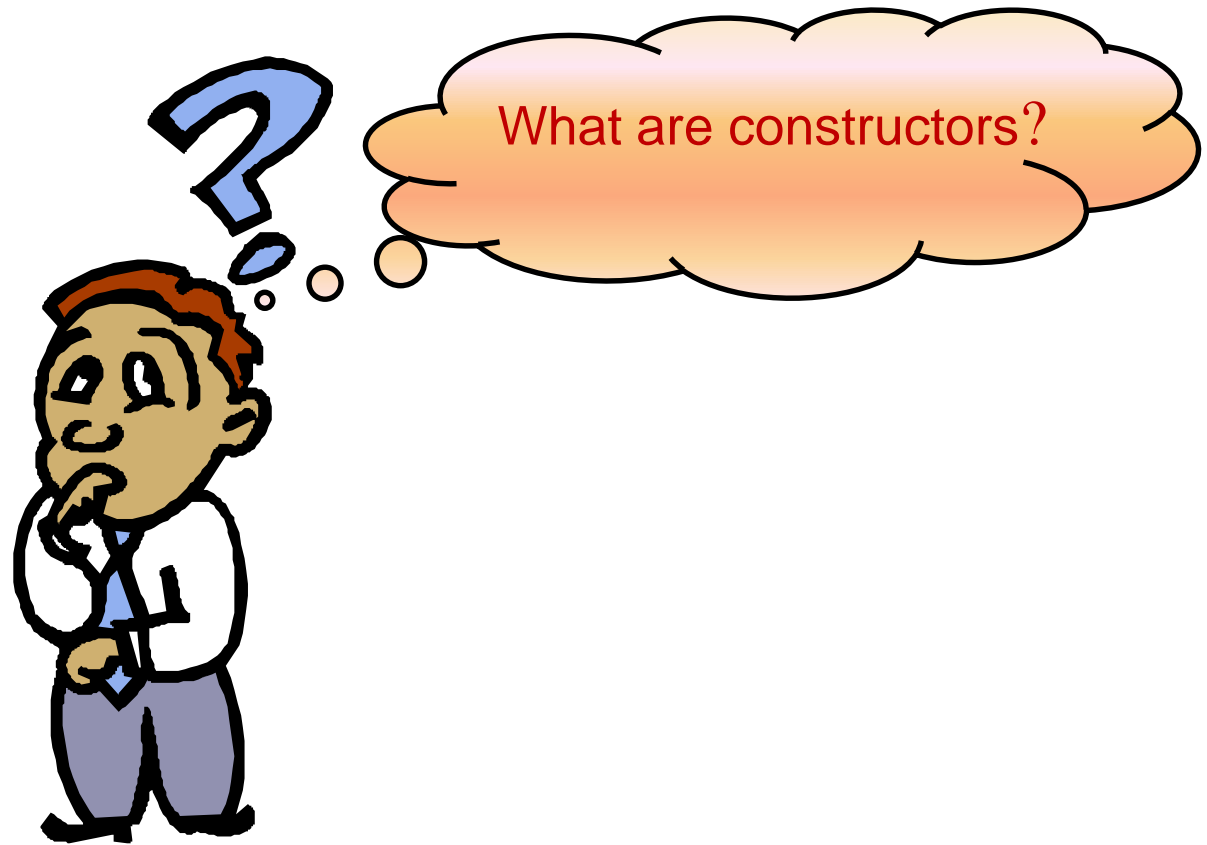
# Working With Constructor and Static Members

## Objectives

---

- ◆ In this session, you will learn to:
- ◆ Implement constructors

# Implementing Constructors



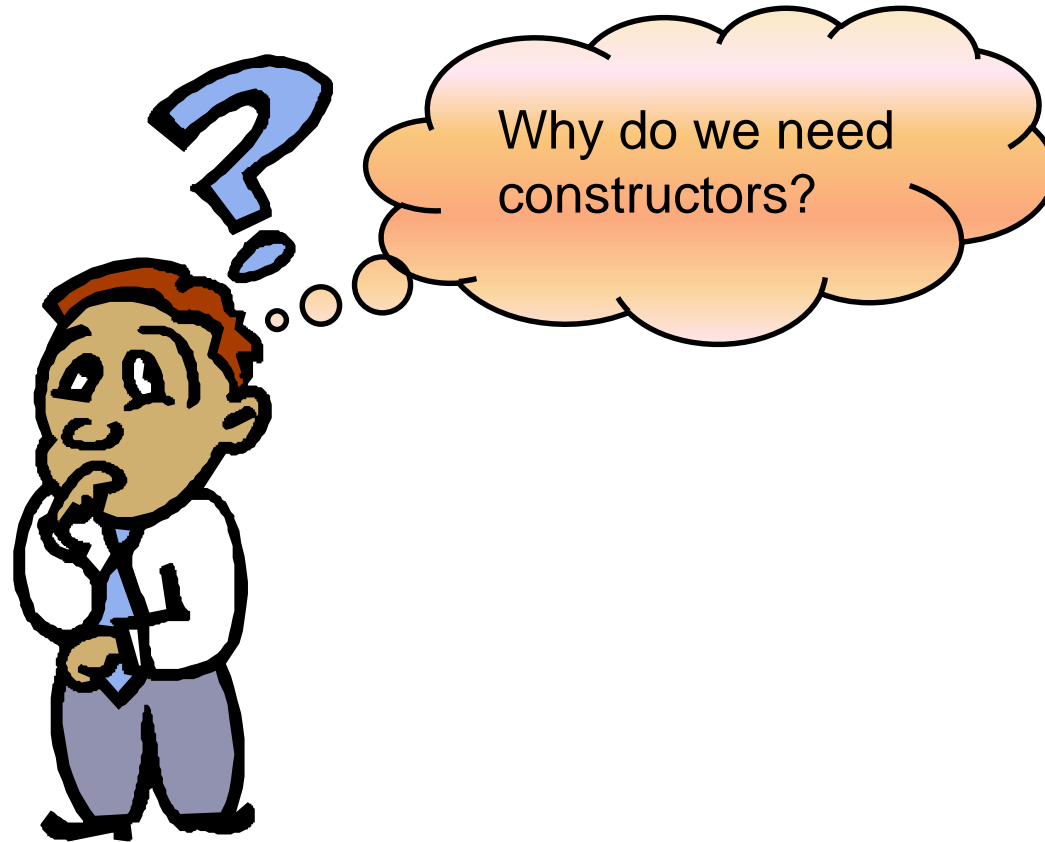
## Implementing Constructors (Contd.)

---

- ◆ A constructor is a special type of method that is invoked automatically when you create an instance of a class.
- ◆ A constructor is used to initialize the members of the class.
- ◆ The name of a constructor is the same as the name of the class that contains it.
- ◆ Always have the same name as class.
- ◆ Don't returns a value(Even void is not allowed in its declaration).
- ◆ Invoked Automatically, When object or Instance of a class is created.
- ◆ Can make a call to other methods.

# The Need of Constructors

---



## The Need of Constructors (Contd.)

---

- ◆ A constructor can be used to initialize member variables when an object of the class is created.
- ◆ A constructor gets invoked only once during the execution of a program.
- ◆ The following example shows the use of constructor in a program:

```
public class Class1
{
    int number1;
    int number2;
    Class1()    // Constructors have the same name as
                //the class
    {
        number1 = 10;
                number2 = 3;    }    }
```

## Constructor Example

```
class Customer{  
    String custName ;  
    int custID ;  
    public Customer(int custName, string custID)  
    {  
        this.custName = custName;  
        this.custID = custID;  
    }  
    static void Main(string[] args)  
    {  
        Customer cust = new  
        Customer("Tom", "C001" );  
    }  
}
```

Invoked  
automatically  
when object  
or instance is  
created

Used to indicates the  
current type instance.  
Here, we are referring  
class instance filed  
through it.

# Default Constructor

- Compiler creates the default constructor ,if no constructors created by programmer.
  - Default constructor created by compiler is public parameterless instance constructor.
  - Has empty body
  - Calls only parameterless constructor from the base class.

```
public class  
Customer  
{  
int age;  
string city;  
}
```

Is equivalent  
to:

```
public class Customer  
{  
int age;  
string city;  
public Customer() :  
base()  
{  
}
```




# Types of Constructors

---

- ◆ The two types of constructors supported by C# are:
- ◆ Instance constructors: They are called whenever an instance of a class is created. These constructors are used to initialize the data members of the class.
- ◆ Static constructors: They are used to initialize the static variables of a class. These variables are created using static keyword and they store values that can be shared by all the instances of a class.

## Types of Constructors (Contd.)

---

A small blue diamond icon with a white outline.

Let us understand the two types of constructors with the help of various programs.

## Types of Constructors (Contd.)

---

- ◆ The following code shows the use of an instance constructor:

```
using System;
namespace Calc
{
    class Calculator
    {
        static int number1, number2, total;
        Calculator()
        {
            number1 = 10;
            number2 = 20;
        }
    }
}
```

## Types of Constructors (Contd.)

---

```
        public void AddNumber()
        {
            total = number1 + number2;
        }
    public void DisplayNumber()
    {
        Console.WriteLine("The Total is :{0}",          total);
    }

    public static void Main(string[] args)
    {
        Calculator c1 = new Calculator ();
        c1.AddNumber ();
        c1.DisplayNumber ();
    }
}
```

## Types of Constructors (Contd.)

---

- ◆ The following code shows the use of a static constructor:

```
public class Class1
{
    static int number1;
    int number2;
    static Class1()
    {
        number1 = 10; /*OK. Since Number1 is a static
        variable in class1 */
        number2 = 3; /*Error. Since Number2 is a
        non-static variable in class1 */
    }
}
```

# Constructors with Parameters

---

- ◆ Constructors can also initialize the variables of the program with the values by the user.
- ◆ The following code shows the use of constructor with parameters:

```
using System;
namespace Calc
{
    class Calculator
    {
        static int number1, number2, total;
        //Constructor defined with two integer parameters
        Calculator(int num1, int num2)
        {
```

## Constructors with Parameters (Contd.)

---

```
        number1 = num1;
        number2 = num2;
    }
    public void AddNumber()
    {
        total = number1 + number2;
    }
    public void DisplayNumber()
    {
        Console.WriteLine("The Total is :{0}", total);
    }
    public static void Main(string[] args)
    {
        int var1, var2;
```

## Constructors with Parameters (Contd.)

---

```
        Console.WriteLine("Enter Value 1:");  
        var1=Convert.ToInt16(Console.ReadLine());  
        Console.WriteLine("Enter Value 2  :");  
        var2 = Convert.ToInt16(Console.ReadLine());  
        Calculator C1 = new Calculator(var1,var2);  
        C1.AddNumber();  
        C1.DisplayNumber();  
        Console.ReadLine();  
    }  
}  
}
```



# Summary

---

- ◆ In this lesson, you learned that:
  - ◆ Constructors are member functions of a class and are invoked when an instance of the class to which they belong is created.
  - ◆ A constructor has the same name as its class.



Thank You