

Visual Basic Constructors

In visual basic, Constructor is a method and it will invoke automatically whenever an instance of class or struct is created. The constructor is useful to initialize and set default values for the data members of the new object.

In case, if we create a class without having any constructor, the compiler will automatically create a one default constructor for that class. So, there is always one constructor that will exist in every class.

In visual basic, a class can contain more than one constructor with a different type of arguments and the constructors will never return anything, so we don't need to use any return type, not even void while defining the constructor method in the class.

Visual Basic Constructor Syntax

As discussed, the constructor is a method and it won't contain any return type. If we want to create a constructor in visual basic, we need to create a method with New keyword.

Following is the syntax of creating a constructor in visual basic programming language.

```
Public Class User
```

```
    ' Constructor
```

```
    Public Sub New()
```

```
        ' Your Custom Code
```

```
    End Sub
```

```
End Class
```

If you observe the above syntax, we created a class called "User" and a method with New keyword. Here the method New() will become a constructor of our class.

Visual Basic Constructor Types

In visual basic, we have a different type of constructors available, those are

Default Constructor

Parameterized Constructor

Copy Constructor

Private Constructor

Now, we will learn about each constructor in detail with examples in a visual basic programming language.

Visual Basic Default Constructor

In visual basic, if we create a constructor without having any parameters, we will call it as default constructor and the instance of the class will be initialized without any parameter values.

Following is the example of defining the default constructor in visual basic programming language.

Module Module1

Class User

Public name, location As String

' Default Constructor

Public Sub New()

name = "Vinay Kumar"

location = "Jeypore"

End Sub

End Class

Sub Main()

' The constructor will be called automatically once the instance of the class created

Dim user As User = New User()

Console.WriteLine(user.name)

Console.WriteLine(user.location)

Console.WriteLine("Press Enter Key to Exit..")

Console.ReadLine()

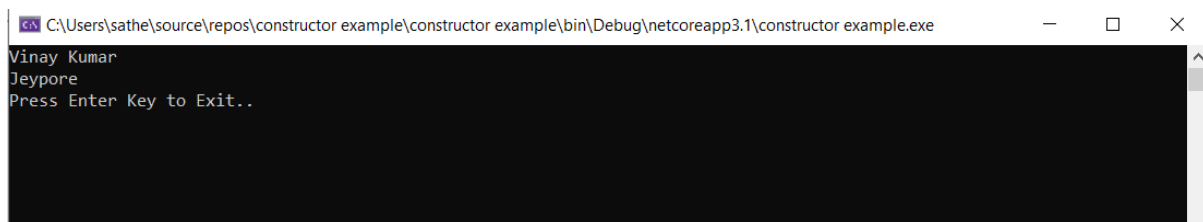
End Sub

End Module

If you observe the above example, we created a class called "User" and the constructor method "New()" without having any parameters. When we create an instance of our class (User), then automatically our constructor method will be called.

When we execute the above visual basic program, we will get the result as shown below.

Visual Basic Default Constructor Example Result



If you observe the above result, our constructor method has called automatically and initialized the parameter values after creating an instance of our class.

Visual Basic Parameterized Constructor

In visual basic, if we create a constructor with at least one parameter, we will call it a parameterized constructor and every time the instance of the class must be initialized with parameter values.

Following is the example of defining the parameterized constructor in a visual basic programming language.

Module Module1

Class User

Public name, location As String

' Parameterized Constructor

Public Sub New(ByVal a As String, ByVal b As String)

name = a

location = b

End Sub

End Class

Sub Main()

' The constructor will be called automatically once the instance of the class created

Dim user As User = New User("Nainika", "Sunabeda")

Console.WriteLine(user.name)

Console.WriteLine(user.location)

Console.WriteLine("Press Enter Key to Exit..")

Console.ReadLine()

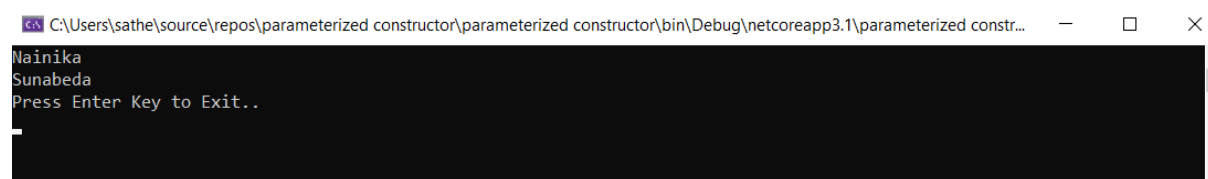
End Sub

End Module

If you observe the above example, we created a class called "User" and the constructor method "New(string, string)" with parameters. When we create an instance of our class (User) with required parameters, then automatically our constructor method will be called.

When we execute the above visual basic program, we will get the result as shown below.

Visual Basic Parameterized Constructor Example Result



```
C:\Users\sathe\source\repos\parameterized constructor\parameterized constructor\bin\Debug\netcoreapp3.1\parameterized constr...  
Nainika  
Sunabeda  
Press Enter Key to Exit..  

```

If you observe the above result, our constructor method has called automatically and initialized the parameter values after creating an instance of our class with the required parameters.

Visual Basic Constructor Overloading

In visual basic, we can overload the constructor by creating another constructor with the same method name but with different parameters.

Following is the example of implementing a constructor overloading in a visual basic programming language.

Module Module1

Class User

Public name, location As String

' Default Constructor

Public Sub New()

name = "Saheli"

location = "Bhubaneswar"

End Sub

' Parameterized Constructor

Public Sub New(ByVal a As String, ByVal b As String)

name = a

location = b

End Sub

End Class

Sub Main()

' Default Constructor will be called

Dim user As User = New User()

' Parameterized Constructor will be called

Dim user1 As User = New User("Anchal", "Bhubaneswar")

Console.WriteLine(user.name & ", " & user.location)

Console.WriteLine(user1.name & ", " & user1.location)

Console.WriteLine(vbLf & "Press Enter Key to Exit..")

Console.ReadLine()

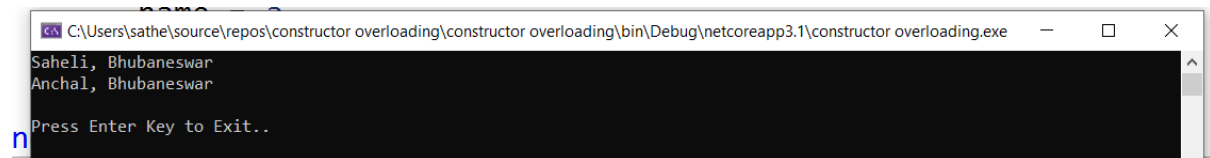
End Sub

End Module

If you observe the above example, we created a class called “User” and overloaded the constructor “New()” by creating another constructor “New(string, string)” with same name but with different parameters.

When we execute the above visual basic program, we will get the result as shown below.

Visual Basic Constructor Overloading Example Result



If you observe the above result, the respective constructor methods have been called automatically when we created an instance of our class with or without parameters based on our requirements.

Visual Basic Constructor Chaining

In visual basic, Constructor Chaining is an approach to invoke one constructor from another constructor. To achieve constructor chaining we need to use Me keyword after our constructor definition.

Following is the example of implementing a constructor chaining in a visual basic programming language.

Module Module1

Class User

Public Sub New()

Console.Write("Hi, ")

End Sub

Public Sub New(ByVal a As String)

Me.New()

Console.Write(a)

End Sub

Public Sub New(ByVal a As String, ByVal b As String)

Me.New("welcome")

Console.Write(a & " " & b)

End Sub

End Class

Sub Main()

Dim user1 As User = New User(" to", "VB.Net")

```
Console.WriteLine()
```

```
Console.WriteLine("Press Enter Key to Exit..")
```

```
Console.ReadLine()
```

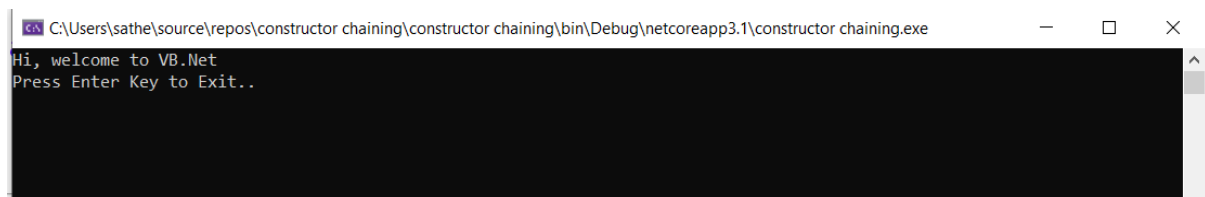
```
End Sub
```

```
End Module
```

If you observe the above example, we created a different constructor with different parameters and we are calling one constructor from another constructor using Me keyword.

When we execute the above visual basic program, we will get the result as shown below.

Visual Basic Constructor Chaining Example Result



If you observe the above result, we are able to call one constructor from another constructor to achieve constructor chaining in a visual basic programming language.

This is how we can achieve constructor chaining in our applications using a visual basic programming language.