Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace CSharpAssignment\_1

{

class Program

{

delegate List<int> delegate\_obj1(List<int> abc);

static void Main(string[] args)

{

Console.WriteLine("------------------2D Array-------------------");

TwoDArray A = new TwoDArray();

A.GetValue();

Console.WriteLine("------------------Multiple Inheritance-------------------");

MultipleInheritance M = new MultipleInheritance();

M.GetMessage();

Console.WriteLine("------------------Virtual class-------------------");

BaseClass B = new BaseClass();

B.GetMessage();

VirtualMethod V = new VirtualMethod();

V.GetMessage();

Console.WriteLine("------------------Delegates-------------------");

clsDelegate c = new clsDelegate();

delegate\_obj1 \_delegateobj = c.AllDivisibleBy3;

List<int> numberlist = new List<int> { 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 18, 21, 30 };

Console.Write("divisible by 3 list from numlist : ");

\_delegateobj(numberlist);

Console.WriteLine();

Console.WriteLine("------------------lambda expression-------------------");

foreach (var item in numberlist.FindAll(i => i % 3 == 0))

{

Console.Write(item + ",");

}

Console.WriteLine();

Console.WriteLine("------------------Extend the functionality of the System.String-------------------");

Console.WriteLine("Check mail is vaild : subasree@gmail.com is {0}", Extented.IsEmail("subasree@gmail.com"));

Console.WriteLine("Check mail is vaild : subasree.com is {0}", Extented.IsEmail("subasree.com"));

Console.ReadLine();

}

}

#region 2dArray

class TwoDArray

{

public void GetValue()

{

string[,] Matrix = new string[,]

{

{"1","2","3" },

{"4","5","6" },

{"7","8","9" }

};

for (int i = 0; i <= Matrix.GetUpperBound(0); i++)

{

Console.WriteLine("{0} {1} {2}", Matrix[i, 0], Matrix[i, 1], Matrix[i, 2]);

}

}

}

#endregion 2dArray

#region Multiple Inheritance

class MultipleInheritance :Ibase1,Ibase2

{

public void GetMessage()

{

Console.WriteLine("Multiple Inheritance !");

}

}

public interface Ibase1

{

void GetMessage();

}

public interface Ibase2

{

void GetMessage();

}

#endregion Multiple Inheritance

#region Virtual Method

class VirtualMethod : BaseClass

{

public override void GetMessage()

{

Console.WriteLine("From Child Class");

}

}

public class BaseClass

{

public virtual void GetMessage()

{

Console.WriteLine("From Base Class");

}

}

#endregion Virtual Method

#region Delegate

class clsDelegate

{

public List<int> AllDivisibleBy3(List<int> numlist)

{

List<int> result = numlist.FindAll((int i) => { return i % 3 == 0; });

foreach (var item in result)

{

Console.Write(item + ",");

}

return result;

}

}

#endregion Delegate

#region Regular Expression

public static class Extented

{

public static bool IsEmail(string Input)

{

var result = Regex.Match(Input, @"\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*", RegexOptions.IgnoreCase);

return result.Success;

}

}

#endregion Regular Expression

}

Output:

