**Important topic**

var pass = new char[10];

            var random = new Random();

            for (int i = 0; i < 10; i++)

            {

                pass[i]=(char)('a'+random.Next(0,26));

                Console.WriteLine(pass);

                var password= new string(pass);

                Console.WriteLine(pass);

            }

Code 02

var datetime = new DateTime(2015, 1, 1);

             var currenttimr=DateTime.Now;

           //Console.WriteLine(datetime);

           //Console.WriteLine(currenttimr);

           //Console.WriteLine("hour" + currenttimr.Hour);

           //var tomarrow= currenttimr.AddDays(1);

           //var yestarday= currenttimr.AddDays(-1);

           //Console.WriteLine(tomarrow);

           //Console.WriteLine(yestarday);

           var timespan = new TimeSpan(1, 5, 8);

               Console.WriteLine(timespan);

Console.WriteLine("Enter the number ");

           var lst = new List<int>();

           var str= Console.ReadLine().Trim().Split('-');

           foreach (var item in str)

           {

               lst.Add(int.Parse(item));

           }

           for (int i = 0; i < lst.Count; i++)

           {

               if (lst[i] == lst[i+1] )

               {

                   Console.WriteLine("Consecutive");

                   break;

               }

               else Console.WriteLine("non Consecutive");

Console.WriteLine("Enter the english wourd ");

           var count = 0;

           var list=new List<char>();

           var inp= Console.ReadLine().ToLower();

           foreach (var item in inp)

           {

               if (list.Contains('a') && list.Contains('e') && list.Contains('u'))

                   {

                   list.Add(item); }

               //Console.WriteLine(item);

           }

                   Console.WriteLine(list.Count);

Uses’ of substring

var str = "ankur mall is living in mumbie ";

            var empty = " ";

            var str2= str.Split(' ');

            foreach (var c in str2)

            {

                Console.WriteLine(c.Substring(1));

            }

OPPES IN C#

using System.Collections.Generic;

namespace ConsoleApp1

{

    public class customer

    {

        public string Name { get; set; }

        public int Id { get; set; }

        public List<string> orders;

        public customer()

        {

            orders = new List<string>();

        }

        public customer(int id )

            : this()

        {

                this.Id = id;

        }

        public customer(string name , int id)

        {

            this.Name = name;

            this.Id = id;

        }

    }

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

    internal partial class Program

    {

        static void Main(string[] args)

        {

            var custom = new customer("ankur" , 3250120);

            Console.WriteLine(custom.Id);

            Console.WriteLine(custom.Name);

            var odr = new order();

            Console.WriteLine(DateTime.Now);

            Console.ReadLine();

        }

    }

}

namespace ConsoleApp1

{

    internal partial class Program

    {

        public class point

        {

           public  int x;

            public int y;

            public point(int x, int y)

            {

                this.x = x;

                this.y = y;

            }

            public void Move(int x , int y)

            {

                this.x=x;

                this.y=y;

            }

            public void Move(point newLoaction)

            {

                this.x=newLoaction.x;

                this.y=newLoaction.y;

            }

        }

    }

}

var point = new point(10, 20);

            point.Move(new point(40, 60));

            Console.WriteLine((point.x, point.y));

            Console.WriteLine(DateTime.Now);

            Console.ReadLine();

……………………………………………………………….

\*\*\* Using of parmas using of calculators class

namespace ConsoleApp1

{

    internal partial class Program

    {

        public class calculator

        {

            public int Add(params int[] numbers)

            {

                var sum = 0;

                foreach (var number in numbers)

                {

                    sum+=number;

                }

                return sum;

            }

        }

    }

}

var cal = new calculator();

            Console.WriteLine(cal.Add(1, 2,3,4,5,8,9,11,45));

            Console.WriteLine(DateTime.Now);

            Console.ReadLine();

* Main topic
* using System.Collections.Generic;
* namespace ConsoleApp1
* {
* public class customer
* {
* public string Name { get; set; }
* public int Id { get; set; }
* public List<string> lst = new List <string> ();



* public customer(string name , int id)
* {
* this.Name = name;
* this.Id = id;
* }
* public void prmoate()
* {
* var lst = new List<string>();
* }

* }
* }

using System.Collections.Generic;

namespace ConsoleApp1

{

    public class customer

    {

        public string Name { get; set; }

        public int Id { get; set; }

        public List<string> lst = new List <string> ();

        public customer(string name , int id)

        {

            this.Name = name;

            this.Id = id;

        }

        public void prmoate()

        {

            var lst = new List<string>();

        }

    }

}

Excesses Modifer

using System;

namespace ConsoleApp1

{

    internal partial class Program

    {

        public class person

        {

            private DateTime \_birthdate;

            public void setbirthdate(DateTime birthdate)

            {

                \_birthdate = birthdate;

            }

            public DateTime getbirtdate()

            {

                return \_birthdate;

            }

        }

    }

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

    internal  partial class Program

    {

        static void Main(string[] args)

        {

            var pers = new person();

            pers.setbirthdate(new DateTime(1997, 03, 08));

            Console.WriteLine(pers.getbirtdate());

            Console.WriteLine(DateTime.Now);

            Console.ReadLine();

        }

    }

}

Properties… get , set

using System;

namespace oppes

{

    public class person

    {

        public DateTime Birthdate { get; set; }

        public int age

        {

            get

            {

                var year = (DateTime.Now - Birthdate);

                Console.WriteLine(year);

                var cuurentage = year.Days / 365;

                Console.WriteLine(cuurentage);

                return cuurentage;

            }

        }

    }

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace oppes

{

    public  class Program

    {

        static void Main(string[] args)

        {

            var person=new person();

            person.Birthdate =new  DateTime(1997,03,08);

            Console.WriteLine(person.age);

            Console.ReadLine();

        }

    }

}

If somone change set to private set so you have used ctor and set the properties

using System;

namespace oppes

{

    public class person

    {

        public DateTime Birthdate { get;  private set; }

        public person(DateTime birthdate)

        {

            Birthdate = birthdate;

        }

        public int age

        {

            get

            {

                var year = (DateTime.Now - Birthdate);

                Console.WriteLine(year);

                var cuurentage = year.Days / 365;

                Console.WriteLine(cuurentage);

                return cuurentage;

            }

        }

    }

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace oppes

{

    public  class Program

    {

        static void Main(string[] args)

        {

            var person=new person(new DateTime(1997, 03, 08));

            Console.WriteLine(person.age);

            Console.ReadLine();

        }

    }

}

using System;

namespace oppes

{

    public class person

    {

        public string Name  { get; set; }

        public string  username  { get; set; }

        public DateTime Birthdate { get;  private set; }

        public person(DateTime birthdate)

        {

            Birthdate = birthdate;

        }

        public int age

        {

            get

            {

                var year = (DateTime.Now - Birthdate);

                Console.WriteLine(year);

                var cuurentage = year.Days / 365;

                Console.WriteLine(cuurentage);

                return cuurentage;

            }

        }

    }

}

Using prop + tab code snipe for this .

Indexer

using System.Collections.Generic;

using System.Runtime.Remoting.Messaging;

namespace oppes

{

    public class HttpCookie

    {

        private Dictionary<string, string> \_dictionary;

        public HttpCookie()

        {

            \_dictionary = new Dictionary<string, string>();

        }

        public string this[string key]

        {

            get { return \_dictionary[key]; }

            set

            {

                \_dictionary[key] = value;

            }

        }

    }

}

Main function

……………………………………………………………………………………………..

static void Main(string[] args)

        {

            var cookie = new HttpCookie();

            cookie["name"] = "mosh";

            Console.WriteLine(cookie["name"]);

            Console.ReadLine();

        }

Classes coupling

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace oppes

{

    public class PresntaionObject

    {

        public int Width { get; set; }

        public int height  { get; set; }

        public void Copy()

        {

            Console.WriteLine("object copyed to clipboard ");

        }

        public void Duplicate()

        {

            Console.WriteLine("object duplicate to clipboard ");

        }

    }

}

………………..

    using System;

    using System.Collections.Generic;

    using System.Linq;

    using System.Net;

    using System.Text;

    using System.Threading.Tasks;

namespace oppes

{

    public class Text : PresntaionObject////////////////////

    {

        public int FrontSize { get; set; }

        public string  FrontName  { get; set; }

        public void AddHyperlink(string url)

        {

            Console.WriteLine("we added link to url"+ url );

        }

    }

}

 public  class Program

    {

        static void Main(string[] args)

        {

            var txt = new Text();

            txt.Width = 100;

            Console.WriteLine("the width of sample is " +  txt.Width);

            txt.Copy();

            Console.ReadLine();

        }

    }

}

Compostion in C#

Make three class

Dbmigrater

Install

Logger

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace oppes

{

    public class Logger

    {

        public void Log(string message)

        {

            Console.WriteLine(message);

        }

    }

}

///////////////////////////

using System.Runtime.CompilerServices;

namespace oppes

{

    public class DbMigrator

    {

        private readonly Logger \_logger;

        public DbMigrator(Logger logger)

        {

            \_logger = logger;

        }

        public void Migrate()

        {

            \_logger.Log("We are migrating blah blah blha .........");

        }

    }

}

///////////////////////////////////////////////////////////

namespace oppes

{

    public class Installer

    {

        private readonly Logger \_logger;

        public Installer(Logger logger)

        {

            \_logger = logger;

        }

        public void Install()

        {

            \_logger.Log("er are installing the application ");

        }

    }

}

///////////////////////////

    static void Main(string[] args)

    {

        var dbmig= new DbMigrator(new Logger());

        var log= new Logger();

        var instal = new Installer(log);

        dbmig.Migrate();

        instal.Install();

        Console.ReadLine();

    }

}

Base classes constaructor called first

using System;

namespace oppes

{

    public class vehical

    {

        private readonly string \_registrationnumber;

        //public vehical()

        //{

        //    Console.WriteLine("hum first hum first");

        //}

        public vehical(string  registrationnumber)

        {

            \_registrationnumber = registrationnumber;

            Console.WriteLine(registrationnumber);

        }

    }

}

////////////////////////////////

using System;

namespace oppes

{

    public class car : vehical

    {

        public car( string registrationNumber)

            :base(registrationNumber)

        {

            Console.WriteLine(registrationNumber);

        }

    }

}

//////////////////////////

var cr = new car("xyz1234");

concept of upcasting and downcasting

namespace oppes

{

    public class Shape

    {

        public int width { get; set; }

        public int Height  { get; set; }

        public int X { get; set; }

        public int  Y  { get; set; }

        public void Draw()

        {

        }

    }

}

//////////////////////////

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace oppes

{

    public class Text : Shape

    {

        public int FrontSize { get; set; }

        public string  FrontName  { get; set; }

        public void AddHyperlink(string url)

        {

            Console.WriteLine("we added link to url"+ url );

        }

    }

}

////////////

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace oppes

{

    public  class Program

    {

        static void Main(string[] args)

        {

            var txt = new Text();

            Shape shape = txt;

            txt.width = 200;

            shape.width = 200;

            Console.WriteLine(txt.width);

            Console.WriteLine(shape.width);

            Console.ReadLine();

        }

    }

}

Using the value type to refrence type

var lst = new ArrayList();

           lst.Add(1);

           lst.Add("ankur");

           lst.Add(DateTime.Now);

           foreach (var item in lst)

           {

               Console.WriteLine(item);

           }

           Console.ReadLine();

       }

If we convert the(int ) lst[1] to but it throw exception

\*\*

**Polymorphism**

namespace oppes

{

    public class Shape

    {

        public int width { get; set; }

        public int Height { get; set; }

        public Position Position { get; set; }

        public ShapeType Type { get; set; }

        public virtual void Draw()

        {

        }

    }

}

////////////////////////////////////////////////////

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;

namespace oppes

{

    public  class Circle : Shape

    {

        public override void Draw()

        {

            Console.WriteLine("draw the ranctle cilcle");

        }

    }

}

/////////////////////

namespace oppes

{

    public class Rectangle : Shape

    {

        public override void Draw()

        {

            Console.WriteLine("Draw a rectangle");

        }

    }

}

////////////////////////using System.Collections.Generic;

using System;

using System.Collections;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace oppes

{

    public class Canvas

    {

        public void DrawShapes(List<Shape> shapes)

        {

            foreach (Shape shape in shapes)

            {

                shape.Draw();

            }

        }

                                                                                                                    //foreach (var item in shapes)

                                                                                                                    //{

                                                                                                                    //    switch (item.Type)

                                                                                                                    //    {

                                                                                                                    //        case ShapeType.Circle:

                                                                                                                    //            Console.WriteLine("Draw the Circle");

                                                                                                                    //            break;

                                                                                                                    //        case ShapeType.Rectangle:

                                                                                                                    //            Console.WriteLine("Draw the rectangle ");

                                                                                                                    //            break ;

                                                                                                                    //    }

                                                                                                                    //}

    }

}

**Abstract classes uses for the forcing something and**

**Problem 01**

Exercise 1: Design a Stopwatch Design a class called Stopwatch. The job of this class is to simulate a stopwatch. It should provide two methods: Start and Stop. We call the start method first, and the stop method next. Then we ask the stopwatch about the duration between start and stop. Duration should be a value in TimeSpan. Display the duration on the console. We should also be able to use a stopwatch multiple times. So we may start and stop it and then start and stop it again. Make sure the duration value each time is calculated properly. We should not be able to start a stopwatch twice in a row (because that may overwrite the initial start time). So the class should throw an InvalidOperationException if its started twice.

Solution

 public class Stopwatch

    {

        private DateTime \_startTime;

        private DateTime \_endTime;

        private bool \_isRunning;

        public void start()

        {

            if (\_isRunning)

            {

                throw new InvalidOperationException("stopwatch is already running");

            }

            else

            {

                \_startTime = DateTime.Now;

                \_isRunning=true;

            }

        }

        public void stop()

        {

            if (!\_isRunning)

                throw new InvalidOperationException("stopwatch is not running");

            {

                \_endTime= DateTime.Now;

                \_isRunning = false;

            }

        }

        public TimeSpan getIntrval()

        {

            var duration = \_endTime- \_startTime;

            return duration;

        }

    }

}

//////////////////////main function

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace oppes

{

    public  class Program

    {

        static void Main(string[] args)

        {

            var watch= new Stopwatch();

            for (int i = 0; i < 2; i++)

            {

                watch.start();

               Thread.Sleep(10000);

                watch.stop();

                Console.WriteLine(watch.getIntrval());

            }

            Console.ReadLine();

        }

    }

}

**Interfaces**

**Using the loss coupling**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace oppes

{

    public interface ILogger

    {

        void LogError(string message);

        void LogInfo(string message);

    }

}

////////////////////

using System;

using System.Runtime.CompilerServices;

namespace oppes

{

    public class DbMigrator

    {

       private readonly ILogger \_logger;

        public DbMigrator(ILogger logger)

        {

            \_logger = logger;

        }

        public void Migrate()

        {

            \_logger.LogInfo("migration started time is  ........." + DateTime.Now);

           // Console.WriteLine ("migration started time is  ........." + DateTime.Now);

            \_logger.LogInfo("migration finshied  time is  ........." + DateTime.Now);

            // Detils of migrating the database

            //Console.WriteLine ("migration finished is  ........." + DateTime.Now);

        }

    }

}///////////////////////////

 public class ConsoleLogger : ILogger

    {

        public void LogError(string message)

        {

            Console.ForegroundColor = ConsoleColor.Red;

            Console.WriteLine(message);

        }

        public void LogInfo(string message)

        {

            Console.ForegroundColor= ConsoleColor.Green;

            Console.WriteLine(message);

        }

    }

}//////////////////////////////////////\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace oppes

{

    public  class Program

    {

        static void Main(string[] args)

        {

            var dbm = new DbMigrator(new ConsoleLogger());

            dbm.Migrate();

            Console.ReadLine();

        }

    }

}

**Interface not kind of inharit it is used for implemented for the proprrty**

using System;

namespace oppes

{

    public class UiControl

    {

        public string Name { get; set; }

        public Size Size { get; set; }

        public Position TopLeft  { get; set; }

        public virtual void Draw()

        {

        }

        public void Focus()

        {

            Console.WriteLine("Received focus.");

        }

    }

}

//////////////////

using System;

namespace oppes

{

    public class TextBox : UiControl, IDraggable, IDroppable

    {

        public void Drag()

        {

            throw new NotImplementedException();

        }

        public void Drop()

        {

            throw new NotImplementedException();

        }

    }

}

    public interface IDraggable

    {

        void Drag();

    }

}

    public interface IDroppable

    {

        void Drop();

    }

}

public class Message

   {

   }

////

public interface INotificationChannel

   {

       void Send (Message message);

   }

///////////

public  class Video

    {

        public Video()

        {

        }

    }

///////////////

 public class MailNotificationChannel : INotificationChannel

    {

        public void Send (Message message)

        {

            Console.WriteLine( "Sending mail...." );

        }

    }

///////////////

}

namespace ServerChat\_piolet\_projet

{

    public class SmsNotificationChannel : INotificationChannel

    {

         public void Send(Message message)

        {

            Console.WriteLine("Sending SMS..............");

        }

    }

}

//////////////////////

**// Deligete is the kind of object of how to call of the group of method**

using System;

namespace extra\_advance\_topic\_of\_csharp

{

    public class Photo

    {

        public static Photo Load(String path)

        {

            return new Photo();

        }

        public void save()

        {

        }

    }

}/////////////////

    using System;

namespace extra\_advance\_topic\_of\_csharp

{

    public  class PhotoFilter

    {

        public void ApplyBrightness(Photo photo)

        {

            Console.WriteLine("Aply brightness");

        }

        public void ApplyConstrast( Photo photo)

        {

        }

        public void Resize(Photo photo)

        {

        }

    }

}

//////////////////

using System;

namespace extra\_advance\_topic\_of\_csharp

{

    public class Photoprocesser

    {

        public void Process(string path)

        {

            var photo = Photo.Load(path);

            var filter = new PhotoFilter();

            filter.ApplyBrightness(photo);

            filter.ApplyConstrast(photo);

            filter.Resize(photo);

            photo.save();

        }

    }

}

var processer= new Photoprocesser();

           processer.Process("photo.jpg");

           Console.ReadLine();

**Lambda function**

using System;

using System.Collections.Generic;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace extra\_advance\_topic\_of\_csharp

{

    public class Program

    {

        static void Main(string[] args)

        {

            Func<int, int> square = n => n \* n;

             Console.WriteLine(square(5));

            Console.ReadLine();

        }

    }

    //    static public  int Square(int number)

    //    {

    //        return number \* number;

    //    }

    //}

}

**Extension Method**

**Throw new ……….null**

**\*\***

**LINQ**

**Language Integrated Quarry \*\*\***

**First start with basic**

namespace extra\_advance\_topic\_of\_csharp

{

    public class Book

    {

        public string  Title  { get; set; }

        public  float  Price { get; set; }

    }

}

/////////////////

using System.Collections;

using System.Collections.Generic;

namespace extra\_advance\_topic\_of\_csharp

{

    public class BookRepositary

    {

        public IEnumerable<Book> GetBooks()

        {

            return new List<Book>

            {

                new Book() {Title = "Add.Net Step by step " ,Price = 15},

                new Book() {Title = "Add.Net Step by step " ,Price = 79.99f},

                new Book() {Title = "Add.Net Step by step " ,Price = 5},

                new Book() {Title = "Add.Net Step by step " ,Price = 12},

                new Book() {Title = "Add.Net Step by step " ,Price = 7},

            };

        }

    }

}

///////////////////

using System.Collections.Generic;

using System.Globalization;

using System.Linq;

using System.Linq.Expressions;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

using System;

namespace extra\_advance\_topic\_of\_csharp

{

    public class Program

    {

        static void Main(string[] args)

        {

            var bk= new BookRepositary ().GetBooks();

            var cheapbooklst = new List<Book>();

            foreach (var book in bk)

            {

                if (book.Price < 10)

                {

                    cheapbooklst.Add(book);

                }

            }

            foreach (var book in cheapbooklst)

            {

                Console.WriteLine(book.Title + "   "+  book.Price);

            }

            Console.ReadLine();

        }

    }

}

Using that Linq we have also that

using System.Collections.Generic;

using System.Globalization;

using System.Linq;

using System.Linq.Expressions;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

using System;

namespace extra\_advance\_topic\_of\_csharp

{

    public class Program

    {

        static void Main(string[] args)

        {

            var bk= new BookRepositary ().GetBooks();

            var cheapbooklst = bk.Where(b => b.Price < 10);

            foreach (var book in cheapbooklst)

            {

                Console.WriteLine(book.Title + "   "+  book.Price);

            }

            Console.ReadLine();

        }

    }

}

using System.Collections.Generic;

using System.Globalization;

using System.Linq;

using System.Linq.Expressions;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

using System;

using System.Security.Cryptography;

namespace extra\_advance\_topic\_of\_csharp

{

    public class Program

    {

        static void Main(string[] args)

        {

            var bk= new BookRepositary ().GetBooks();

             var cheaperbook =

                from b in bk

                where b.Price <10

                orderby b.Title

                select b;

            // writing same code using LINQ QUERY OPERATORS

            // LINQ EXTENSION METHODS

            // shorted by Where // .orderby /.select

            var cheapbooklst = bk

                                 .Where(b => b.Price < 100)

                                 .OrderBy(b => b.Title)

                                 .Select(b => b.Title);

            foreach (var book in cheapbooklst)

            {

                Console.WriteLine(book);

               // Console.WriteLine(book.Title + "   "+  book.Price);

            }

            Console.ReadLine();

        }

    }

}

var bk= new BookRepositary ().GetBooks();

           //LINQ Extension Methods

           try {

              var binbook =bk.

                   Single(b=>b.Title == "Add.Net MVC+++ ");

               Console.WriteLine(binbook.Title);

           }

           catch (Exception)

           {

               Console.WriteLine ("Not Found");

           }

           Console.ReadLine();

       }

   }

Design a Stack

A Stack is a data structure for storing a list of elements in a LIFO (last in, first out) fashion. Design a class called Stack with three methods.

using System;

using System.Collections.Generic;

namespace ServerChat\_piolet\_projet

{

        public class stack

        {

            private readonly List<object> \_list = new List<object>();

            public void Push(object obj)

            {

                if(obj == null)

                {

                    throw new InvalidOperationException("you can not add a null object to the stack.");

                }

                \_list.Add(obj);

            }

            public Object Pop()

            {

                if (\_list.Count == 0) throw new InvalidOperationException("There are no element in the stack yet.");

                 var index = \_list.Count - 1;

                var toReturn= \_list[index];

                \_list.RemoveAt(index);

                return toReturn;

            }

        }

    }

///////////////////////

Main

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ServerChat\_piolet\_projet

{

    public class Program

    {

        static void Main(string[] args)

        {

            Console.ForegroundColor = ConsoleColor.Red;

            Console.BackgroundColor= ConsoleColor.White;

            Console.Clear();

            var stk = new stack();

            for (int i = 0; i < 500; i++)

            {

                stk.Push(i);

            }

            for (int i = 0; i < 500; i++)

            {

                Console.WriteLine(stk.Pop());

            }

            Console.ReadLine();

        }

    }

}

\*\*

Topic who leave in first part

string[,] matrix;

            var arr2d = new int[,] {

                {1, 2, 3},// row 0

                {4, 5, 6},// row 1

                {5, 6, 7},// row 2

            };

            Console.WriteLine("the central value is " + arr2d[2,2]); //7

Console.WriteLine("the central value is " + arr2d[2,2]); //7

           var srr2Dstring = new string[,]

           {

               {"one" ,"two"},

               {"three", "four"},

               {"five" , "six" }

           };

           srr2Dstring[0, 0] = "chicken";

           Console.WriteLine(srr2Dstring[0,0]);

Hashtable \*\*\*

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ServerChat\_piolet\_projet

{

    public class Program

    {

        static void Main(string[] args)

        {

            Console.ForegroundColor = ConsoleColor.Red;

            Console.BackgroundColor = ConsoleColor.White;

            Console.Clear();

            // Hashtables

            var table= new Hashtable();

            var std1 = new student(20, "ankur", 9);

            var std2 = new student(19, "deepak", 8);

            var std3 = new student(18, "ayan", 7);

            var std4 = new student(17, "chandan", 4);

             table.Add(std1.id ,std1);

             table.Add(std2.id ,std2);

             table.Add(std3.id ,std3);

             table.Add(std4.id ,std4);

            foreach (DictionaryEntry i in table)

            {

                  var temp = (student)i.Value;

                Console.WriteLine("student id is "   +temp.id);

                Console.WriteLine("student name is "   +temp.name);

                Console.WriteLine("student gpa is "+temp.GPA);

            }

            // table.Add(std1.id ,std1);

            // retrive single item with know ID

           // var   storedata = (student)table[std2.id];

            // retreve allvalues from a hashtable

            //Console.WriteLine(storedata.id);

            //Console.WriteLine(storedata.name);

            //Console.WriteLine(storedata.GPA);

            Console.ReadLine();

        }

    }

    class student

    {

        public int id { get; set; }

        public string name { get; set; }

        public int GPA { get; set; }

        public student(int Id, string name,  int gPA)

        {

            this.id = Id;

            this.name = name;

            GPA = gPA;

        }

    }

}

Problem 01

Using same class what ever using before only change to methode

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Linq.Expressions;

using System.Text;

using System.Threading.Tasks;

namespace ServerChat\_piolet\_projet

{

    public class Program

    {

        static void Main(string[] args)

        {

            Console.ForegroundColor = ConsoleColor.Red;

            Console.BackgroundColor = ConsoleColor.White;

            Console.Clear();

            // Hashtables

            var table = new Hashtable();

            var student = new student[7];

            student[0] = new student(15, "ankur", 9);

            student[1] = new student(16, "deepak", 1);

            student[2] = new student(16, "ayan", 4);

            student[3] = new student(18, "puspendra", 8);

            student[4] = new student(19, "chandan", 7);

            student[5] = new student(20, "Gargi", 10);

            student[6] = new student(20, "Gargi", 10);

            foreach (var i in student)

            {

                if (!table.ContainsKey(i.id))

                {

                    table.Add(i.id, i);

                    Console.WriteLine("student of id 1 is added " + i.id);

                }

                else

                {

                    Console.WriteLine("sorry A student the same id already exist id {0}", i.id);

                }

            }

            //table.Add(student[0].id, student[0]);

            //table.Add(student[1].id, student[1]);

            //table.Add(student[2].id, student[2]);

            //table.Add(student[3].id, student[3]);

            //table.Add(student[4].id, student[4]);

            //table.Add(student[5].id, student[5]);

            ////table.Add(student[6].id, student[6]);

            //foreach (DictionaryEntry item in table)

            //{

            //    var temp = (student)item.Value;

            //    Console.WriteLine("the student id is " + temp.id);

            //    Console.WriteLine("the student name is " + temp.name);

            //    Console.WriteLine("the student gpa is " + temp.GPA);

            //}

            Console.ReadLine();

        }

    }

}

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Linq.Expressions;

using System.Text;

using System.Threading.Tasks;

namespace ServerChat\_piolet\_projet

{

    public class Program

    {

        static void Main(string[] args)

        {

            Console.ForegroundColor = ConsoleColor.Red;

            Console.BackgroundColor = ConsoleColor.White;

            Console.Clear();

            // Hashtables

            var table = new Hashtable();

            var student = new student[7];

            student[0] = new student(15, "ankur", 9);

            student[1] = new student(16, "deepak", 1);

            student[2] = new student(16, "ayan", 4);

            student[3] = new student(18, "puspendra", 8);

            student[4] = new student(19, "chandan", 7);

            student[5] = new student(20, "Gargi", 10);

            student[6] = new student(20, "Gargi", 1);

            foreach (var i in student)

            {

                if (!table.ContainsKey(i.GPA))

                {

                    table.Add(i.GPA, i);

                    Console.WriteLine("student of id 1 is added " + i.GPA);

                }

                else

                {

                    var  changegpa= new Random().Next(0,8)+ i.GPA;

                    Console.WriteLine("sorry A student the same id already exist id {0} ", i.GPA);

                    Console.WriteLine("your new id is " +  changegpa);

                }

            }

            Console.ReadLine();

        }

    }

}

D

Dictionary :-

namespace ServerChat\_piolet\_projet

{

    public class Employess

    {

        public string Role { get; set; }

        public string Name { get; set; }

        public int Age { get; set; }

        public float Rate { get; set; }

        public float Salary

        {

            get { return Rate \* 8 \* 5 \* 4 \* 12; }

        }

        public Employess(string role, string name ,int age,float rate)

        {

            this.Role = role;

            this.Name = name;

            this.Age = age;

            this.Rate= rate;

        }

    }

}

Sol is

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Linq.Expressions;

using System.Text;

using System.Threading.Tasks;

namespace ServerChat\_piolet\_projet

{

    public class Program

    {

        static void Main(string[] args)

        {

            Console.ForegroundColor = ConsoleColor.Red;

            Console.BackgroundColor = ConsoleColor.White;

            Console.Clear();

            Employess[] empl ={

                         new Employess("ceo", "ankur", 95, 800),

                         new Employess("Manger", "saini", 9, 200),

                          new Employess("HR", "chandan", 5, 200),

                           new Employess("sec", "ayan", 75, 200),

                           new Employess("srsoft", "farook", 55, 20),

                          new Employess("soft", "brijesh", 15, 100),

                          };

            var myDictionary = new Dictionary<int, string>() {

                    // key - value

                    { 1 , "one" },

                    { 2 , "two" },

                    { 3 , "three" }

                                       };

            var employdatabase = new Dictionary<string, Employess>();

            foreach (var key in empl)

            {

                employdatabase.Add(key.Role, key);

            }

               var fetchdatabase=employdatabase["ceo"];

            Console.WriteLine(fetchdatabase.Role);

            Console.WriteLine(fetchdatabase.Name);

            Console.WriteLine(fetchdatabase.Salary);

            foreach (var  item in employdatabase)

            {

                var itr = item.Value;

                Console.WriteLine("my name is " + itr.Name);

                Console.WriteLine("my role is " + itr.Role);

                Console.WriteLine("my age  is " + itr.Age);

                Console.WriteLine("my salaray  is " + itr.Salary);

                //Console.WriteLine("my name is " + itr.Name);

            }

            // DictonariesDemo

            // Auto - car

            Console.ReadLine();

        }

    }

}

Using for

foreach (var key in empl)

           {

               employdatabase.Add(key.Role, key);

           }

           var key2 = "ceo";

           if (employdatabase.ContainsKey(key2))

           {

                      var fetchdatabase=employdatabase["ceo"];

                   Console.WriteLine(fetchdatabase.Role);

                   Console.WriteLine(fetchdatabase.Name);

                   Console.WriteLine(fetchdatabase.Salary);

           }

           else

           {

               throw new Exception("key is not present");

           }

           for (int i = 0; i < employdatabase.Count; i++)

           {

               var kp = employdatabase.ElementAt(i);

               Console.WriteLine(kp.Key);

              var temp= kp.Value;

               Console.WriteLine("Empoly Name" + temp.Name );

               Console.WriteLine("Employee Role " + temp.Role);

               Console.WriteLine("Employee Role " + temp.Role);

               Console.WriteLine("Employee age " + temp.Age);

               Console.WriteLine("Employee salery " + temp.Salary);

           }

// how to update inside the dicenry

var keytoudate = "HR";

if (employdatabase.ContainsKey(keytoudate))

{

    employdatabase[keytoudate] = new Employess("Hr", "devesh", 26, 18);

}

Remove an entry from dictionary

var keyToRemove = "Manger";

            if (employdatabase.Remove(keyToRemove))

            {

                Console.WriteLine("Empoloyee with Role/key is removed" + keyToRemove);

            }

\*\* Stack and Queues

static void Main(string[] args)

{

    Console.ForegroundColor = ConsoleColor.Red;

    Console.BackgroundColor = ConsoleColor.White;

    Console.Clear();

    var stk= new Stack<int>(); // define a new stack

    stk.Push(1);

    stk.Push(2);

        stk.Push(3);

         stk.Push(8);

        stk.Push(4);

    while (stk.Count>0)

    {

        Console.WriteLine(stk.Pop());

        //  print the stack count

        Console.WriteLine("current stack is count is {0}",stk.Count);

    }

    // Peek() will return at the top of the without removing it

   // Console.WriteLine(stk.Peek());

    // DictonariesDemo

    // Auto - car