OOP LAB TASK 10

Name: Shahmeer khan.

Class ID: 106278.

Student ID: 12113.

Task:

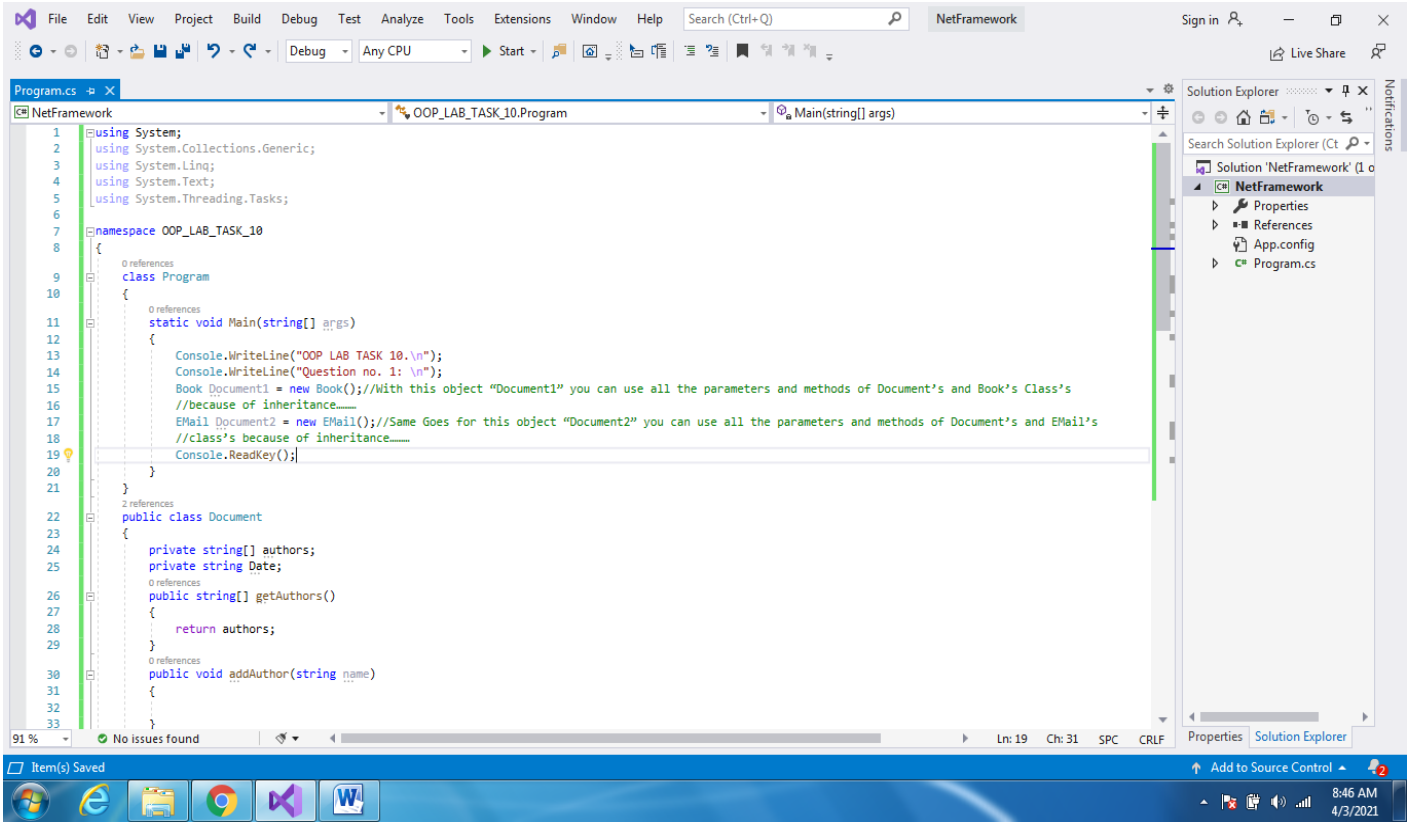
Question 1:

*Inputted Code:

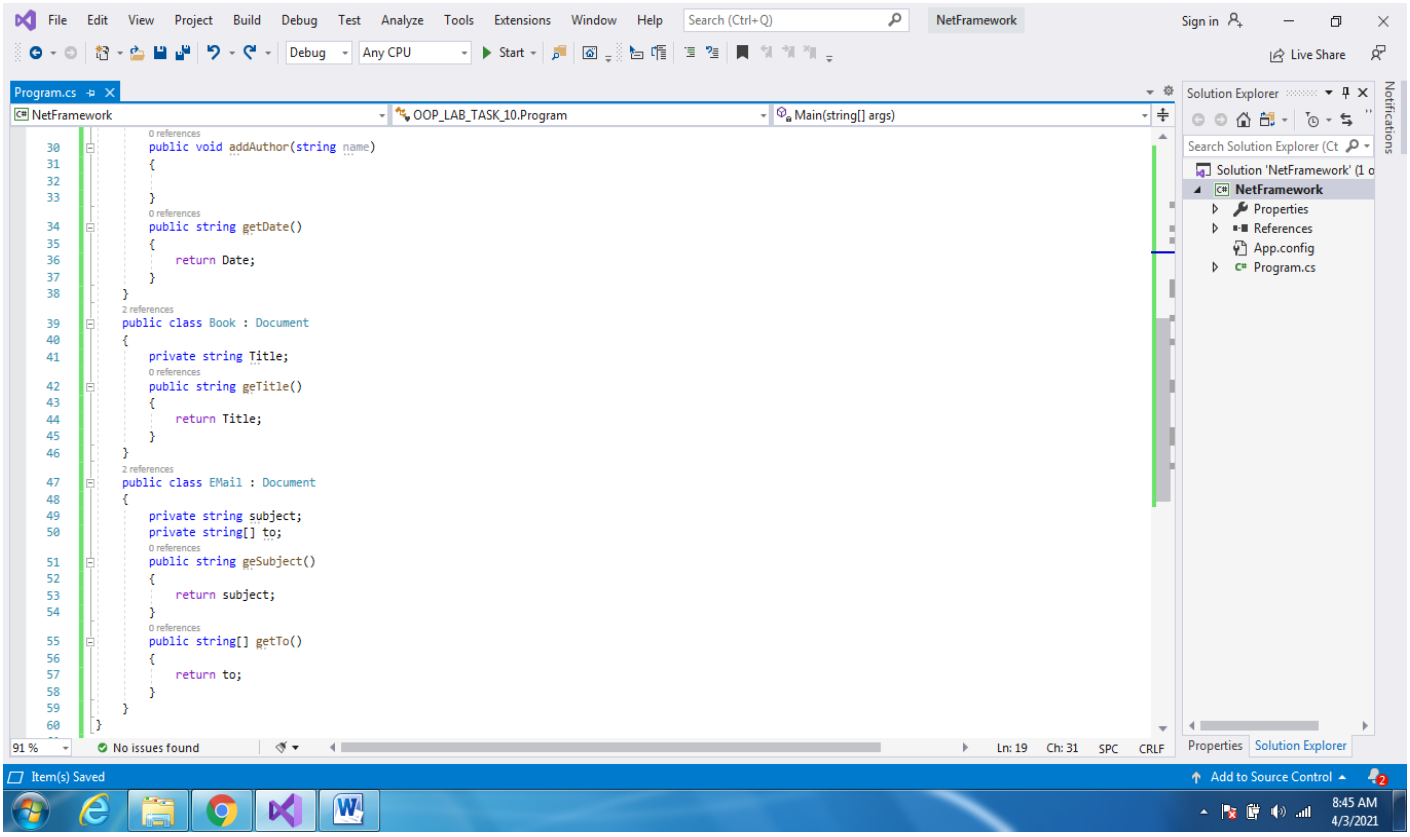
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OOP_LAB_TASK_10
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("OOP LAB TASK 10.\n");
            Console.WriteLine("Question no. 1: \n");
            Book Document1 = new Book();//With this object “Document1” you can use all
the parameters and methods of Document’s and Book’s Class’s
            //because of inheritance……
            EMail Document2 = new EMail();//Same Goes for this object “Document2” you can
use all the parameters and methods of Document’s and EMail’s
            //class’s because of inheritance……
            Console.ReadKey();
        }
    }
    public class Document
    {
        private string[] authors;
        private string Date;
        public string[] getAuthors()
        {
            return authors;
        }
        public void addAuthor(string name)
```

```
        {

        }
        public string getDate()
        {
            return Date;
        }
    }
    public class Book : Document
    {
        private string Title;
        public string geTitle()
        {
            return Title;
        }
    }
    public class EMail : Document
    {
        private string subject;
        private string[] to;
        public string geSubject()
        {
            return subject;
        }
        public string[] getTo()
        {
            return to;
        }
    }
}
```

//SS on Next Page……

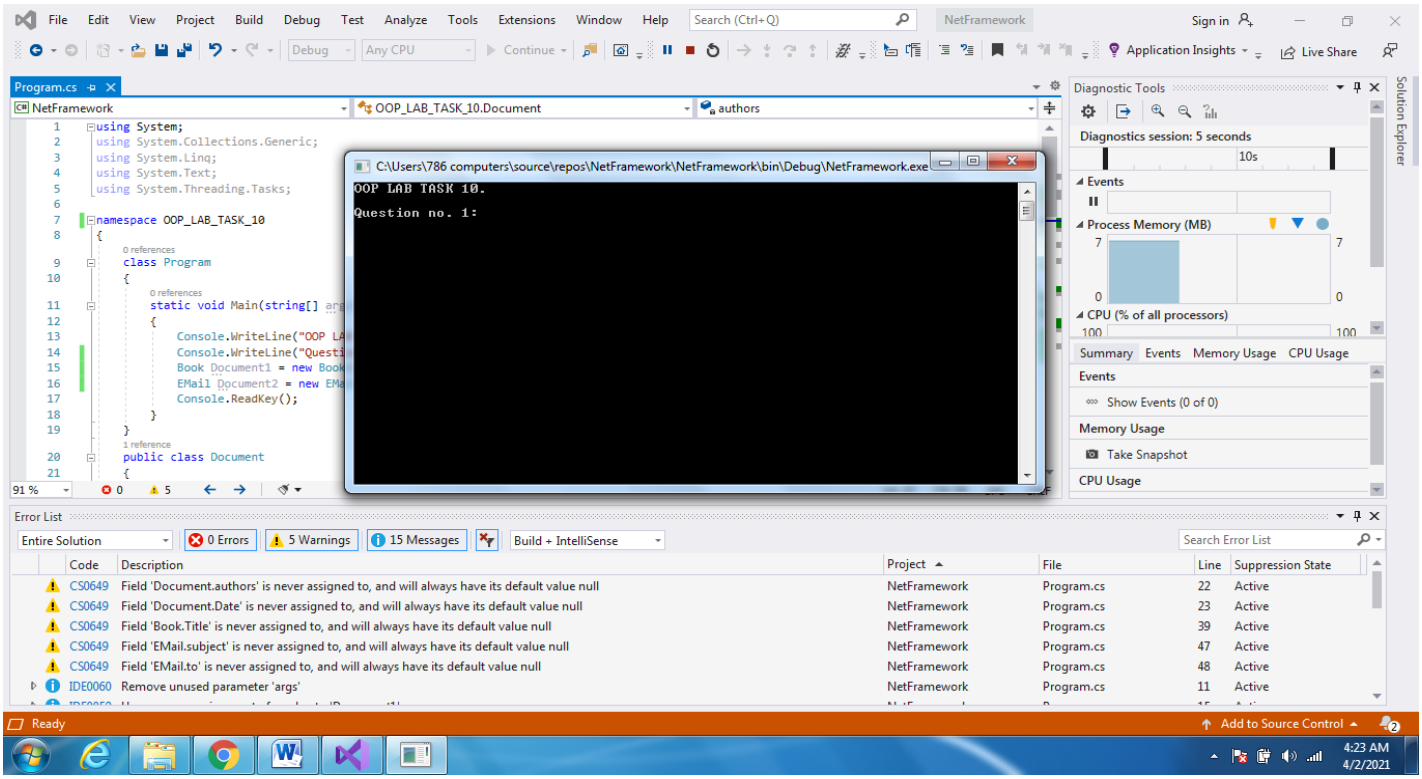


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OOP_LAB_TASK_10
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("OOP LAB TASK 10.\n");
            Console.WriteLine("Question no. 1: \n");
            Book Document1 = new Book();//With this object “Document1” you can use all the parameters and methods of Document’s and Book’s Class’s
            //because of inheritance……..
            EMail Document2 = new EMail();//Same Goes for this object “Document2” you can use all the parameters and methods of Document’s and EMail’s
            //class’s because of inheritance……..
            Console.ReadKey();
        }
    }
    2 references
    public class Document
    {
        private string[] authors;
        private string Date;
        0 references
        public string[] getAuthors()
        {
            return authors;
        }
        0 references
        public void addAuthor(string name)
        {

        }
```



```
        0 references
        public void addAuthor(string name)
        {

        }
        0 references
        public string getDate()
        {
            return Date;
        }
    }
    2 references
    public class Book : Document
    {
        private string Title;
        0 references
        public string geTitle()
        {
            return Title;
        }
    }
    2 references
    public class EMail : Document
    {
        private string subject;
        private string[] to;
        0 references
        public string geSubject()
        {
            return subject;
        }
        0 references
        public string[] getTo()
        {
            return to;
        }
    }
}
```

Output: (//Well apparently it won't have any output just a black screen since it was only the implementation of classes diagrams any way…):



Question 2:

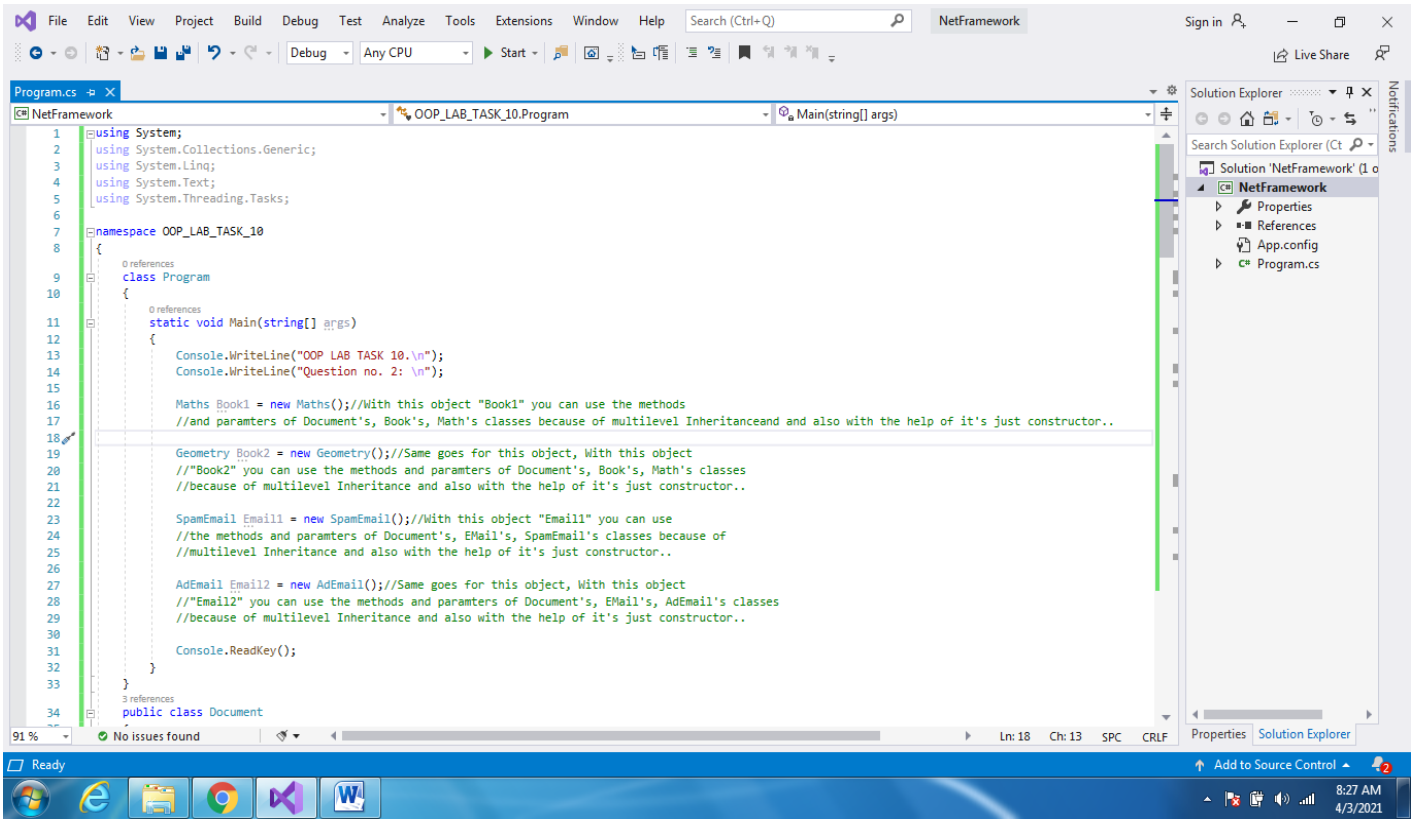
*Inputted Code:

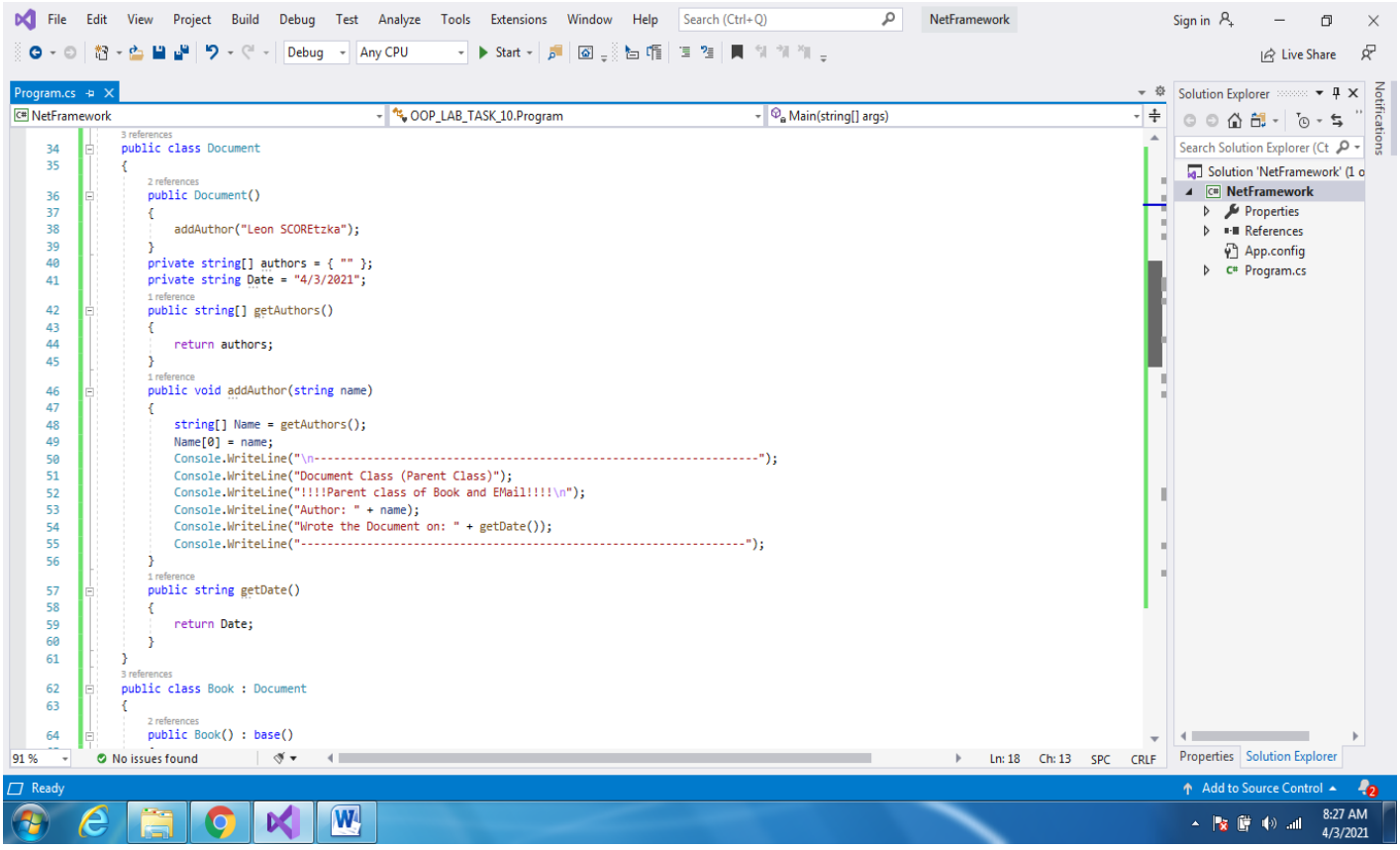
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OOP_LAB_TASK_10
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("OOP LAB TASK 10.\n");
            Console.WriteLine("Question no. 2: \n");

            Maths Book1 = new Maths();//With this object "Book1" you can use the methods
            //and paramters of Document's, Book's, Math's classes because of multilevel
Inheritanceand and also with the help of it's just constructor..
```

```
            Geometry Book2 = new Geometry();//Same goes for this object, With this object
            //"Book2" you can use the methods and paramters of Document's, Book's, Math's
classes
            //because of multilevel Inheritance and also with the help of it's just
constructor..

            SpamEmail Email1 = new SpamEmail();//With this object "Email1" you can use
            //the methods and paramters of Document's, EMail's, SpamEmail's classes
because of
            //multilevel Inheritance and also with the help of it's just constructor..

            AdEmail Email2 = new AdEmail();//Same goes for this object, With this object
            //"Email2" you can use the methods and paramters of Document's, EMail's,
AdEmail's classes
            //because of multilevel Inheritance and also with the help of it's just
constructor..

            Console.ReadKey();
        }
    }
    public class Document
    {
        public Document()
        {
            addAuthor("Leon SCOREtzka");
        }
        private string[] authors = { "" };
        private string Date = "4/3/2021";
        public string[] getAuthors()
        {
            return authors;
        }
        public void addAuthor(string name)
        {
            string[] Name = getAuthors();
            Name[0] = name;
            Console.WriteLine("\n-------------------------------------------------------
-----------");
            Console.WriteLine("Document Class (Parent Class)");
            Console.WriteLine("!!!!Parent class of Book and EMail!!!!\n");
            Console.WriteLine("Author: " + name);
            Console.WriteLine("Wrote the Document on: " + getDate());
            Console.WriteLine("-------------------------------------------------------
---------");
        }
        public string getDate()
        {
            return Date;
        }
    }
    public class Book : Document
    {
        public Book() : base()
        {
            method();
        }
        private string Title = "Title of the Books are Maths and Geometry.";
```

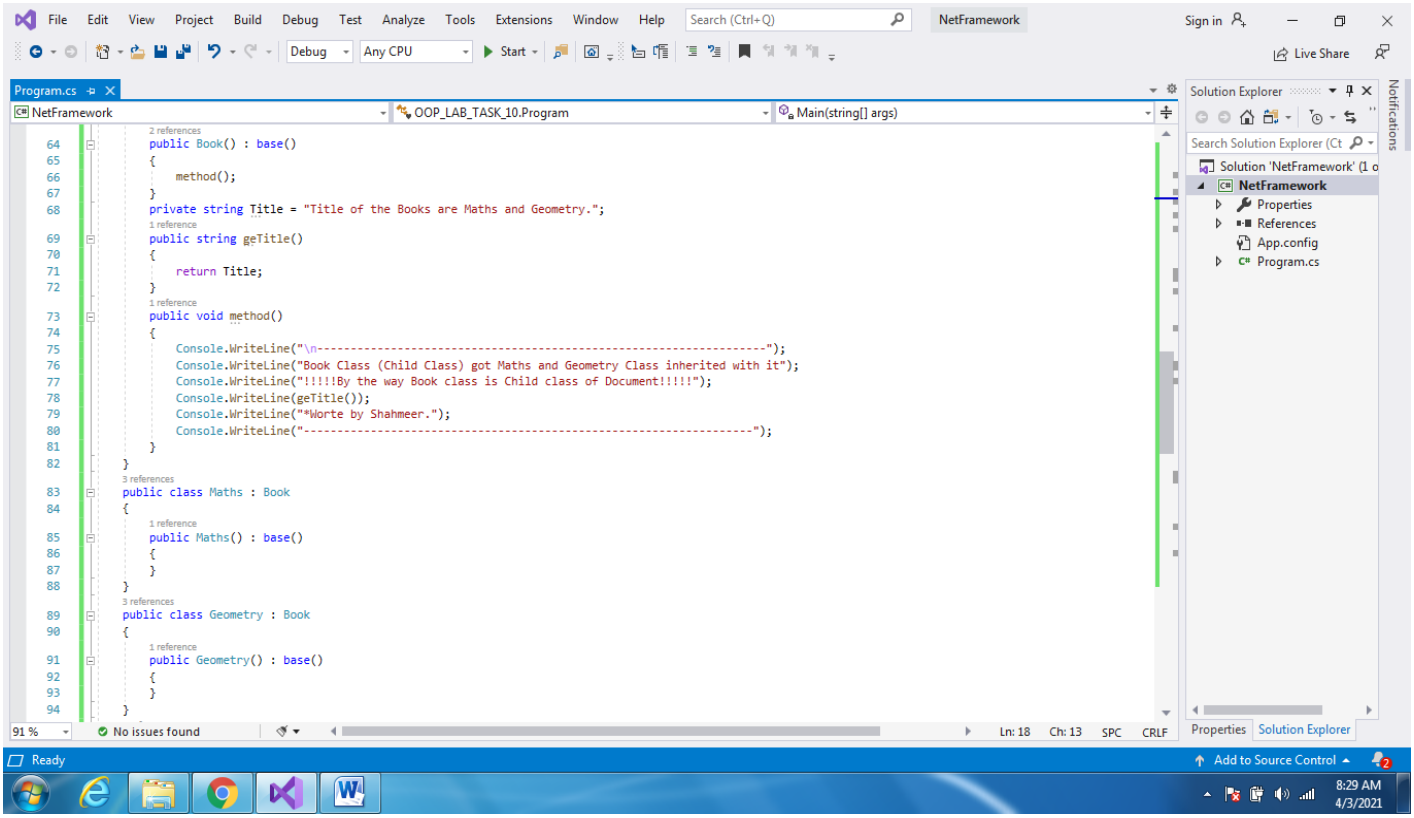
```
        public string geTitle()
        {
            return Title;
        }
        public void method()
        {
            Console.WriteLine("\n-------------------------------------------------------
-----------");
            Console.WriteLine("Book Class (Child Class) got Maths and Geometry Class
inherited with it");
            Console.WriteLine("!!!!!By the way Book class is Child class of
Document!!!!!");
            Console.WriteLine(geTitle());
            Console.WriteLine("*Worte by Shahmeer.");
            Console.WriteLine("-------------------------------------------------------
---------");
        }
    }
    public class Maths : Book
    {
        public Maths() : base()
        {
        }
    }
    public class Geometry : Book
    {
        public Geometry() : base()
        {
        }
    }
    public class EMail : Document
    {
        public EMail() : base()
        {
            Method("To FCBM");
        }
        private string subject = "got Transferred";
        private string[] to = { "" };
        public string getSubject()
        {
            return subject;
        }
        public void Method(string updater)
        {
            string[] TO = getTo();
            TO[0] = updater;
            Console.WriteLine("\n-------------------------------------------------------
-----------");
            Console.WriteLine("EMail Class (Child Class) got SpamEmail and AdEmail class
inherited with it.");
            Console.WriteLine("!!!!!By the way EMail Class is Child Class of Document
Class!!!!!");
            Console.WriteLine("\nLeon SCOREtzka " + getSubject() + " " + updater);
            Console.WriteLine("-------------------------------------------------------
---------");
        }
        public string[] getTo()
        {
```

```
            return to;
        }
    }
    public class SpamEmail : EMail
    {
        public SpamEmail()
        {
        }
    }
    public class AdEmail : EMail
    {
        public AdEmail():base()
        {
        }
    }
}
```

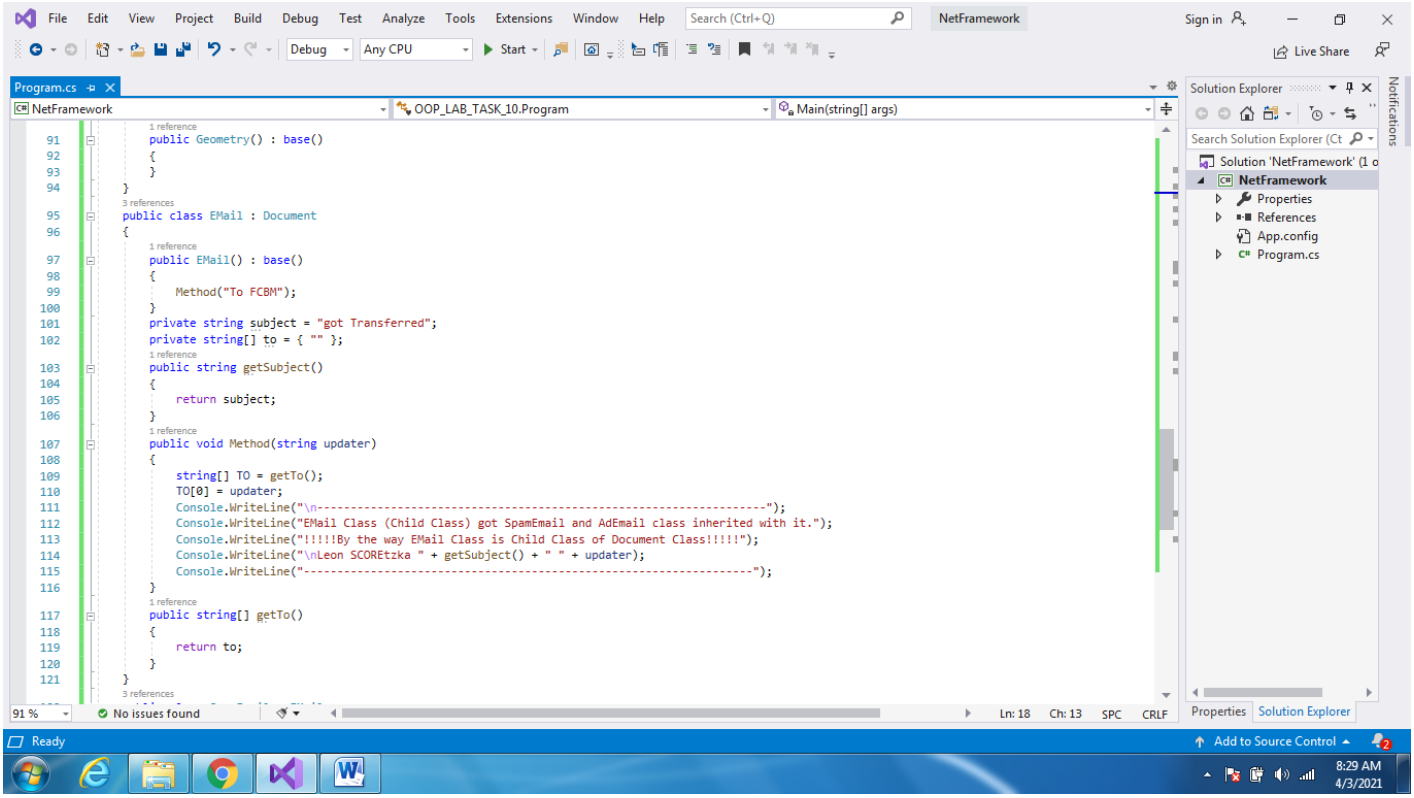




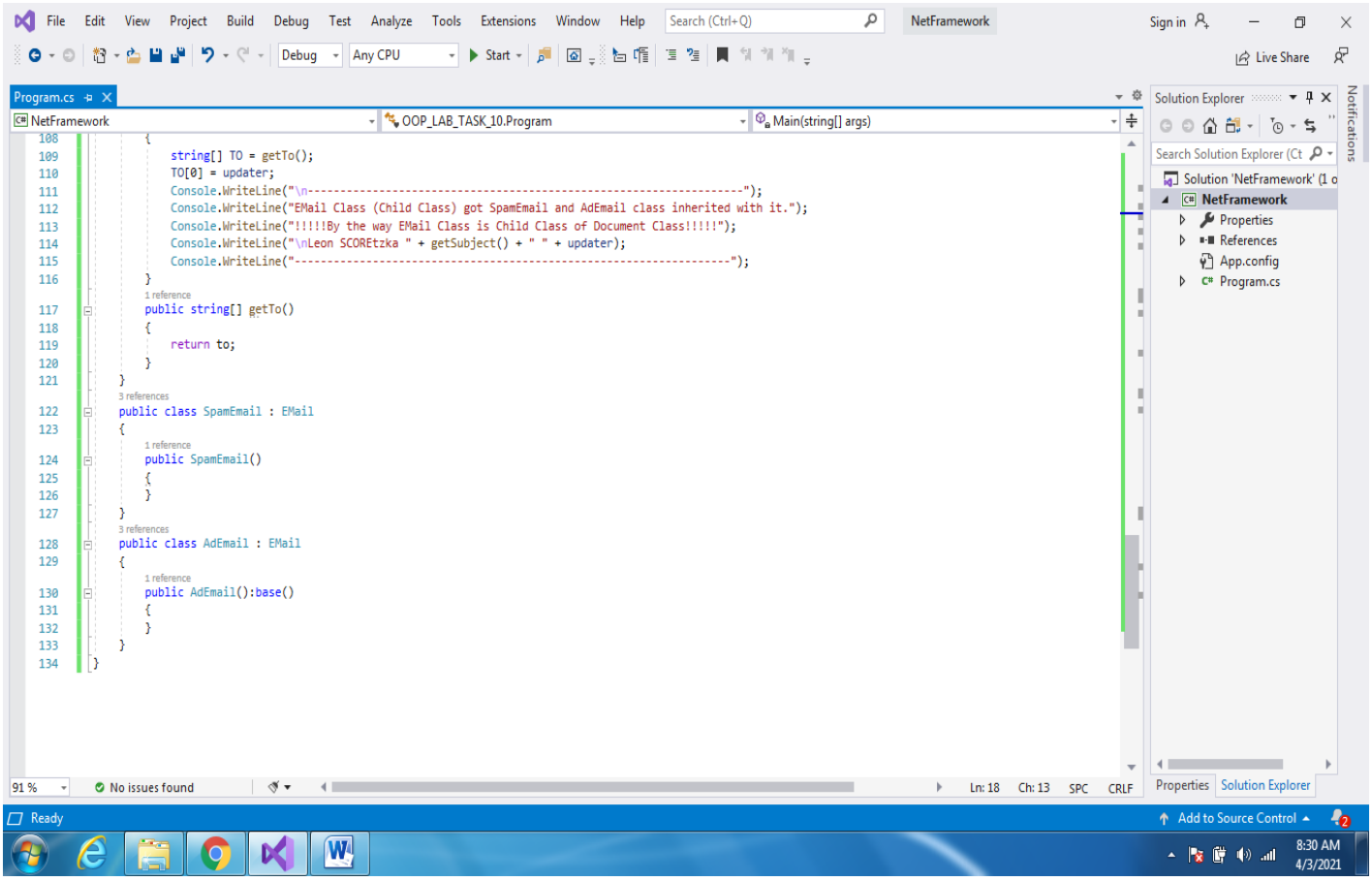
```
3 references
public class Document
{
    2 references
    public Document()
    {
        addAuthor("Leon SCOREtzka");
    }
    private string[] authors = { "" };
    private string Date = "4/3/2021";
    1 reference
    public string[] getAuthors()
    {
        return authors;
    }
    1 reference
    public void addAuthor(string name)
    {
        string[] Name = getAuthors();
        Name[0] = name;
        Console.WriteLine("\n--------------------------------------------------------------------");
        Console.WriteLine("Document Class (Parent Class)");
        Console.WriteLine("!!!!Parent class of Book and EMail!!!!\n");
        Console.WriteLine("Author: " + name);
        Console.WriteLine("Wrote the Document on: " + getDate());
        Console.WriteLine("--------------------------------------------------------------------");
    }
    1 reference
    public string getDate()
    {
        return Date;
    }
}
3 references
public class Book : Document
{
    2 references
    public Book() : base()
```



```
    2 references
    public Book() : base()
    {
        method();
    }
    private string Title = "Title of the Books are Maths and Geometry.";
    1 reference
    public string geTitle()
    {
        return Title;
    }
    1 reference
    public void method()
    {
        Console.WriteLine("\n--------------------------------------------------------------------");
        Console.WriteLine("Book Class (Child Class) got Maths and Geometry Class inherited with it");
        Console.WriteLine("!!!!!By the way Book class is Child class of Document!!!!!");
        Console.WriteLine(geTitle());
        Console.WriteLine("*Worte by Shahmeer.");
        Console.WriteLine("--------------------------------------------------------------------");
    }
}
3 references
public class Maths : Book
{
    1 reference
    public Maths() : base()
    {
    }
}
3 references
public class Geometry : Book
{
    1 reference
    public Geometry() : base()
    {
    }
}
```

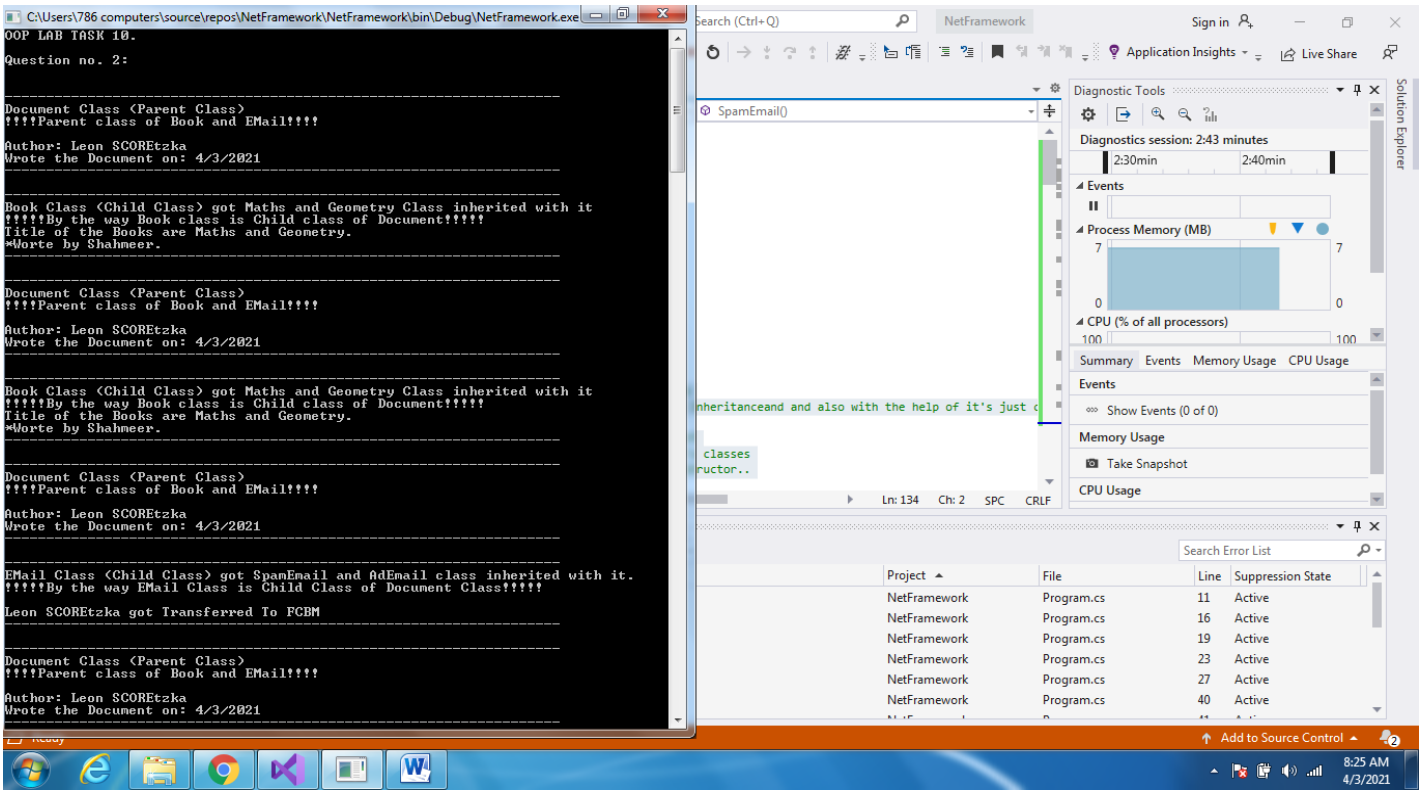


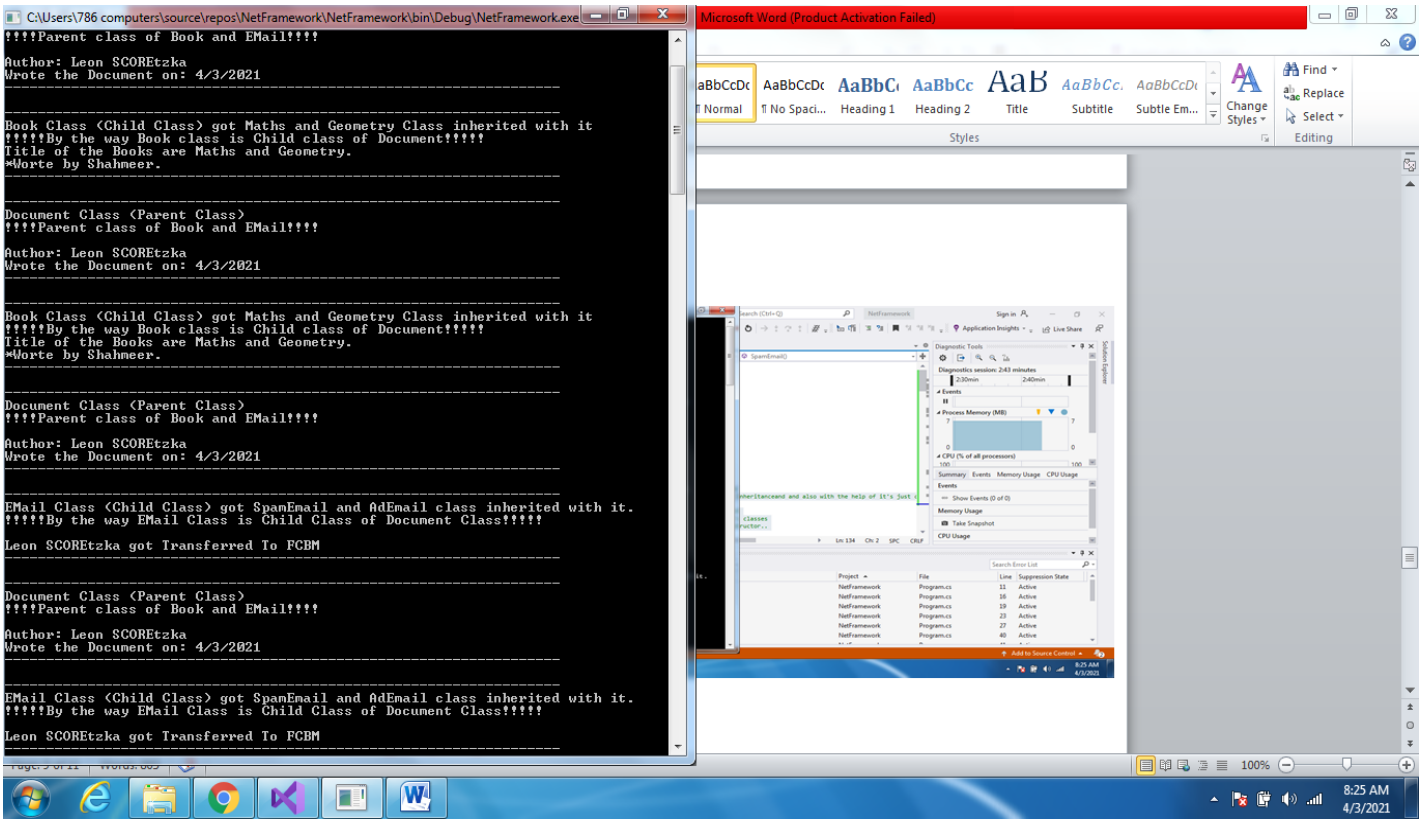
```
1 reference
public Geometry() : base()
{
}
}
3 references
public class EMail : Document
{
1 reference
public EMail() : base()
{
    Method("To FCBM");
}
private string subject = "got Transferred";
private string[] to = { "" };
1 reference
public string getSubject()
{
    return subject;
}
1 reference
public void Method(string updater)
{
    string[] TO = getTo();
    TO[0] = updater;
    Console.WriteLine("\n-----------------------------------------------------------------");
    Console.WriteLine("EMail Class (Child Class) got SpamEmail and AdEmail class inherited with it.");
    Console.WriteLine("!!!!!By the way EMail Class is Child Class of Document Class!!!!!");
    Console.WriteLine("\nLeon SCOREtzka " + getSubject() + " " + updater);
    Console.WriteLine("-----------------------------------------------------------------");
}
1 reference
public string[] getTo()
{
    return to;
}
}
3 references
```



```
{
    string[] TO = getTo();
    TO[0] = updater;
    Console.WriteLine("\n-----------------------------------------------------------------");
    Console.WriteLine("EMail Class (Child Class) got SpamEmail and AdEmail class inherited with it.");
    Console.WriteLine("!!!!!By the way EMail Class is Child Class of Document Class!!!!!");
    Console.WriteLine("\nLeon SCOREtzka " + getSubject() + " " + updater);
    Console.WriteLine("-----------------------------------------------------------------");
}
1 reference
public string[] getTo()
{
    return to;
}
}
3 references
public class SpamEmail : EMail
{
1 reference
public SpamEmail()
{
}
}
3 references
public class AdEmail : EMail
{
1 reference
public AdEmail():base()
{
}
}
}
```

Output: (//since it was only the implementation of classes with multi-level-inheritance and printing some things through the constructors and objects of the child classes of like Math, geometry, AdEmail, SpamEmail just like asked in the question by the teacher which was question 2 sequel of question 1 any way…):



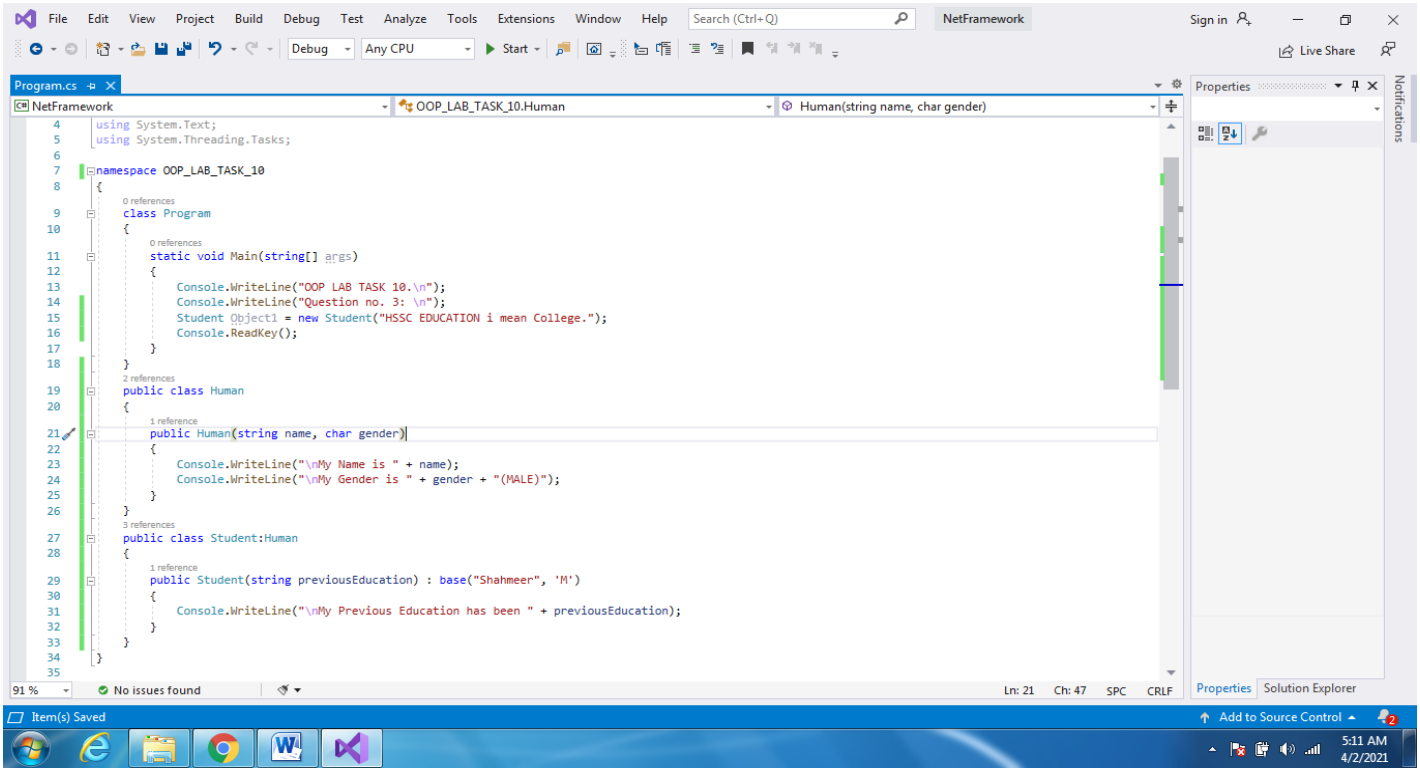


Question 3:

*Inputted Code:

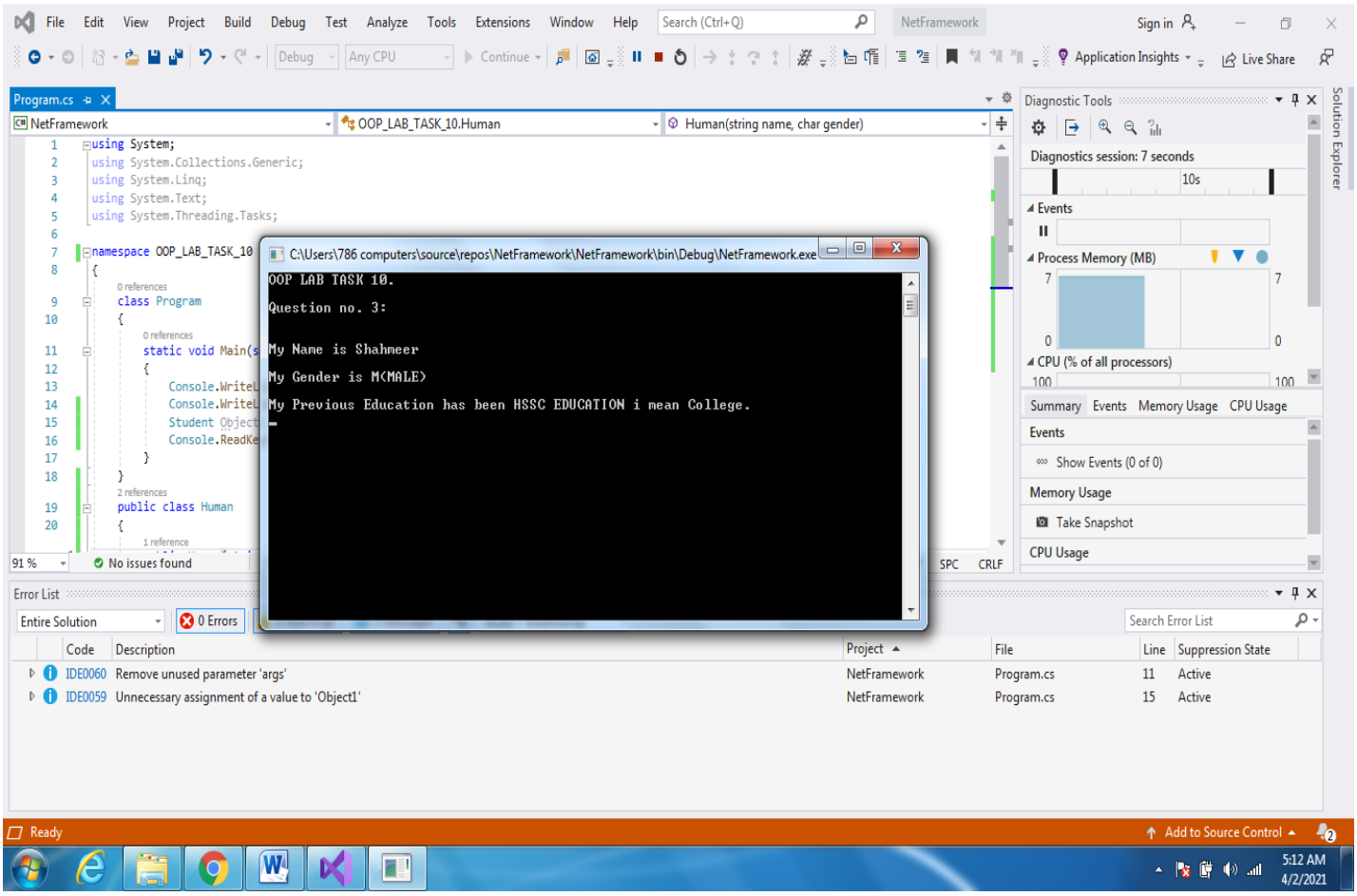
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OOP_LAB_TASK_10
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("OOP LAB TASK 10.\n");
            Console.WriteLine("Question no. 3: \n");
            Student Object1 = new Student("HSSC EDUCATION i mean College.");
            Console.ReadKey();
        }
    }
    public class Human
    {
        public Human(string name, char gender)
        {
            Console.WriteLine("\nMy Name is " + name);
```

```
            Console.WriteLine("\nMy Gender is " + gender + "(MALE)");
        }
    }
    public class Student:Human
    {
        public Student(string previousEducation) : base("Shahmeer", 'M')
        {
            Console.WriteLine("\nMy Previous Education has been " + previousEducation);
        }
    }
}
```



//OUTPUT ON NEXT PAGE……

*Output:


C:\Users\786 computers\source\repos\NetFramework\NetFramework\bin\Debug\NetFramework.exe
OOP LAB TASK 10.
Question no. 3:
My Name is Shahmeer
My Gender is M(MALE)
My Previous Education has been HSSC EDUCATION i mean College.