**Classes and objects**

















For example :

 int Findsum ( int num1 , int num2 )

 {

 int sum = num1 + num2;

 return sum ;

 }

 Here, we defined a method named Findsum which takes two parameters of int type (num1 and num2) and returns a value of type int using the keyword return. If a method does not return anything, its return type would be void. A method can also optionally take no parameter ( a parameterless method ).









 







**Constructors**

A *constructor* initializes an object when it is created. It has the same name as its class and is syntactically similar to a method. However, constructors have no explicit return type. The general form of a constructor is shown here:

*access class*-*name*(*param-list*) {

// constructor code

}

Typically, you will use a constructor to give initial values to the instance variables defined by the class or to perform any other startup procedures required to create a fully formed object. Also, usually, *access* is **public** because constructors are normally called from outside their class. The *param-list* can be empty, or it can specify one or more parameters.

All classes have constructors, whether you define one or not, because C# automatically provides a default constructor that causes all member variables to be initialized to their default values. For most value types, the default value is zero. For **bool**, the default is **false**.

For reference types, the default is null. However, once you define your own constructor, the default constructor is no longer used.

Here is a simple example that uses a constructor:

**Ex : Design and implement a class to find the area of Circle?**

class Circle

{

 private double radius;

 public Circle(double r) // constructor

 {

 radius = r;

 }

 public double Area()

 {

 return radius \* radius \* 3.14;

 }

 }

class Program

{

 static void Main(string[] args)

 {

 Circle obj1 = new Circle(3.5);

 Circle obj2 = new Circle(10);

 Console.WriteLine("the area of first circle is " + obj1.Area());

 Console.WriteLine("the area of second circle is " + obj2.Area());

 Console.ReadKey();

 }

}

**Ex2 Design and implement a class to read two integer numbers and find the largest number**

**namespace CollectionsApplication**

**{**

 **class MaxNumber**

 **{**

 **private int number1;**

 **private int number2;**

 **public MaxNumber(int num1, int num2) // constructor**

 **{**

 **number1 = num1;**

 **number2 = num2;**

 **}**

 **public int FindMax()**

 **{**

 **if (number1 > number2)**

 **return number1;**

 **else**

 **return number2;**

 **}**

 **}**

 **class Test**

 **{**

 **static void Main(string[] args)**

 **{**

 **MaxNumber Max1 = new MaxNumber(50, 100);**

 **MaxNumber Max2 = new MaxNumber(80, 30);**

 **Console.WriteLine("Max value is : " + Max1.FindMax());**

 **Console.WriteLine("Max value is : " + Max2.FindMax());**

 **Console.ReadLine();**

 **}**

 **}**

**}**

**Example 3: Design and implement a class to create array table of 9 real numbers , and search the value ele in array and print index of numbers?**

**namespace CollectionsApplication**

**{**

 **class Search1**

 **{**

 **private double[] table;**

 **private double element;**

 **private int size;**

 **public Search1(int n, double ele) // constructor**

 **{**

 **table = new double[n];**

 **element = ele;**

 **size = n;**

 **}**

 **public void ReadArr()**

 **{**

 **Console.WriteLine("input the " + size + "elements ");**

 **for (int i = 0; i < size; i++)**

 **{**

 **Console.WriteLine("input element " + i);**

 **table[i] = double.Parse(Console.ReadLine());**

 **}**

 **}**

 **public void searchele()**

 **{**

 **int i;**

 **for (i = 0; i < size; i++)**

 **if (element == table[i]) break;**

 **if (i == size)**

 **Console.WriteLine("Value : " + element + "Not Found.");**

 **else**

 **Console.WriteLine("Value : " + element + " Order :" + i);**

 **}**

 **static void Main(string[] args)**

 **{**

 **Search1 obj1 = new Search1(10, 3.5);**

 **Search1 obj2 = new Search1(5, 10);**

 **obj1.ReadArr();**

 **obj1.searchele();**

 **obj2.ReadArr();**

 **obj2.searchele();**

 **Console.ReadKey();**

 **}**

 **}**

**}**