

What is Generics?

Creating collection by using generics helps you understand the best use of generics. For example, you need a collection class that you can use to create type safe collection of the Teacher and Student objects. In other words Teacher collection should consist of objects of only the Teacher class, Student collection should consist of objects of only the student class.

Note: The .Net Framework 4.0 provide number of collection classes in the `System.Collection.Generics` and `System.Collections.ObjectModel` namespace

Implementation of Generics

The following code example shows you, how to implement generics in C#. The code example defines a generics class called `CommenData`, which is used to create two objects, one to store the string value and one to store the float value. The `CommenData` class ensure the Type-Sefty by accepting the required type in its constructor.

Code:

```
using System;

using System.Collections.Generic;

class Program
{
    public static void Main()
    {
        CommonData<string> name = new CommonData<string>();
        name.Value = ".Net FrameWork 4.0";

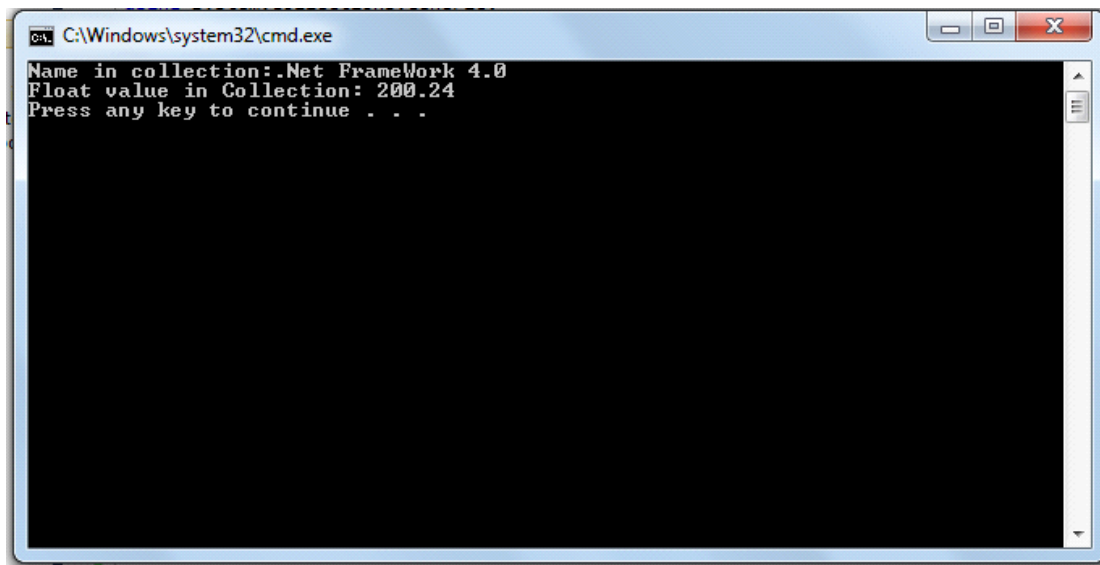
        CommonData<float> version = new CommonData<float>();
        version.Value = 200.24f;

        Console.WriteLine("Name in collection:" + name.Value);

        Console.WriteLine("Float value in Collection: " + version.Value);
    }
}
```

```
public class CommonData<T>
{
    private T _data;
    public T Value
    {
        get
        {
            return this._data;
        }
        set
        {
            this._data = value;
        }
    }
}
```

OutPut:



```
C:\Windows\system32\cmd.exe
Name in collection:.Net Framework 4.0
Float value in Collection: 200.24
Press any key to continue . . .
```

Advantage of Generics

- Reusability: A single generics type definition can be used for multiple scenarios in the same code, without any alterations.
- Type Safety: Generics data type provide better type safety, especially in situations where collection are used. In collection class, when you add objects, the compiler does not check the type of the object you are adding. As a result cast at runtime may fail.
- Performance: Generics type perform better than normal system types because they reduce the need for boxing , unboxing and the type casting of the objects and variable.

Prashant Nimbare

Faculty

NIIT