

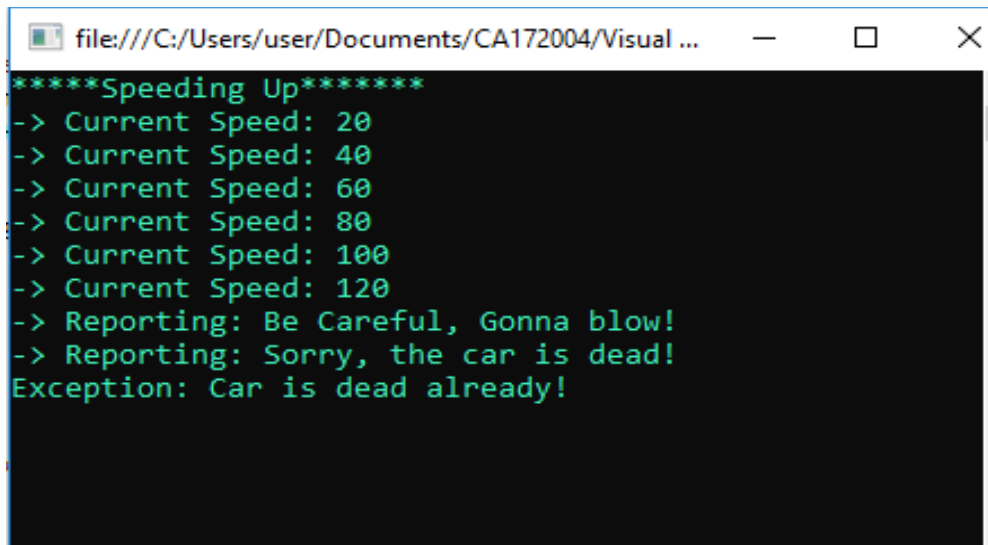
**10) Describe delegates, events, errors and exceptions.**

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

namespace ConsoleApplication26
{
    class Car
    {
        public delegate void EventHandler(string msg);
        public event EventHandler explodeListener; public event EventHandler
        aboutToBlowListener;
        private string name; private bool isExhausted; private int currentSpeed;
        private const int maxSpeed = 140;
        public Car(String name)
        {
            this.name = name;
        }
        public void accelerate(int delta)
        {
            if (isExhausted)
            {
                if (explodeListener != null) explodeListener("Sorry, the car is dead!");
            }
            else
            {
                currentSpeed += delta;
                if (10 >= maxSpeed - currentSpeed && aboutToBlowListener != null)
                {
                    aboutToBlowListener("Be Careful, Gonna blow!");
                }
                if (currentSpeed >= maxSpeed) isExhausted = true;
            }
            else
            {
                Console.WriteLine("-> Current Speed: {0}", currentSpeed);
            }
        }
    }
    class Program
    {
        static void Main(string[] args)
        {

```

```
Car car = new Car("Tesla");
car.aboutToBlowListener += new Car.EventHandler(aboutToBlow);
car.explodListener += new Car.EventHandler(exploded);
Console.WriteLine("*****Speeding Up*****");
try
{
    for (int i = 0; i < 20; i++)
    {
        car.accelerate(20);
    }
}
catch (Exception e)
{
    Console.WriteLine("Exception: Car is dead already!");
}
Console.ReadLine();
}
public static void aboutToBlow(string msg)
{
    Console.WriteLine("-> Reporting: {0}", msg);
}
public static void exploded(string msg)
{
    Console.WriteLine("-> Reporting: {0}", msg); throw new Exception("Car dead");
}
}
}
```

**OUTPUT**

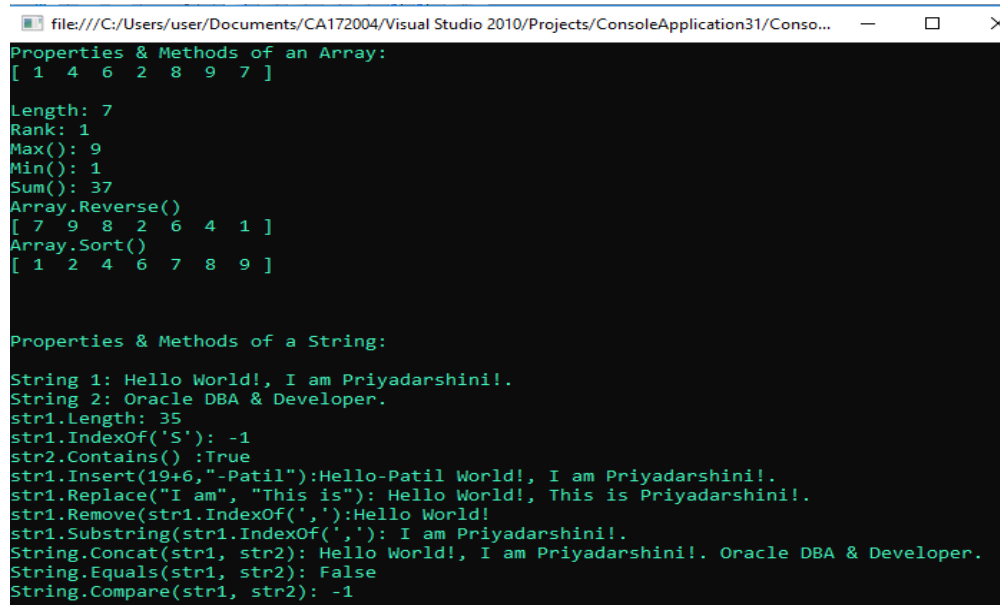
```
file:///C:/Users/user/Documents/CA172004/Visual ...  
*****Speeding Up*****  
-> Current Speed: 20  
-> Current Speed: 40  
-> Current Speed: 60  
-> Current Speed: 80  
-> Current Speed: 100  
-> Current Speed: 120  
-> Reporting: Be Careful, Gonna blow!  
-> Reporting: Sorry, the car is dead!  
Exception: Car is dead already!
```

**11) Describe Arrays and Strings methods with suitable C# program.**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication31
{
class Program
{
static void Main(string[] args)
{
    int[] array = { 1, 4, 6, 2, 8, 9, 7 };
    Console.WriteLine("Properties & Methods of an Array: ");
    displayArray(array);
    Console.WriteLine();
    Console.WriteLine("Length: {0}", array.Length);
    Console.WriteLine("Rank: {0}", array.Rank);
    Console.WriteLine("Max(): {0}", array.Max());
    Console.WriteLine("Min(): {0}", array.Min());
    Console.WriteLine("Sum(): {0}", array.Sum());
    Console.WriteLine("Array.Reverse()"); Array.Reverse(array);
    displayArray(array);
    Console.WriteLine("Array.Sort()"); Array.Sort(array);
    displayArray(array);
    Console.WriteLine();
    Console.WriteLine(" ");
    Console.WriteLine();
    Console.WriteLine("Properties & Methods of a String: ");
    String str1 = "Hello World!, I am Priyadarshini! ";
    Console.WriteLine();
    String str2 = "Oracle DBA & Developer.";
    Console.WriteLine("String 1: {0}", str1);
    Console.WriteLine("String 2: {0}", str2);
    Console.WriteLine("str1.Length: {0}", str1.Length);
    Console.WriteLine("str1.IndexOf('S'): {0}", str1.IndexOf('B'));
    Console.WriteLine("str2.Contains() :{0}",str2.Contains("Developer"));
    Console.WriteLine("str1.Insert(19+6,\"-Patil\"):{0}", str1.Insert(str1.IndexOf('J') + 6,
    "-Patil"));
    Console.WriteLine("str1.Replace(\"I am\", \"This is\"): {0}", str1.Replace("I am",
    "This is"));
    Console.WriteLine("str1.Remove(str1.IndexOf(',')):{0}",
    str1.Remove(str1.IndexOf(',')));
    Console.WriteLine("str1.Substring(str1.IndexOf(',')):{0}",
    str1.Substring(str1.IndexOf(',') + 1));
    Console.WriteLine("String.Concat(str1, str2): {0}", String.Concat(str1, str2));
    Console.WriteLine("String.Equals(str1, str2): {0}", String.Equals(str1, str2));
    Console.WriteLine("String.Compare(str1, str2): {0}", String.Compare(str1, str2));
}
```

```
    Console.ReadLine();  
}  
static void displayArray(int[] a)  
{  
    Console.Write("[");  
    for (int i = 0; i < a.Length; i++)  
    {  
        Console.Write(" {0} ", a[i]);  
    }  
    Console.WriteLine("]");  
}  
}  
}
```

## OUTPUT



```
file:///C:/Users/user/Documents/CA172004/Visual Studio 2010/Projects/ConsoleApplication31/Conso...
Properties & Methods of an Array:
[ 1 4 6 2 8 9 7 ]

Length: 7
Rank: 1
Max(): 9
Min(): 1
Sum(): 37
Array.Reverse()
[ 7 9 8 2 6 4 1 ]
Array.Sort()
[ 1 2 4 6 7 8 9 ]

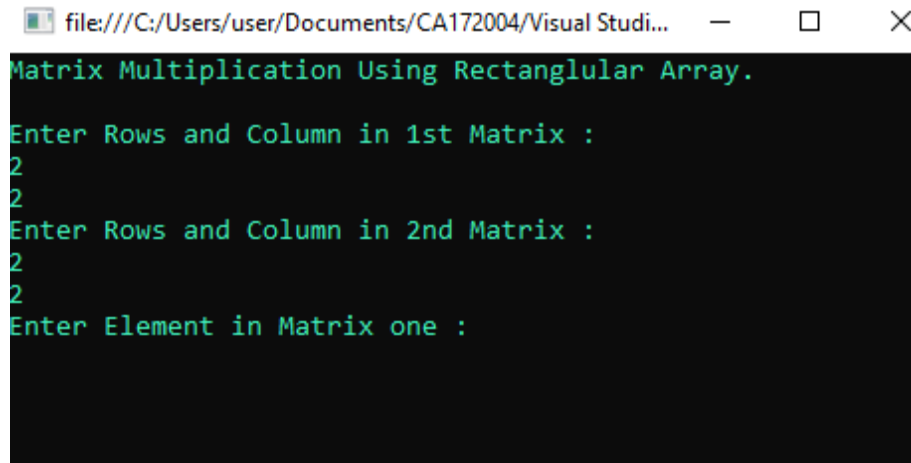
Properties & Methods of a String:
String 1: Hello World!, I am Priyadarshini!.
String 2: Oracle DBA & Developer.
str1.Length: 35
str1.IndexOf('S'): -1
str2.Contains() :True
str1.Insert(19+6,"-Patil"):Hello-Patil World!, I am Priyadarshini!.
str1.Replace("I am", "This is"): Hello World!, This is Priyadarshini!.
str1.Remove(str1.IndexOf(',')):Hello World!
str1.Substring(str1.IndexOf(',')): I am Priyadarshini!.
String.Concat(str1, str2): Hello World!, I am Priyadarshini!. Oracle DBA & Developer.
String.Equals(str1, str2): False
String.Compare(str1, str2): -1
```

**12) Program to multiply to matrices using Rectangular array.**

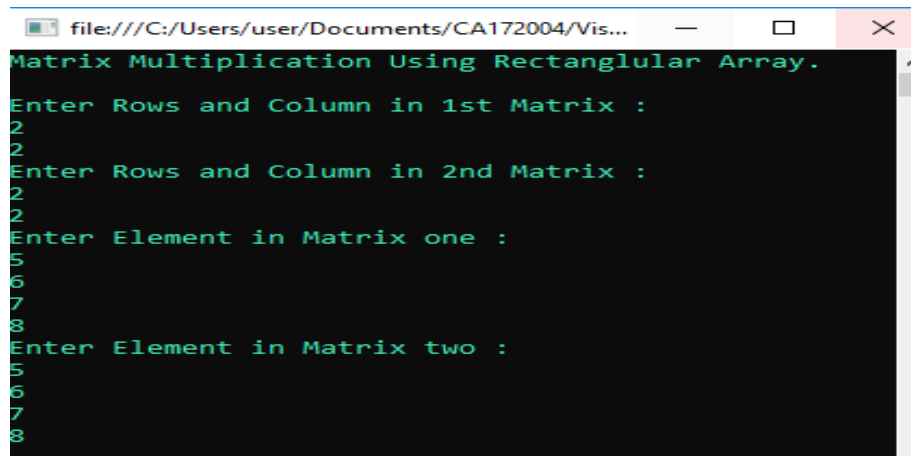
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication30
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Matrix Multiplication Using Rectanglular Array.");
            Console.WriteLine(" ");
            try
            {
                Console.WriteLine("Enter Rows and Column in 1st Matrix : ");
                int r1 = Convert.ToInt16(Console.ReadLine());
                int c1 = Convert.ToInt16(Console.ReadLine());
                Console.WriteLine("Enter Rows and Column in 2nd Matrix : ");
                int r2 = Convert.ToInt16(Console.ReadLine());
                int c2 = Convert.ToInt16(Console.ReadLine());
                if (r1 != c2)
                {
                    Console.WriteLine("Matrix Multiplication Row Column Rule Violated.");
                }
                else
                {
                    int[,] mat1 = new int[r1, c1]; int[,] mat2 = new int[r2, c2];
                    int[,] mat3 = new int[r1, c2];
                    Console.WriteLine("Enter Element in Matrix one : ");
                    for (int i = 0; i < r1; i++)
                    {
                        for (int j = 0; j < c1; j++)
                        {
                            mat1[i, j] = (Convert.ToInt16(Console.ReadLine()));
                        }
                    }
                    Console.WriteLine("Enter Element in Matrix two : ");
                    for (int i = 0; i < r2; i++)
                    {
                        for (int j = 0; j < c2; j++)
                        {
                            mat2[i, j] = (Convert.ToInt16(Console.ReadLine()));
                        }
                    }
                    Console.WriteLine("\nFirst Matrix\n");
                    for (int i = 0; i < r1; i++)
                    {
```

```
for (int j = 0; j < c1; j++)
{
    Console.Write("\t" + mat1[i, j]);
}
Console.WriteLine();
}
Console.WriteLine("\nSecond Matrix\n");
for (int i = 0; i < r2; i++)
{
    for (int j = 0; j < c2; j++)
    {
        Console.Write("\t" + mat2[i, j]);
    }
    Console.WriteLine();
}
Console.WriteLine("\nMultiplication of Matrix\n");
for (int i = 0; i < r1; i++)
{
    for (int j = 0; j < c2; j++)
    {
        for (int k = 0; k < c1; k++)
        {
            mat3[i, j] += mat1[i, k] * mat2[k, j];
        }
    }
}
for (int i = 0; i < r2; i++)
{
    for (int j = 0; j < c2; j++)
    {
        Console.Write("\t" + mat3[i, j]);
    }
    Console.WriteLine();
}
}
}
catch (Exception ex) {
    Console.WriteLine("\n*****");
    Console.WriteLine("Please Enter Numaric value.");
    Console.WriteLine("\n*****");
}
Console.ReadKey();
}
}
}
```

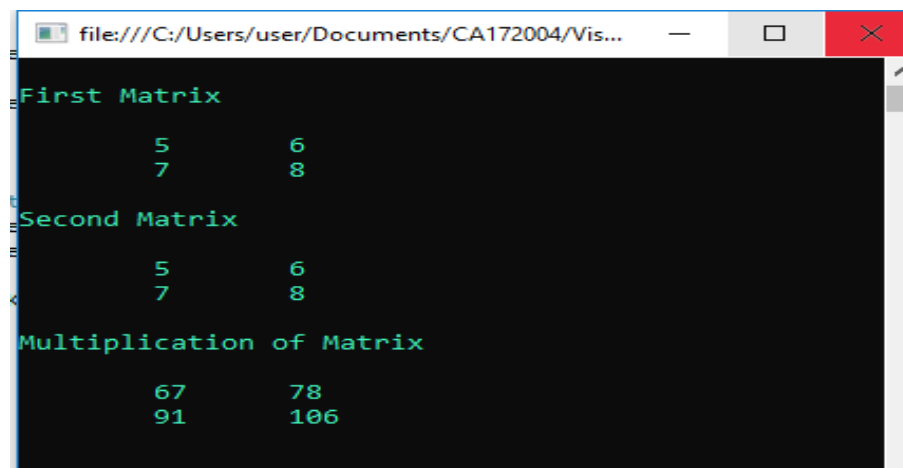


**OUTPUT**

```
file:///C:/Users/user/Documents/CA172004/Visual Studi...
Matrix Multiplication Using Rectangular Array.
Enter Rows and Column in 1st Matrix :
2
2
Enter Rows and Column in 2nd Matrix :
2
2
Enter Element in Matrix one :
```



```
file:///C:/Users/user/Documents/CA172004/Vis...
Matrix Multiplication Using Rectangular Array.
Enter Rows and Column in 1st Matrix :
2
2
Enter Rows and Column in 2nd Matrix :
2
2
Enter Element in Matrix one :
5
6
7
8
Enter Element in Matrix two :
5
6
7
8
```



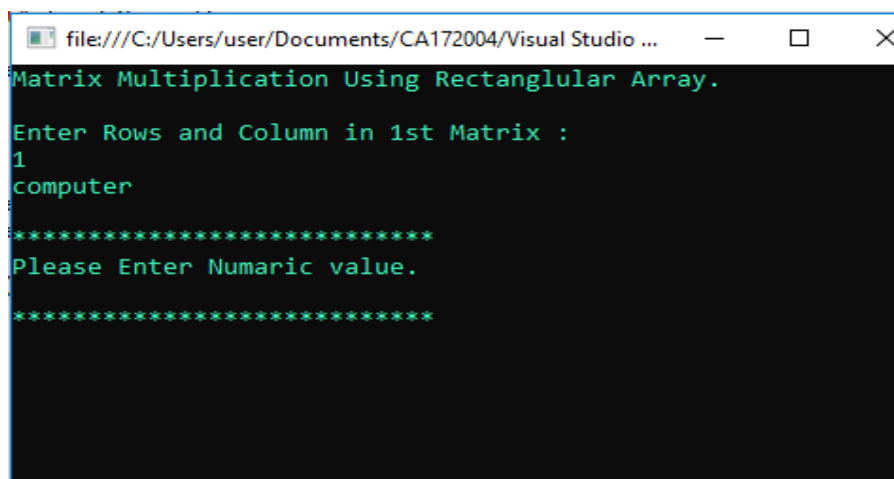
A screenshot of a C# console application window. The title bar shows the file path: file:///C:/Users/user/Documents/CA172004/Vis... The console output is as follows:

```
First Matrix
    5    6
    7    8

Second Matrix
    5    6
    7    8

Multiplication of Matrix

    67    78
    91   106
```



A screenshot of a C# console application window. The title bar shows the file path: file:///C:/Users/user/Documents/CA172004/Visual Studio ... The console output is as follows:

```
Matrix Multiplication Using Rectangular Array.

Enter Rows and Column in 1st Matrix :
1
computer

*****
Please Enter Numeric value.
*****
```

**13) Demonstrate Use of Virtual and override keyword in C# with a simple Program.**

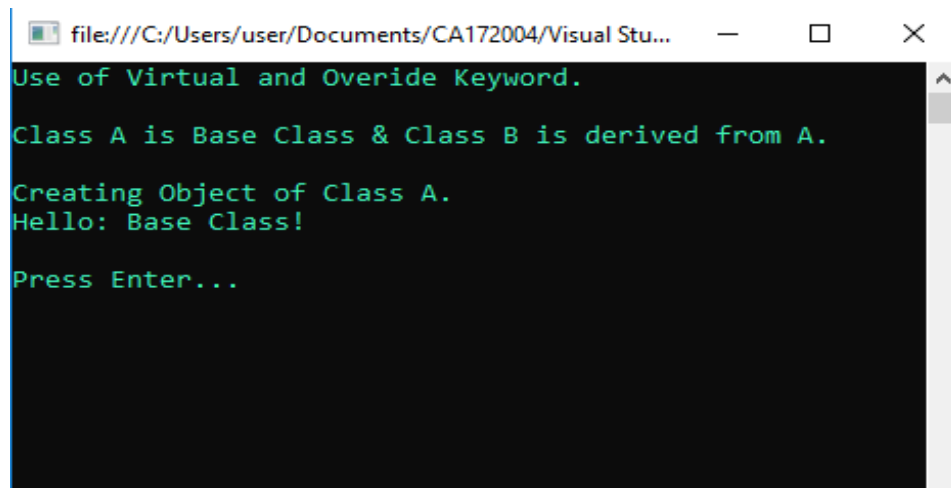
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication19
{
    class Program
    {
        class A
        {
            public virtual void show()
            {
                Console.WriteLine("Hello: Base Class!");
                Console.Write("\nPress Enter...");
                Console.ReadLine();
            }
        }

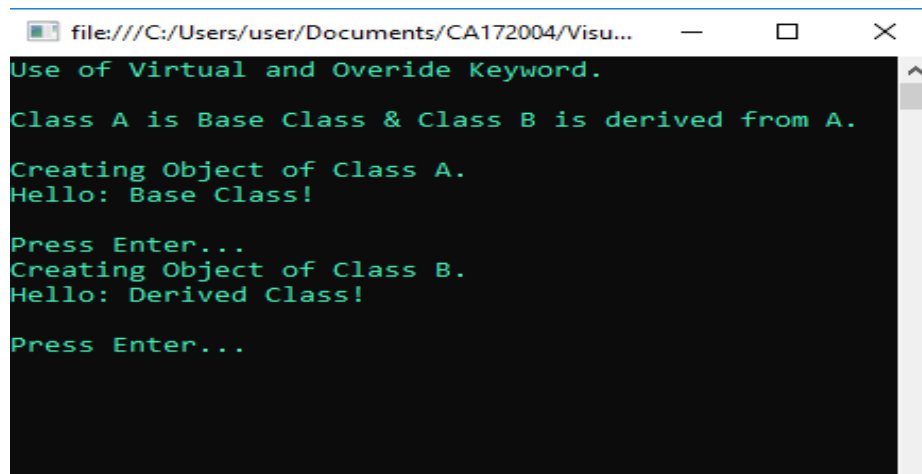
        class B : A
        {
            public override void show()
            {
                Console.WriteLine("Hello: Derived Class!");
                Console.Write("\nPress Enter...");
                Console.ReadLine();
            }
        }
    }

    static void Main(string[] args)
    {
        Console.WriteLine("Use of Virtual and Override Keyword.");
        Console.WriteLine("\nClass A is Base Class & Class B is derived from A.\n");
        Console.WriteLine("Creating Object of Class A.");
        A a1 = new A();
        a1.show();
        Console.WriteLine("Creating Object of Class B.");
        B b1 = new B();
        b1.show();
        Console.WriteLine("Creating Object of Class A & Calling Method of Class B.");
        A a2 = new B();
        a2.show();
        Console.ReadKey();
    }
}
```

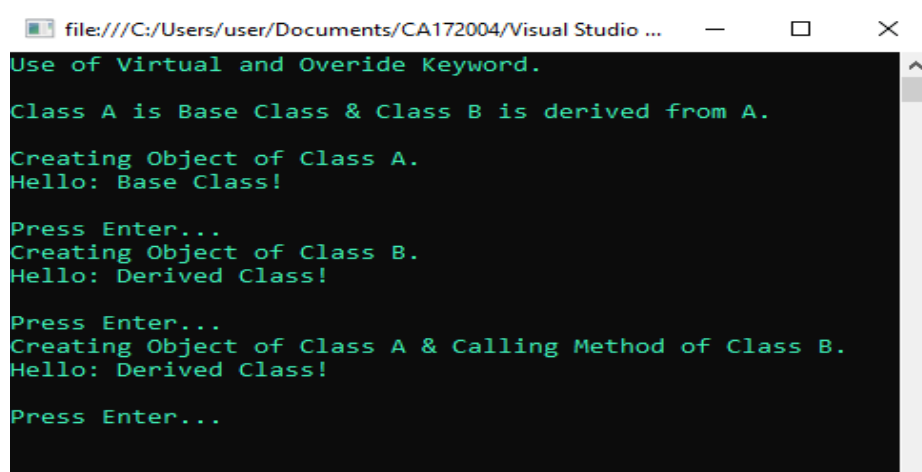
## OUTPUT



```
file:///C:/Users/user/Documents/CA172004/Visual Stu...
Use of Virtual and Override Keyword.
Class A is Base Class & Class B is derived from A.
Creating Object of Class A.
Hello: Base Class!
Press Enter...
```



```
file:///C:/Users/user/Documents/CA172004/Visu...
Use of Virtual and Override Keyword.
Class A is Base Class & Class B is derived from A.
Creating Object of Class A.
Hello: Base Class!
Press Enter...
Creating Object of Class B.
Hello: Derived Class!
Press Enter...
```



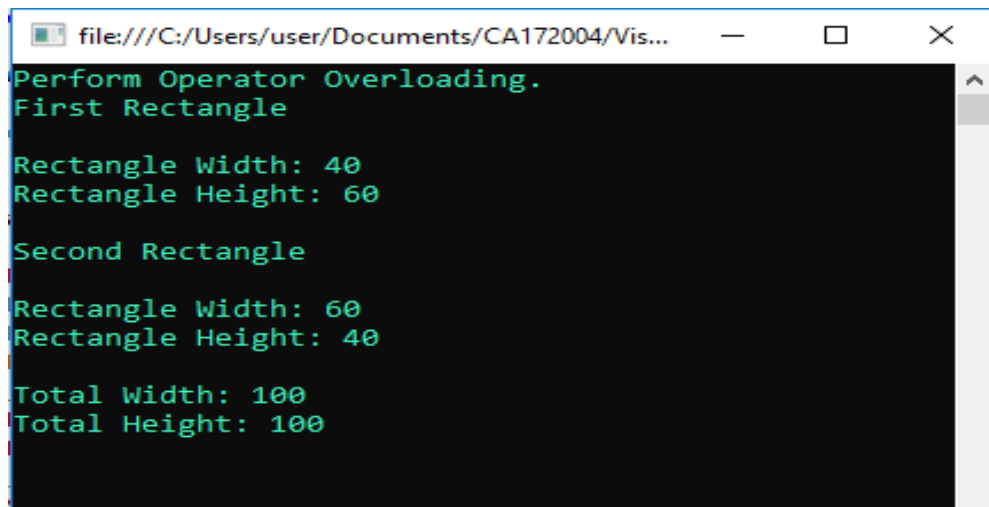
```
file:///C:/Users/user/Documents/CA172004/Visual Studio ...
Use of Virtual and Override Keyword.
Class A is Base Class & Class B is derived from A.
Creating Object of Class A.
Hello: Base Class!
Press Enter...
Creating Object of Class B.
Hello: Derived Class!
Press Enter...
Creating Object of Class A & Calling Method of Class B.
Hello: Derived Class!
Press Enter...
```

**14) Perform operator overloading.**

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

namespace ConsoleApplication18
{
    class Program
    {
        class Rectangle
        {
            int width;
            int height;
            Rectangle(int width, int height)
            {
                this.width = width;
                this.height = height;
            }
            public static Rectangle operator +(Rectangle a, Rectangle b)
            {
                int totalWidth = a.width + b.width;
                int totalHeight = a.height + b.height;
                return new Rectangle(totalWidth, totalHeight);
            }
            static void Main(string[] args)
            {
                Console.WriteLine("Perform Operator Overloading.");
                Rectangle r1 = new Rectangle(40, 60);
                Rectangle r2 = new Rectangle(60, 40);
                Console.WriteLine("First Rectangle");
                Console.WriteLine();
                Console.WriteLine("Rectangle Width: {0}", r1.width);
                Console.WriteLine("Rectangle Height: {0}", r1.height);
                Console.WriteLine();
                Console.WriteLine("Second Rectangle");
                Console.WriteLine("");
                Console.WriteLine("Rectangle Width: {0}", r2.width);
                Console.WriteLine("Rectangle Height: {0}", r2.height);
                Console.WriteLine();
                Rectangle r3 = r1 + r2;
                Console.WriteLine("Total Width: {0}", r3.width);
                Console.WriteLine("Total Height: {0}", r3.height);
                Console.ReadKey();
            }
        }
    }
}
```

## OUTPUT



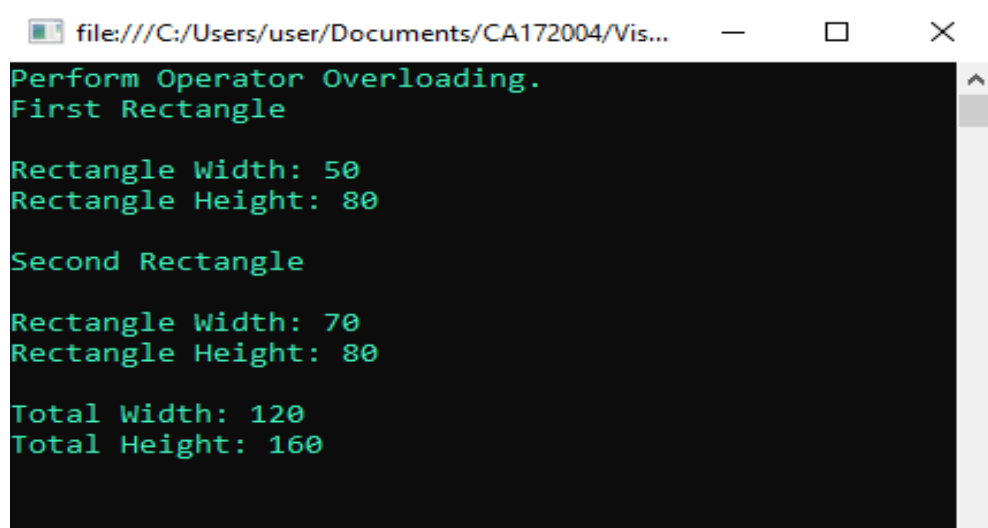
```
file:///C:/Users/user/Documents/CA172004/Vis...
Perform Operator Overloading.
First Rectangle

Rectangle Width: 40
Rectangle Height: 60

Second Rectangle

Rectangle Width: 60
Rectangle Height: 40

Total Width: 100
Total Height: 100
```



```
file:///C:/Users/user/Documents/CA172004/Vis...
Perform Operator Overloading.
First Rectangle

Rectangle Width: 50
Rectangle Height: 80

Second Rectangle

Rectangle Width: 70
Rectangle Height: 80

Total Width: 120
Total Height: 160
```

- 15) Create classes, they are reference types in C# and hence are allocated on the heap. Classes provide object-oriented constructs such as encapsulation, polymorphism, and inheritance. For instance, the program should print John. Doe twice, illustrating that objects are reference types, allocated on the heap implement the same using C#.

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

namespace ProgramTwo
{
    class Program
    {
        static void Main(string[] args)
        {
            User user1 = new User("Tony");
            Admin user2 = new Admin("Priyadarshini", "Priyadarshini@gmail.com", "Patil");

            Console.WriteLine("User 1:");
            Console.WriteLine("Name: {0}", user1.GetName());
            Console.WriteLine("Email: {0}", user1.getEmail());

            Console.WriteLine();

            Console.WriteLine("User 2 (Admin):");
            Console.WriteLine("Name: {0}", user2.GetName());
            Console.WriteLine("Email: {0}", user2.getEmail());
            Console.WriteLine("Password: {0}", user2.getPassword());

            Console.Read();
        }
    }

    class User {
        private string name;
        private string email;

        public User(String name) {
            this.name = name;
        }

        public User(String name, String email)
        {
            this.name = name;
        }
    }
}
```

```
        this.email = email;
    }

    public string getName() {
        return name;
    }

    public string getEmail()
    {
        return email;
    }

    public void setName(string name)
    {
        this.name = name;
    }

    public void setEmail(string email)
    {
        this.email = email;
    }
}

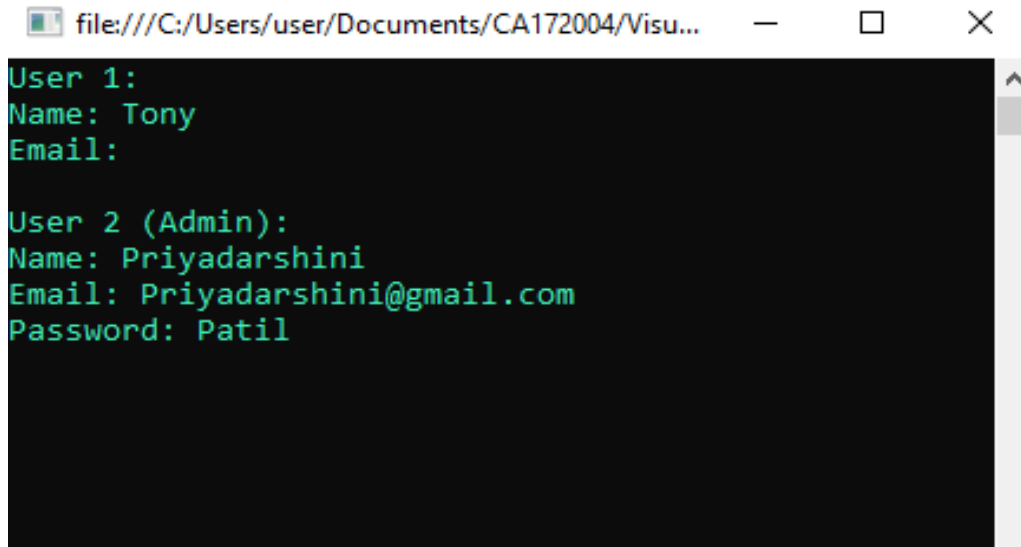
class Admin : User {
    private string password;
    public Admin(string name, string email, string password): base(name, email)
    {
        this.password = password;
    }

    public void setPassword(string password) {
        this.password = password;
    }

    public string getPassword() {
        return password;
    }
}
}
```



## OUTPUT

A screenshot of a Windows file explorer window. The title bar shows the file path: file:///C:/Users/user/Documents/CA172004/Visu... with standard minimize, maximize, and close buttons. The main content area is a black text file with green text. It contains two user entries. The first entry is for 'User 1' with Name 'Tony' and an empty Email field. The second entry is for 'User 2 (Admin)' with Name 'Priyadarshini', Email 'Priyadarshini@gmail.com', and Password 'Patil'.

```
User 1:  
Name: Tony  
Email:  
  
User 2 (Admin):  
Name: Priyadarshini  
Email: Priyadarshini@gmail.com  
Password: Patil
```

**16) Work with Page using ASP.Net.****FileName: Global.asax**

```
<%@ Application Language="C#"%>
<script runat="server">
void Application_Start(object sender, EventArgs e)
{
// Code that runs on application startup
Application["NumberOfVisitors"] = 0;

}
void Session_Start(object sender, EventArgs e)
{
// Code that runs when a new session is started
Application.Lock();
Application["NumberOfVisitors"] = (int)Application["NumberOfVisitors"] + 1;
Application.Unlock();

}
</script>
```

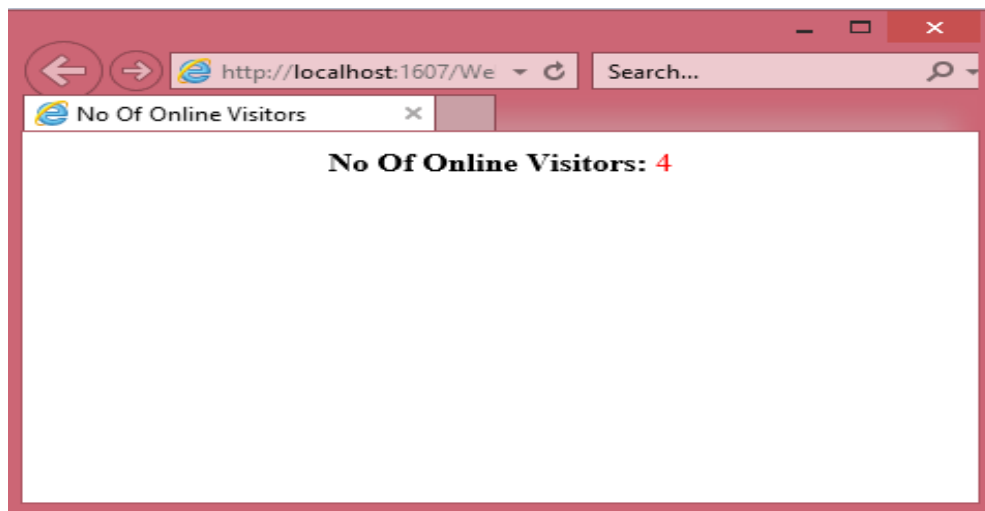
**FileName: Default.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>No Of Online Visitors</title>
</head>
<body>
<form id="form1" runat="server">
<center>
<table>
<tr>
<td><b>No Of Online Visitors:</b></td>
<td><asp:Label ID="lblCount" runat="server" ForeColor="Red" /></td>
</tr>
</table>
</center>
</form>
</body>
</html>
```

**FileName: Default.aspx.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        lblCount.Text = Application["NumberOfVisitors"].ToString();
    }
}
```

**OUTPUT**

**17) Describe access data source through ADO.NET.****Form1.cs**

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Windows.Forms;

namespace ProgramEleven
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void btnFetch_Click(object sender, EventArgs e)
            {
                UserAccessLayer uAL = new UserAccessLayer();
                List<User> users = uAL.getAllUsers();
                if(users.Count == 0)
                    lblStatus.Text = "No data!";
                else
                    lblStatus.Text = "Data Fetched!";

                dGV.DataSource = users;

            }
        }
    }
}
```

**Users.cs**

using System;

namespace ProgramEleven

{

class User

{

public int Id

{

get;

set;

}

public string UserName

{

get;

set;

}

public string RollNumber

{

get;

set;

}

public string Email

{

get;

set;

}

}

}

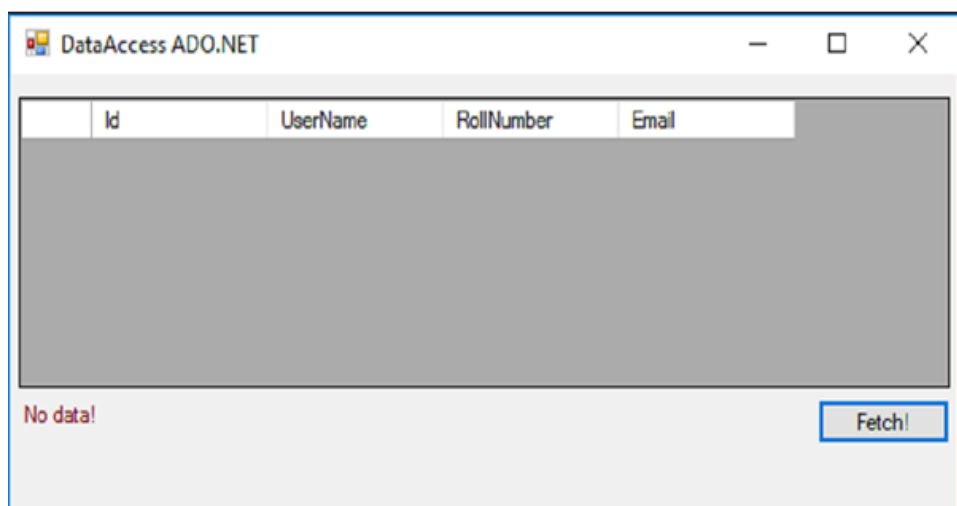
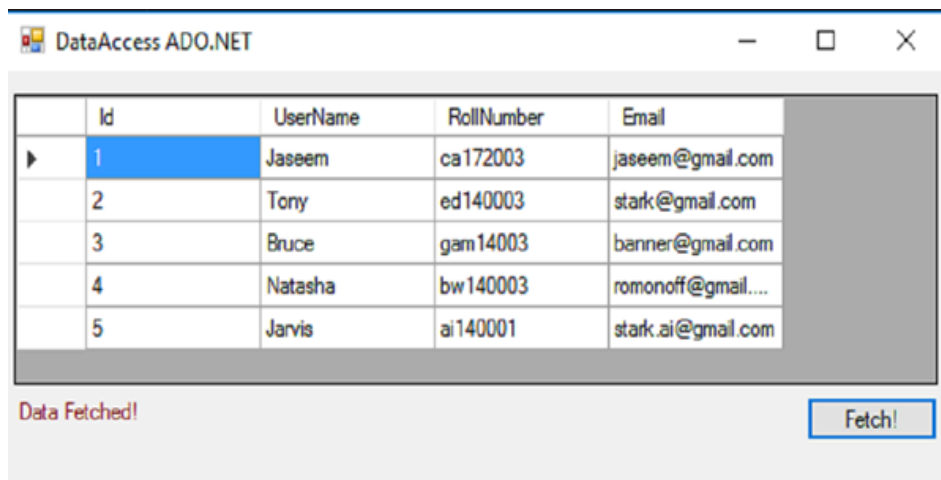
**UserAccessLayer.cs**

```
using System;
using System.Data;
using System.Data.SqlClient;

namespace ProgramEleven
{
    class UserAccessLayer
    {
        private List<User> users;
        private string connectionString = @"Data Source=.\SQLEXPRESS/PSELF;Initial
        Catalog=TestDB; Integrated Security=True";
        private SqlConnection connection;
        private SqlCommand command;
        private string query;

        public List<User> getAllUsers()
        {
            users = new List<User>();
            try
            {
                connection = new SqlConnection(connectionString);
                connection.Open();
                query = "SELECT * FROM user";
                command = new SqlCommand(query, connection);
                SqlDataReader reader = command.ExecuteReader();
                while (reader.Read())
                {
                    User user = new User();
                    user.Id = Convert.ToInt16(reader.GetValue(0));
                    user.UserName = reader.GetValue(1).ToString();
                    user.Email = reader.GetValue(2).ToString();
                    user.RollNumber = reader.GetValue(3).ToString();
                    users.Add(user);
                }
            }
            catch (SqlException ex)
            {
                Console.WriteLine("Error in fetching database!: " + ex.Message);
            }
            return users;
        }
    }
}
```

## OUTPUT





**17)Work with forms using ASP.NET.**

using System;

namespace WindowsFormsApplication1

{

public partial class Form1 : Form

{

string[] names;

string[] passs;

int rows;

public Form1()

{

InitializeComponent();

names = new string[10];

passs = new string[10];

names[0] = "admin";

names[1] = "user";

names[2] = "tony";

passs[0] = "admin";

passs[1] = "user";

passs[2] = "stark";

rows = 3;

}

private void button1\_Click(object sender, EventArgs e)

```
{  
    string username = textBox1.Text.Trim();  
    string password = textBox2.Text.Trim();  
  
    if (username.Equals("") || password.Equals(""))  
    {  
        MessageBox.Show("Fields cannot be empty!");  
        return;  
    }  
    for (int i = 0; i < rows; i++)  
    {  
        if (names[i].Equals(username) && passs[i].Equals(password))  
        {  
            MessageBox.Show("Login Successfull!");  
            return;  
        }  
    }  
    MessageBox.Show("Incorrect username/password!");  
}  
}  
}
```

## OUTPUT



Working with Forms

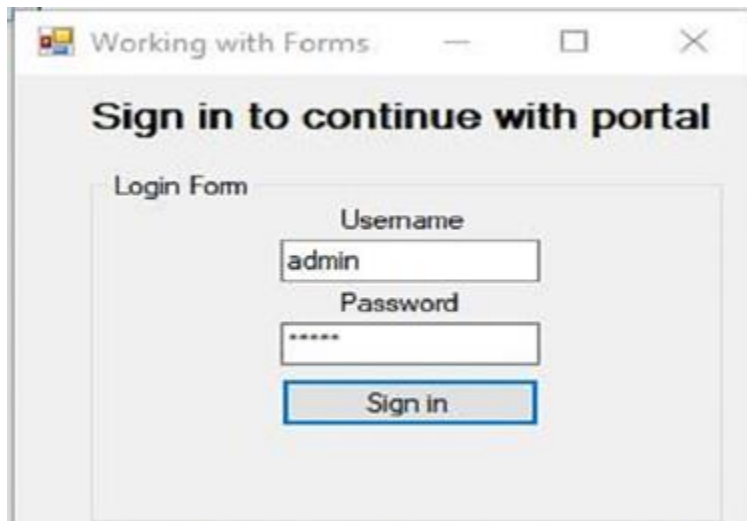
**Sign in to continue with portal**

Login Form

Username

Password

Sign in



Working with Forms

**Sign in to continue with portal**

Login Form

Username

admin

Password

\*\*\*\*\*

Sign in



Login Successfull!

OK



Working with Forms

**Sign in to continue with portal**

Login Form

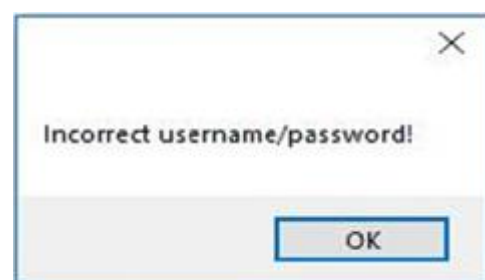
Username

admin

Password

\*\*\*\*\*

Sign in



Incorrect username/password!

OK