1. Write a program in C# Sharp for a 2D array of size 3x3 and print the matrix
2. Write an application to implement multiple inheritance and overriding using virtual method.
3. Application is maintaining a collection (List) of Integers as NumList. We need to print the numbers from NumList those are divisible by 3. Implement the functionality using:

* Delegates
* Lambda Expression.

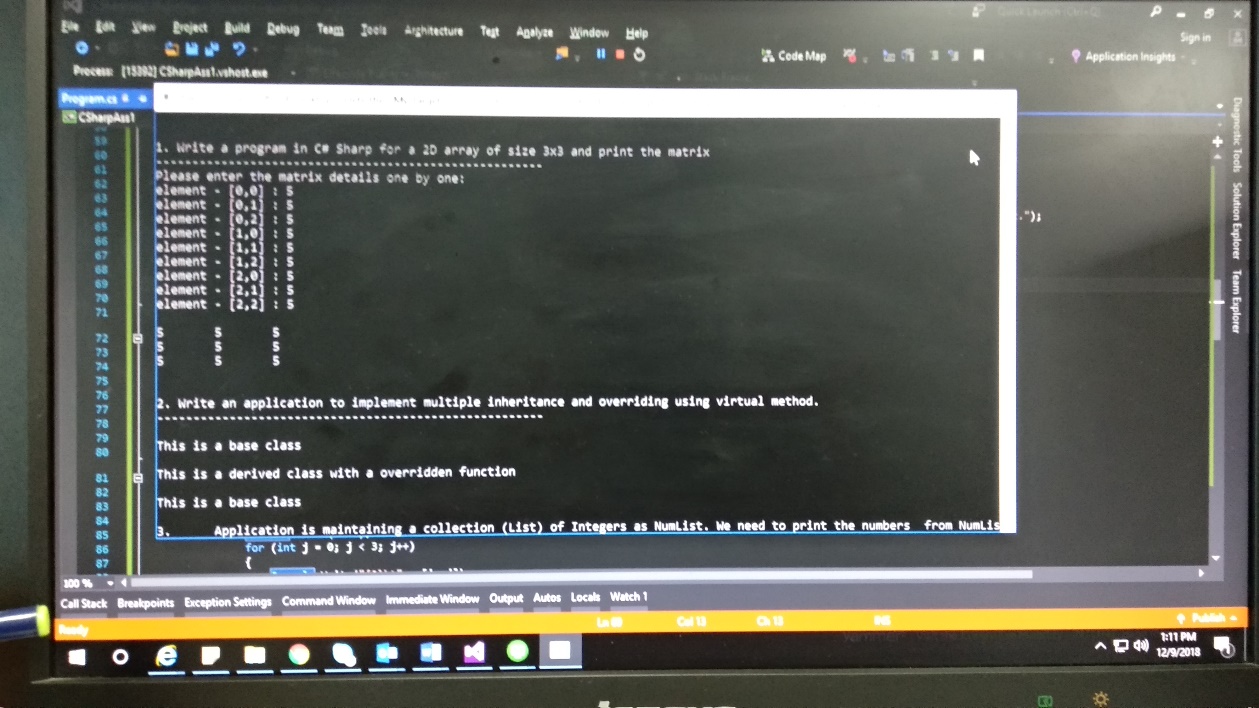
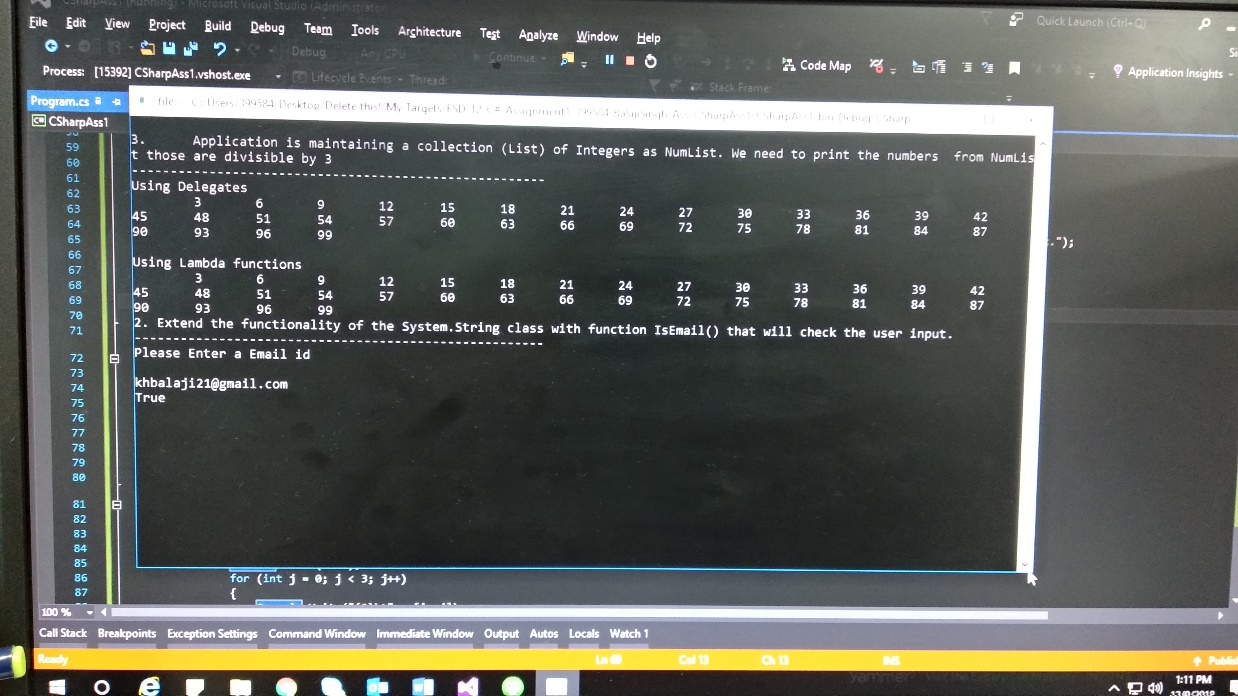
1. Extend the functionality of the System.String class with function IsEmail() that will check the user input.

Sample Input: Usre@MyApp.com

Sample Output: True.

Note: Use Regex Expressions.

Output Screenshots:

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

namespace CSharpAss1

{

public delegate void divby3delegate(List<int> l);

public static class stringExtension

{

public static Boolean IsEmail(string email)

{

Regex regex = new Regex(@"^([\w\.\-]+)@([\w\-]+)((\.(\w){2,3})+)$");

Match match = regex.Match(email);

if (match.Success)

return true;

else

return false;

}

}

public class Program

{

public static void Main(string[] args)

{

int i, j;

int[,] Matrix = new int[3, 3];

// 1. Write a program in C# Sharp for a 2D array of size 3x3 and print the matrix

Console.Write("\n\n1. Write a program in C# Sharp for a 2D array of size 3x3 and print the matrix\n");

Console.Write("------------------------------------------------------\n");

/\* Stored values into the array\*/

Console.Write("Please enter the matrix details one by one:\n");

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

{

Console.Write("element - [{0},{1}] : ", i, j);

Matrix[i, j] = Convert.ToInt32(Console.ReadLine());

}

}

printMatrix(Matrix);

// 2. Write an application to implement multiple inheritance and overriding using virtual method.

displayMulInhOver();

// 3. Application is maintaining a collection (List) of Integers as NumList. We need to print the numbers from NumList those are divisible by 3.

// Implement the functionality using:

// • Delegates

// • Lambda Expression

Console.WriteLine("\n3. Application is maintaining a collection (List) of Integers as NumList. We need to print the numbers from NumList those are divisible by 3");

Console.Write("------------------------------------------------------\n");

divby3delegate delObj = new divby3delegate(f\_divby3);

List<int> l = Enumerable.Range(1, 100).ToList();

delObj(l);

factorsof3();

// 4. Extend the functionality of the System.String class with function IsEmail() that will check the user input.

// Sample Input: Usre @MyApp.com

// Sample Output: True.

// Note: Use Regex Expressions.

Console.WriteLine("\n2. Extend the functionality of the System.String class with function IsEmail() that will check the user input.");

Console.Write("------------------------------------------------------\n");

Console.WriteLine("Please Enter a Email id\n");

Console.WriteLine(stringExtension.IsEmail(Console.ReadLine()).ToString());

Console.ReadLine();

}

public static void f\_divby3(List<int> l)

{

Console.WriteLine("Using Delegates");

List<int> output = l.FindAll(x => x % 3 == 0);

foreach (int item in output)

{

Console.Write("\t" + item);

}

}

public static void printMatrix(int[,] x)

{

for (int i = 0; i < 3; i++)

{

Console.Write("\n");

for (int j = 0; j < 3; j++)

{

Console.Write("{0}\t", x[i, j]);

}

}

Console.Write("\n\n");

}

public static void displayMulInhOver()

{

Console.WriteLine("\n2. Write an application to implement multiple inheritance and overriding using virtual method.");

Console.Write("------------------------------------------------------\n");

baseClass b;

b = new baseClass();

b.displaymsg();

b = new derivedClass();

b.displaymsg();

b = new derivedClass2();

b.displaymsg();

}

public static void factorsof3()

{

List<int> divby3 = Enumerable.Range(1, 100).ToList();

lambdaclass usingLambdaClass = new lambdaclass();

usingLambdaClass.f\_divby3(divby3);

delegateClass usingDelegates = new delegateClass();

//divby3delegate delObj = new divby3delegate(usingDelegates.)

}

}

public class baseClass

{

public virtual void displaymsg()

{

Console.WriteLine("\nThis is a base class");

}

}

public class derivedClass : baseClass

{

public override void displaymsg()

{

Console.WriteLine("\nThis is a derived class with a overridden function");

}

}

public class derivedClass2 : baseClass

{

new void displaymsg()

{

Console.WriteLine("\nThis will not print, as it is not overriden");

}

}

public class delegateClass

{

}

public class lambdaclass

{

public void f\_divby3(List<int> l)

{

Console.WriteLine("\n\nUsing Lambda functions");

List<int> output = l.FindAll(x => x % 3 == 0);

foreach (int item in output)

{

Console.Write("\t" + item);

}

}

}

}