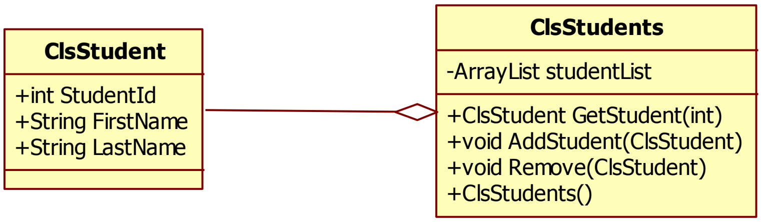
Context:

This document contains assignments to be completed as part of the hands-on for the Collections day2 topic.

Assignment 1: Understanding Custom collection

Estimated time: 25 Mins

Problem description: Create custom collection class ClsStudents as given in figure below:



Create an object of ClsStudents in main method and use the GetStudent, AddStudent, Remove methods of the class.

Hint: Define arraylist variable in ClsStudents. Methods 'GetStudent', 'AddStudent' and 'Remove' will use the arraylist variable for the required operation

Assignment 2: Understanding IEumerator

Estimated time: 25 Mins

Problem description: Consider below code snippet for a class:

```
class ClsCustomCollection
    {
        Arraylist items;
        int indexNumber;

        public ClsCustomCollection()
        {
            items = new ArrayList();
            indexNumber= -1;
        }

        public void AddItem(object itemToAdd)
        {
            Items.Add(itemToAdd);
        }
    }
```

Below code provides the use of the class:

```
ClsCustomCollection myCustomClass = new ClsCustomCollection();
myCustomClass.AddItem(1);
myCustomClass.AddItem(2);
myCustomClass.AddItem(3);
myCustomClass.AddItem(4);
```

Can you modify the CustomCollectionClass to provide below methods/properties:

- Public bool MoveNext():The method increments the integer 'indexNumber'
- public void Reset(): The method sets the value of indexNumber to -1
- Public object Current{get;} The property returns current object from the items array. Position of current object can be determined by variable 'indexNumber'

The usability of the class after providing the new methods/properties is given below:

```
ClsCustomCollection myCustomClass = new ClsCustomCollection();
myCustomClass.AddItem(1);
myCustomClass.AddItem(2);
myCustomClass.AddItem(3);
myCustomClass.AddItem(4);

while(myCustomClass.MoveNext())
{
       Console.WriteLine(myCustomClass.Current.ToString());
}

Console.ReadKey();
```

Can you name the interface that CustomCollectionClass is implementing in that case. The interface is called as ‘IEnumerator’ (System.Collection namespace) and you can define your CustomCollectionClass as

‘**CustomCollectionClass:IEnumerator**’

Right click in IEnumerator keyword(in your code) and click on “Go to definition” to confirm the methods to be implemented for the interface

Assignment 3: Understanding yield keyword

Estimated time: 15 Mins

Problem description: What is the value that the below method always returns?

```
public static String GetValues()
        {
            return "100";
            return "1001";
            return "1002";
            return "1003";
            return "1004";
        }
```

Can you call the method as given below so that all the return values from function are displayed on console?

```
foreach(object s in GetValues())
           {
               Console.Write(s.ToString());
           }
```

What modifications do you need to make in the method to get the required objective.

Hint: use yield keyword

Assignment 3: Understanding IEnumerable

Estimated time: 25 Mins

Problem description: Consider below code for a class:

```
class CustomArrayList
    {
        Arraylist items;
        int iteratorCount;

        public CustomArrayList()
        {
            items = new Arraylist();
            iteratorCount = -1;
        }

        public void AddItem(Object itemToAdd)
        {
            Items.Add(itemToAdd);
        }

    }
```

The class is used as below code shows:

```
static void Main(string[] args)
        {
            CustomArrayList myCustomArray = new CustomArrayList();
            myCustomArray.AddItem("1");
            myCustomArray.AddItem("2");
            myCustomArray.AddItem("3");
            myCustomArray.AddItem("4");
        }
```

Is it possible to use ‘foreach’ loop (foreach(object o in myCustomArray)) to traverse through the items in myCustomArray? If not, what is the error message you get?

How can CustomArrayList be modified so that foreach loop can be applied on the object of the class?

Hint: As the error message shows, you need to provide a method GetEnumerator returning an IEnumerator i.e. you need to implement IEnumerable interface. The method will use ‘yield’ keyword to return objects from the Arraylist variable

Assignment 4: Understanding indexers

Estimated time: 25 Mins

Problem description: Refer UML diagram of assignment1.

Given a custom collection class ‘ClsStudents’. Till now we saw that we can provide “AddStudent” method in the class to add students to the custom collection object.

Can we do something as shown in snippets below to add objects to the custom collection?

```
ClsStudents objStudents = new ClsStudents();
objStudents[0] = objStudent1;
objStudents[1] = objStudent2;
```

Hint: An indexer allows an object to be indexed like an array. Syntax for providing an indexer in a class is as follows:

```
public Student this[int index]
    {
      get
      {
        return students[index] ;
      }
      set
      {
        students.Add(value);
      }
    }
```

Assignment 5: Need for generics

Estimated time: 30 Mins

Problem description: Analyze the below code:

```
Person obj1 = new Person();
obj1.Name = "John";

Person obj2 = new Person();
obj2.Name = "Bill";

Person obj3 = new Person();
obj3.Name = "Dell";

Person obj4 = new Person();
obj4.Name = "Misha";

ArrayList list = new ArrayList();
list.Add(obj1);
list.Add(obj2);
list.Add(obj3);
list.Add(obj4);
list.Add("123");

foreach (Person p in list)
{
    Console.WriteLine(p.Name);
}
Console.ReadKey();
```

The code throws some runtime error. Can you identify the error? What modifications you need to make in the code to avoid the error?

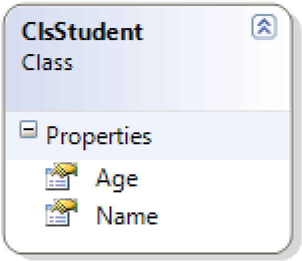
Hint: Use type checking for the object in the list

Assignment 6: Working with generic lists

Estimated time: 10 Mins

Problem description: Create an object of generic list class (System.Collections.Generic.List)

Create some objects of ClsStudent class (given below) and add them to the list. Iterate through the list to display student names on console.



Can you add an object of type other than ClsStudent (say string) to the specified list?

Assignment 7: Working with dictionary

Estimated time: 10 Mins

Problem description: Create an object of generic dictionary class.

Create objects of ClsStudent class with various names say “John”, “Bill”,”Meeta”,”Jolly”,”Bill”.

Add the objects to the dictionary using student name as key. Do you get any error while adding objects with above names to the dictionary? How can you rectify the error?

How can you get the student object with name “Meeta” without traversing through the dictionary?

Summary of assignments:

You have learnt

- How to create custom collection class
- How to implement IEnumerator
- Use of Yield keyword
- How to implement IEnumerable and use it
- How to apply indexer to a custom collection class
- Need for Generics
- Working with generic list collection
- Working with generic dictionary collection