

# TAAZAA TRAINING

## ASSIGNMENT- 5

**SUBMITTED BY- DAS SUKHDEV**

### 1. COLLECTION WITH PROPER NAMING CONVENTION

country.cs

```
namespace collection_demo.Model
{
    public class Country //pascal case
    {
        public string currency{get;set;}//camel case
        public string cname{get;set;}//camel case
    }
}
```

CountryDetails.cs

```
using collection_demo.Model;
using System.Collections;
using System.Collections.Generic;
namespace collection_demo.Proper_Model
{
    public class CountryDetails//pascal case
    {
        public ArrayList countryDetails1()//camel case
        {
            ArrayList arrayList = new ArrayList();//pascal case
            arrayList.Add("Ruppes");
            arrayList.Add("India");
            arrayList.Add("America");
            arrayList.Add("Dollar");
            arrayList.Add("Cameroon");
            arrayList.Add("Franc");
            return arrayList;
        }
        public List<Country> CountryDetails2()
        {
            List<Country> obj = new List<Country>();
            obj.Add(new Country
            {
```

```

        currency = "Dollar",
        cname = "Australia"
    });
    obj.Add(new Country
    {
        currency = "Euro",
        cname = "Austria"
    });
    obj.Add(new Country
    {
        currency = "Euro",
        cname = "Belgium"
    });
    return obj;
}
}
}

```

## Program.cs

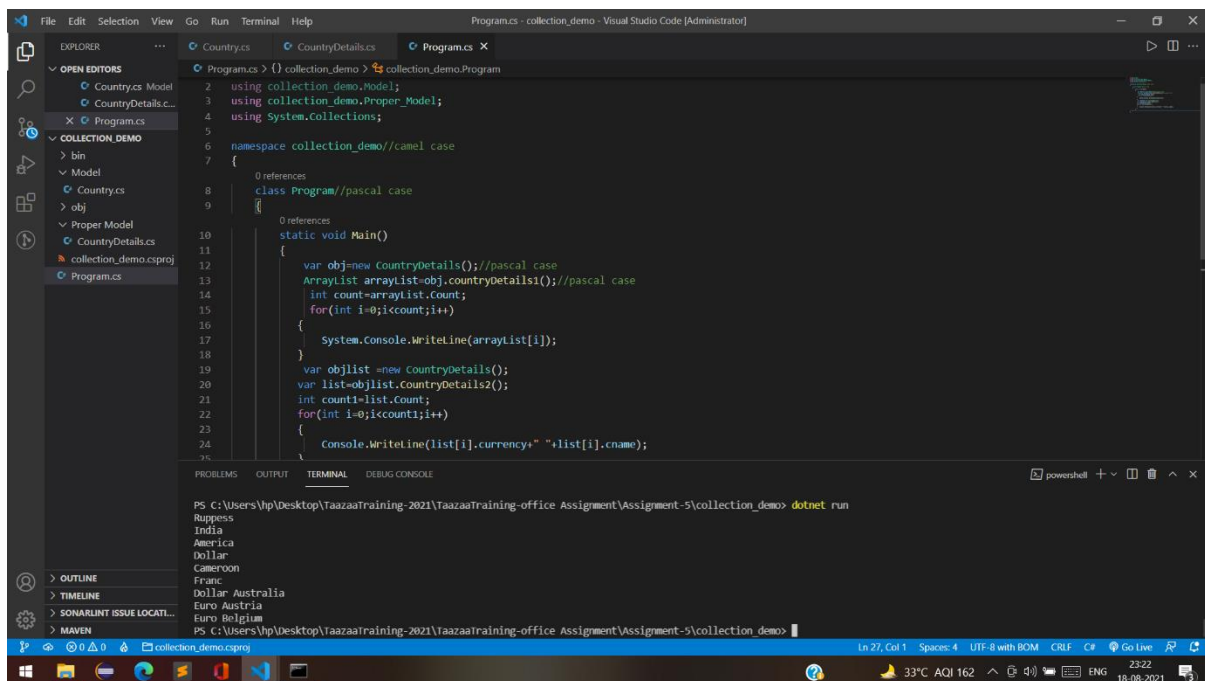
```

using System;
using collection_demo.Model;
using collection_demo.Proper_Model;
using System.Collections;

namespace collection_demo//camel case
{
    class Program//pascal case
    {
        static void Main()
        {
            var obj=new CountryDetails();//pascal case
            ArrayList arrayList=obj.countryDetails1();//pascal case
            int count=arrayList.Count;
            for(int i=0;i<count;i++)
            {
                System.Console.WriteLine(arrayList[i]);
            }
            var objlist =new CountryDetails();
            var list=objlist.CountryDetails2();
            int count1=list.Count;
            for(int i=0;i<count1;i++)
            {
                Console.WriteLine(list[i].currency+" "+list[i].cname);
            }
        }
    }
}

```

## Output of collections



```
File Edit Selection View Go Run Terminal Help
Program.cs - collection_demo - Visual Studio Code [Administrator]

EXPLORER
OPEN EDITORS
Country.cs Model
CountryDetails.c...
Program.cs
COLLECTION_DEMO
bin
Model
Country.cs
obj
Proper Model
CountryDetails.cs
collection_demo.csproj
Program.cs

Program.cs > {} collection_demo > collection_demo.Program
2 using collection_demo.Model;
3 using collection_demo.Proper_Model;
4 using System.Collections;
5
6 namespace collection_demo//camel case
7 {
8     0 references
9     class Program//pascal case
10    {
11        0 references
12        static void Main()
13        {
14            var obj=new CountryDetails();//pascal case
15            ArrayList arrayList=obj.countryDetails1();//pascal case
16            int count=arrayList.Count;
17            for(int i=0;i<count;i++)
18            {
19                System.Console.WriteLine(arrayList[i]);
20            }
21            var objlist =new CountryDetails();
22            var list=objlist.countryDetails2();
23            int count1=list.Count;
24            for(int i=0;i<count1;i++)
25            {
26                Console.WriteLine(list[i].currency+" "+list[i].cname);
27            }
28        }
29    }
30 }

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Users\hvp\Desktop\Taazaatrainig-2021\Taazaatrainig-office Assignment\Assignment-5\collection_demo> dotnet run
Russia
India
America
Dollar
Cameroon
Franc
Dollar Austria
Euro Belgium
PS C:\Users\hvp\Desktop\Taazaatrainig-2021\Taazaatrainig-office Assignment\Assignment-5\collection_demo>
```

## 2 Private Constructor

/\* when a constructor is created with private keyword in a class, and we can't create child class of that class neither it is possible to create instance of that class. They are used in classes that contain static members only. \*/

```
using System;
namespace private_constructor
{
    class Example
    {
        static public int i;
        private Example()
        {}
        public static void getTimeDetails()
        {
            System.Console.WriteLine(DateTime.Now);
        }
        public static int update(int x)
        {
            return ++x;
        }
    }
    /* class Example2:Example
    {
```

```

        error
    } */
class Program
{
    static void Main(string[] args)
    {
        //var obj=new Example(); error
        Example.getTimeDetails();
        var val=Example.update(20);
        System.Console.WriteLine("Incremented Value of I is "+val);

    }
}

```

## Output of private constructor

```

File Edit Selection View Go Run Terminal Help
Program.cs - private_constructor - Visual Studio Code [Administrator]

EXPLORER
  OPEN EDITORS
    Program.cs
  PRIVATE CONSTRUCTOR
    bin
    obj
    private_constructor.cs
    Program.cs

Program.cs
  9      static public int i;
  10     private Example()
  11     {
  12     }
  13     public static void getTimeDetails()
  14     {
  15         System.Console.WriteLine(DateTime.Now);
  16     }
  17     public static int update(int x)
  18     {
  19         return ++x;
  20     }
  21     /* class Example2:Example
  22     {
  23         error
  24     } */
  25     class Program
  26     {
  27     static void Main(string[] args)
  28     {
  29         //var obj=new Example(); error
  30         Example.getTimeDetails();
  31         var val=Example.update(20);
  32         System.Console.WriteLine("Incremented Value of I is "+val);
  33     }
  34 }
  35

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
18-08-2021 23:46:06
Incremented Value of I is 21
PS c:\Users\Up\Desktop\Taazaatraining-2021\Taazaatraining-office Assignment\Assignment-5\private_constructor>

```

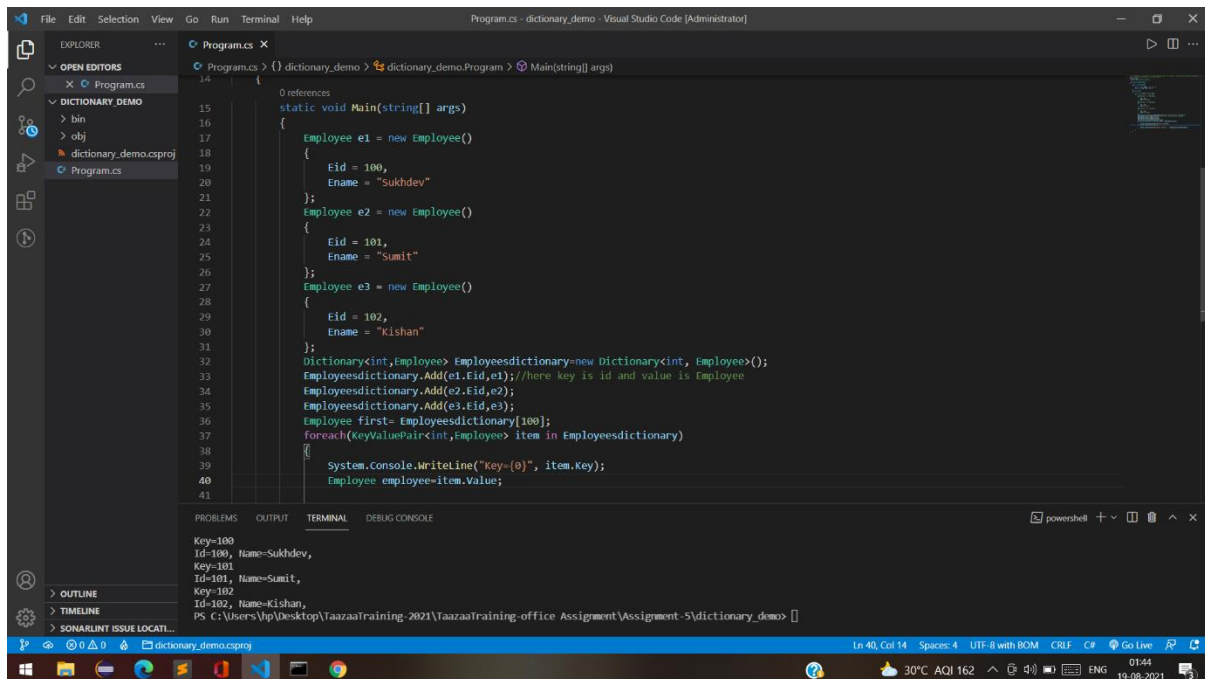
### 3 Dictionary

```
/* A dictionary is a collection of (key,value)pairs and it is present in (system.
collection.Genericnamespace),
here we need to specify types of key and value. */
using System;
using System.Collections.Generic;

namespace dictionary_demo
{
    public class Employee
    {
        public string Ename { get; set; }
        public int Eid { get; set; }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Employee e1 = new Employee()
            {
                Eid = 100,
                Ename = "Sukhdev"
            };
            Employee e2 = new Employee()
            {
                Eid = 101,
                Ename = "Sumit"
            };
            Employee e3 = new Employee()
            {
                Eid = 102,
                Ename = "Kishan"
            };
            Dictionary<int,Employee> Employeesdictionary=new Dictionary<int, Employee>();
            Employeesdictionary.Add(e1.Eid,e1);//here key is id and value is Employee
            Employeesdictionary.Add(e2.Eid,e2);
            Employeesdictionary.Add(e3.Eid,e3);
            Employee first= Employeesdictionary[100];
            foreach(KeyValuePair<int,Employee> item in Employeesdictionary)
            {
                System.Console.WriteLine("Key={0}", item.Key);
                Employee employee=item.Value;

                System.Console.WriteLine("Id={0}, Name={1}, ", employee.Eid,employee.Ename);
            }
        }
    }
}
```

## Output of dictionary program



The screenshot shows the Visual Studio Code interface with a C# program in the editor and its output in the terminal. The program creates a dictionary of employees and prints the first employee's details.

```
14  {
15      static void Main(string[] args)
16      {
17          Employee e1 = new Employee()
18          {
19              Eid = 100,
20              Ename = "Sukhdev"
21          };
22          Employee e2 = new Employee()
23          {
24              Eid = 101,
25              Ename = "Sumit"
26          };
27          Employee e3 = new Employee()
28          {
29              Eid = 102,
30              Ename = "Kishan"
31          };
32          Dictionary<int, Employee> Employeesdictionary = new Dictionary<int, Employee>();
33          Employeesdictionary.Add(e1.Eid, e1); //here key is id and value is Employee
34          Employeesdictionary.Add(e2.Eid, e2);
35          Employeesdictionary.Add(e3.Eid, e3);
36          Employee first = Employeesdictionary[100];
37          foreach (KeyValuePair<int, Employee> item in Employeesdictionary)
38          {
39              System.Console.WriteLine("Key-{0}", item.Key);
40              Employee employee = item.Value;
41          }
42      }
43  }
```

OUTPUT

```
Key=100
Id=100, Name=Sukhdev,
Key=101
Id=101, Name=Sumit,
Key=102
Id=102, Name=Kishan,
PS C:\Users\hp\Desktop\TaazaaTraining-2021\TaazaaTraining-office Assignment\Assignment-5\dictionary_demo>
```

## 4 HashTable

```
/* Hash Table is basically a collection of key-
value pairs where key is an indexer and value is accessed by the key and it im-
plements IDictionary interface*/

using System;
using System.Collections;
namespace hash_demo
{
    class Program
    {
        static void Main(string[] args)
        {

            Hashtable hashtable = new Hashtable();
            //Inserting
            hashtable.Add(1, "Computer science");
            hashtable.Add(2, "Mechanical");
            hashtable.Add(3, "Civil");
            hashtable.Add(4, "Electricals");
            //deletion
            hashtable.Remove(4);
            //search by value()
            System.Console.WriteLine(hashtable.ContainsKey(4)); //return true
            or false based on value
        }
    }
}
```

```

        //search by key
        System.Console.WriteLine(hashtable.ContainsValue("Civil"));
        System.Console.WriteLine("Current value at index 2 is "+hashtable[
2]);

        //updation of value by indexes
        hashtable[2]="Aeronautical Engineering";
        System.Console.WriteLine("value after updation at index 2 is "+has
htable[2]);

        //Accessing elements
        foreach (DictionaryEntry item in hashtable)
        {
            System.Console.WriteLine("value of Key "+item.Key+" is "+item.V
alue);
        }
    }
}
}
}

```

## Output of HashTable Program

The screenshot shows the Visual Studio Code interface with a C# program named `Program.cs` open. The code defines a `Program` class with a `Main` method that manipulates a `Hashtable`. The terminal output at the bottom shows the execution results.

```

6  {
7      0 references
8      class Program
9      {
10         0 references
11         static void Main(string[] args)
12         {
13             Hashtable hashtable=new Hashtable();
14             //Inserting
15             hashtable.Add(1,"Computer science");
16             hashtable.Add(2,"Mechanical");
17             hashtable.Add(3,"Civil");
18             hashtable.Add(4,"Electricals");
19             //deletion
20             hashtable.Remove(4);
21             //search by value()
22             System.Console.WriteLine(hashtable.ContainsKey(4)); //return true or false based on value
23             //search by key
24             System.Console.WriteLine(hashtable.ContainsValue("Civil"));
25             System.Console.WriteLine("Current value at index 2 is "+hashtable[2]);
26             //updation of value by indexes
27             hashtable[2]="Aeronautical Engineering";
28             System.Console.WriteLine("value after updation at index 2 is "+hashtable[2]);
29             //Accessing elements
30             foreach (DictionaryEntry item in hashtable)
31             {
32                 System.Console.WriteLine("value of Key "+item.Key+" is "+item.Value);
33             }
34         }
35     }
36 }

```

Terminal Output:

```

False
True
Current value at index 2 is Mechanical
value after updation at index 2 is Aeronautical Engineering
value of Key 3 is Civil
value of Key 2 is Aeronautical Engineering
value of Key 1 is Computer science
PS C:\Users\Ship\Desktop\Taazaatraining-2021\Taazaatraining-office Assignment\Assignment-5\hash_demo>

```