208W1A12A0

Velagapudi Ramakrishna Siddhartha Engineering College

Program Elective - 1 Course

Dot Net Technologies Lab

Code : 20IT5404B

Week – 1

Aim : Create C# class Employee with the fields eno, ename, address, designation, mobile number, salary, city, pincode. Create necessary methods to read the information and display the information. The employer contains 10 employees. The employer wants to know the total salary paid to its employees. As a programmer how do you suggest a solution to his problem?

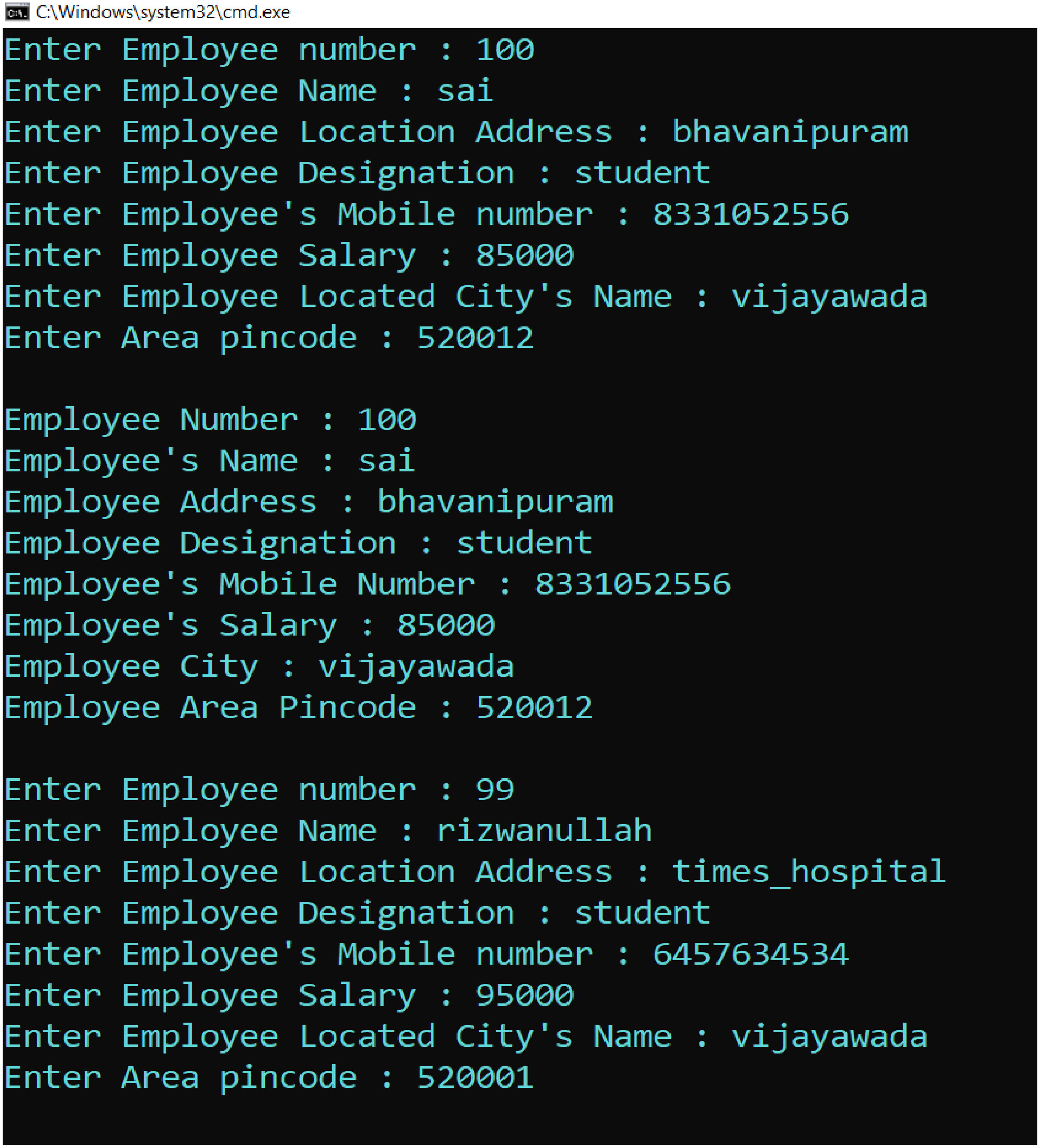
Program :

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;


namespace Employee_1

{

    public class Program

    {

        public decimal total = 0;

        public static void Main(string[] args)

        {
            decimal T = 0;
            Employee[] obj = new Employee[10];


            for(int i = 0; i < 3; i++)
            {
                obj[i] = new Employee();
                obj[i].SetStudentDetails();
                Console.WriteLine();
                obj[i].GetStudentDetails();


                decimal t = obj[i].salary;
                T = T + t;
            }


            Console.WriteLine("Total Salary of all Employee's is : " + T);


        }
    }


    class Employee
    {
```

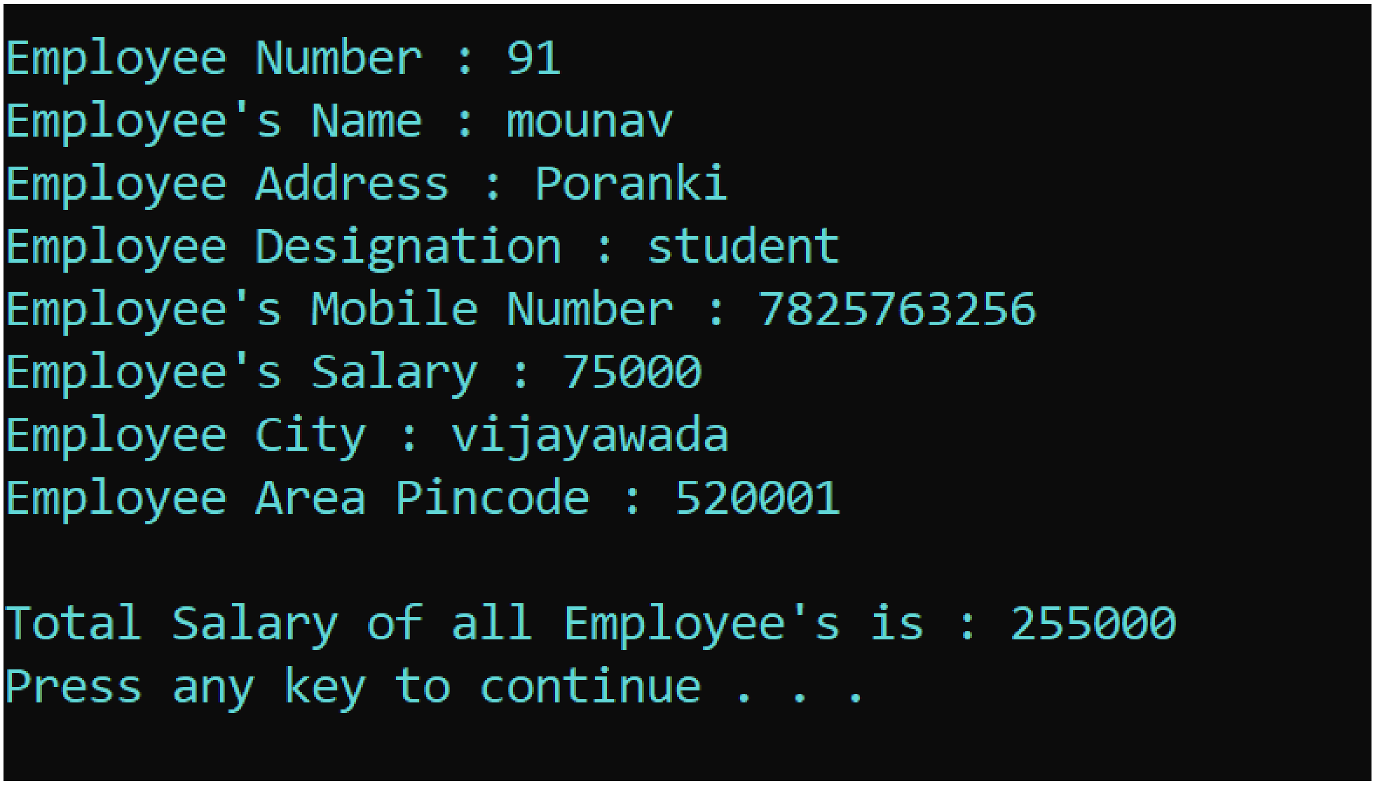
```
public int eno, pincode;

public string ename, address, designation, city;

public decimal salary;

public double mobile;


public void SetStudentDetails()

{

    Console.Write("Enter Employee number : ");

    eno = Convert.ToInt32(Console.ReadLine());


    Console.Write("Enter Employee Name : ");

    ename = Console.ReadLine();


    Console.Write("Enter Employee Location Address : ");

    address = Console.ReadLine();


    Console.Write("Enter Employee Designation : ");

    designation = Console.ReadLine();


    Console.Write("Enter Employee's Mobile number : ");

    mobile = Convert.ToDouble(Console.ReadLine());
```

```
            Console.Write("Enter Employee Salary : ");

            salary = Convert.ToDecimal(Console.ReadLine());

            Console.Write("Enter Employee Located City's Name : ");

            city = Console.ReadLine();

            Console.Write("Enter Area pincode : ");

            pincode = Convert.ToInt32(Console.ReadLine());

        }

        public void GetStudentDetails()

        {

            Console.WriteLine("Employee Number : " + eno);

            Console.WriteLine("Employee's Name : " + ename);

            Console.WriteLine("Employee Address : " + address);

            Console.WriteLine("Employee Designation : " + designation);

            Console.WriteLine("Employee's Mobile Number : " + mobile);

            Console.WriteLine("Employee's Salary : " + salary);

            Console.WriteLine("Employee City : " + city);

            Console.WriteLine("Employee Area Pincode : " + pincode);

        }

    }

}
```

Output :



```
Enter Employee number : 100
Enter Employee Name : sai
Enter Employee Location Address : bhavanipuram
Enter Employee Designation : student
Enter Employee's Mobile number : 8331052556
Enter Employee Salary : 85000
Enter Employee Located City's Name : vijayawada
Enter Area pincode : 520012

Employee Number : 100
Employee's Name : sai
Employee Address : bhavanipuram
Employee Designation : student
Employee's Mobile Number : 8331052556
Employee's Salary : 85000
Employee City : vijayawada
Employee Area Pincode : 520012

Enter Employee number : 99
Enter Employee Name : rizwanullah
Enter Employee Location Address : times_hospital
Enter Employee Designation : student
Enter Employee's Mobile number : 6457634534
Enter Employee Salary : 95000
Enter Employee Located City's Name : vijayawada
Enter Area pincode : 520001
```



```
Employee Number : 91
Employee's Name : mounav
Employee Address : Poranki
Employee Designation : student
Employee's Mobile Number : 7825763256
Employee's Salary : 75000
Employee City : vijayawada
Employee Area Pincode : 520001

Total Salary of all Employee's is : 255000
Press any key to continue . . .
```

Result : Sucessfully Executed the program.

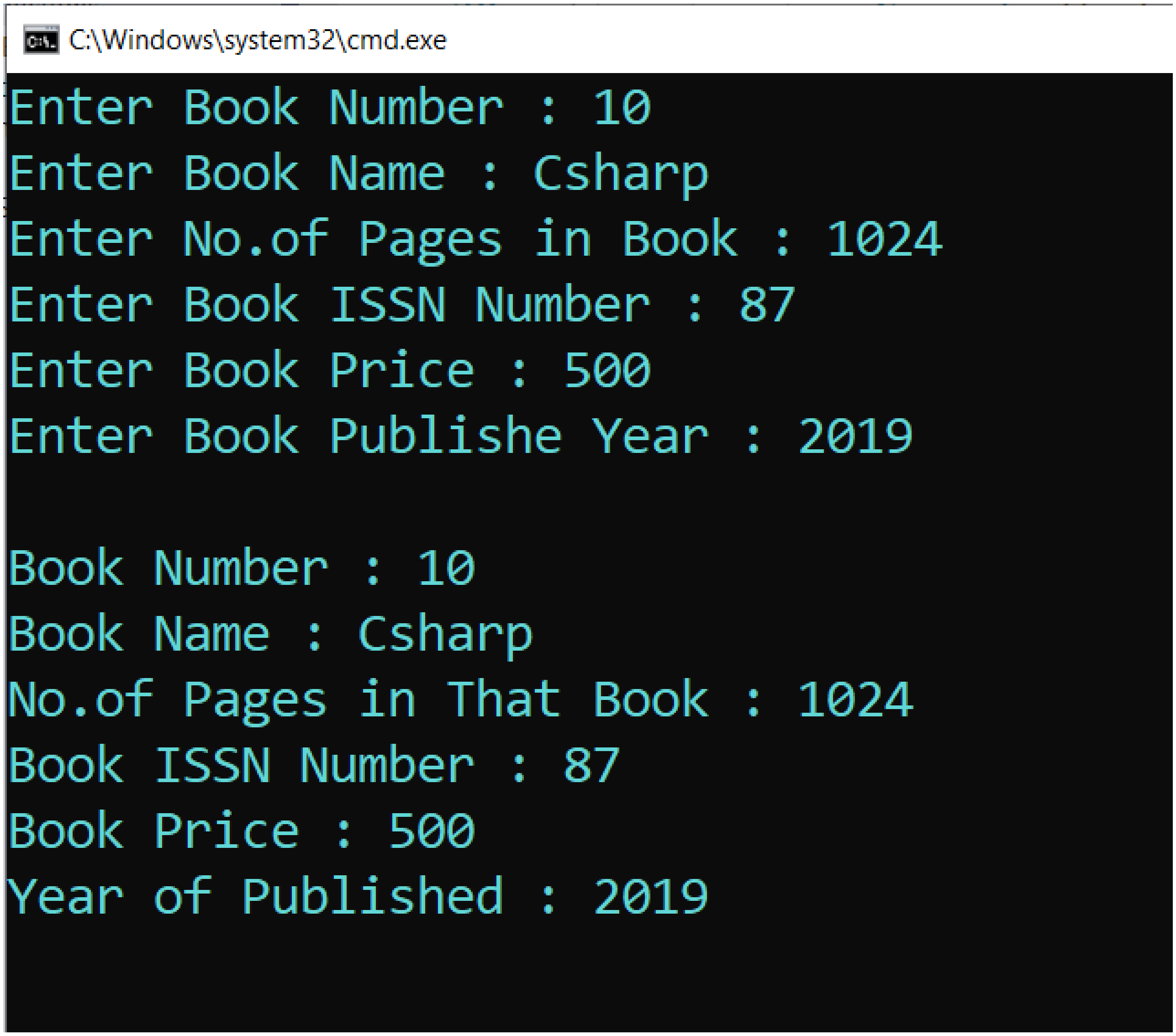
Aim : Create a C# class to read the book information to a library which includes booknumber, name, pages, issn number, price, year_of_publication. Use C# Properties to store the information into these variables and retrieve the information.

Program :

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Properties_1

{

    internal class Program

    {

        public static void Main(string[] args)

        {

            Person obj = new Person();

            Console.Write("Enter Book Number : ");

            obj.book_no = Convert.ToInt32(Console.ReadLine());
            Console.Write("Enter Book Name : ");

            obj.name = Console.ReadLine();

            Console.Write("Enter No.of Pages in Book : ");

            obj.pages = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Book ISSN Number : ");

            obj.issn = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Book Price : ");

            obj.price = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Book Publishe Year : ");

            obj.pub = Convert.ToInt32(Console.ReadLine());


            Console.WriteLine();


            Console.WriteLine("Book Number : " + obj.book_no);

            Console.WriteLine("Book Name : " + obj.name);

            Console.WriteLine("No.of Pages in That Book : " + obj.pages);

            Console.WriteLine("Book ISSN Number : " + obj.issn);

            Console.WriteLine("Book Price : " + obj.price);

            Console.WriteLine("Year of Published : " + obj.pub);

        }

    }
```

```
    class Person
    {
        public int book_no { get; set; }
        public string name { get; set; }
        public int pages { get; set; }
        public int issn { get; set; }
        public int price { get; set; }
        public int pub { get; set; }
    }
}
```

Output :



C:\Windows\system32\cmd.exe

```
Enter Book Number : 10
Enter Book Name : Csharp
Enter No.of Pages in Book : 1024
Enter Book ISSN Number : 87
Enter Book Price : 500
Enter Book Publishe Year : 2019

Book Number : 10
Book Name : Csharp
No.of Pages in That Book : 1024
Book ISSN Number : 87
Book Price : 500
Year of Published : 2019
```

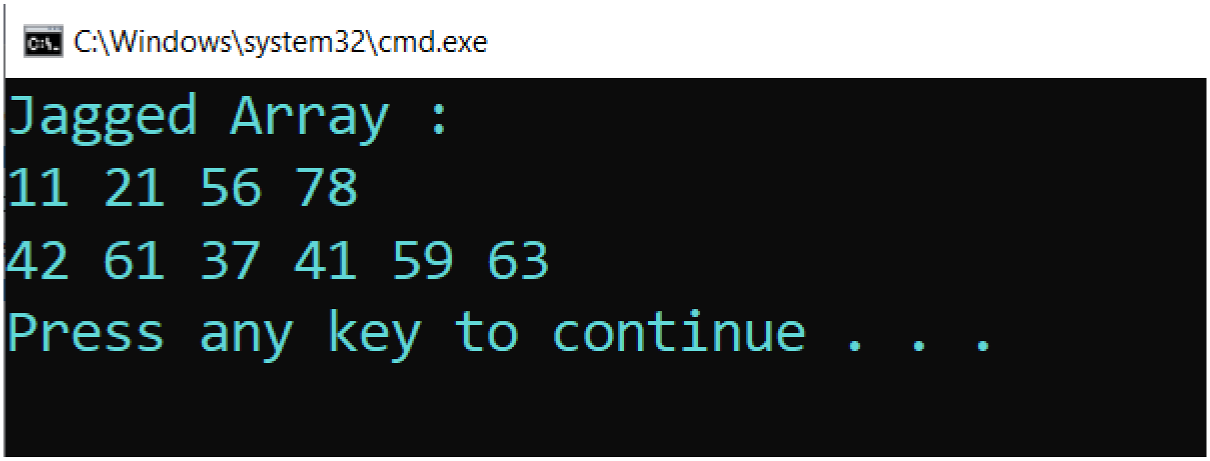
Result : Sucessfully Executed the Program.

Aim : Write a C# program of jagged array which declares, initializes and traverse jagged arrays.

Program :

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Jagged_Arrays

{

    internal class Program

    {

        static void Main(string[] args)

        {

            int[][] arr = new int[2][]; // Declare the array

            arr[0] = new int[] { 11, 21, 56, 78 };// Initialize the array

            arr[1] = new int[] { 42, 61, 37, 41, 59, 63 };

            Console.WriteLine("Jagged Array : ");

            // Traverse array elements

            for (int i = 0; i < arr.Length; i++)

            {
                for (int j = 0; j < arr[i].Length; j++)

                {

                    System.Console.Write(arr[i][j] + " ");

                }

                System.Console.WriteLine();

            }

        }

    }

}
```

Output :



```
Jagged Array :
11 21 56 78
42 61 37 41 59 63
Press any key to continue . . .
```

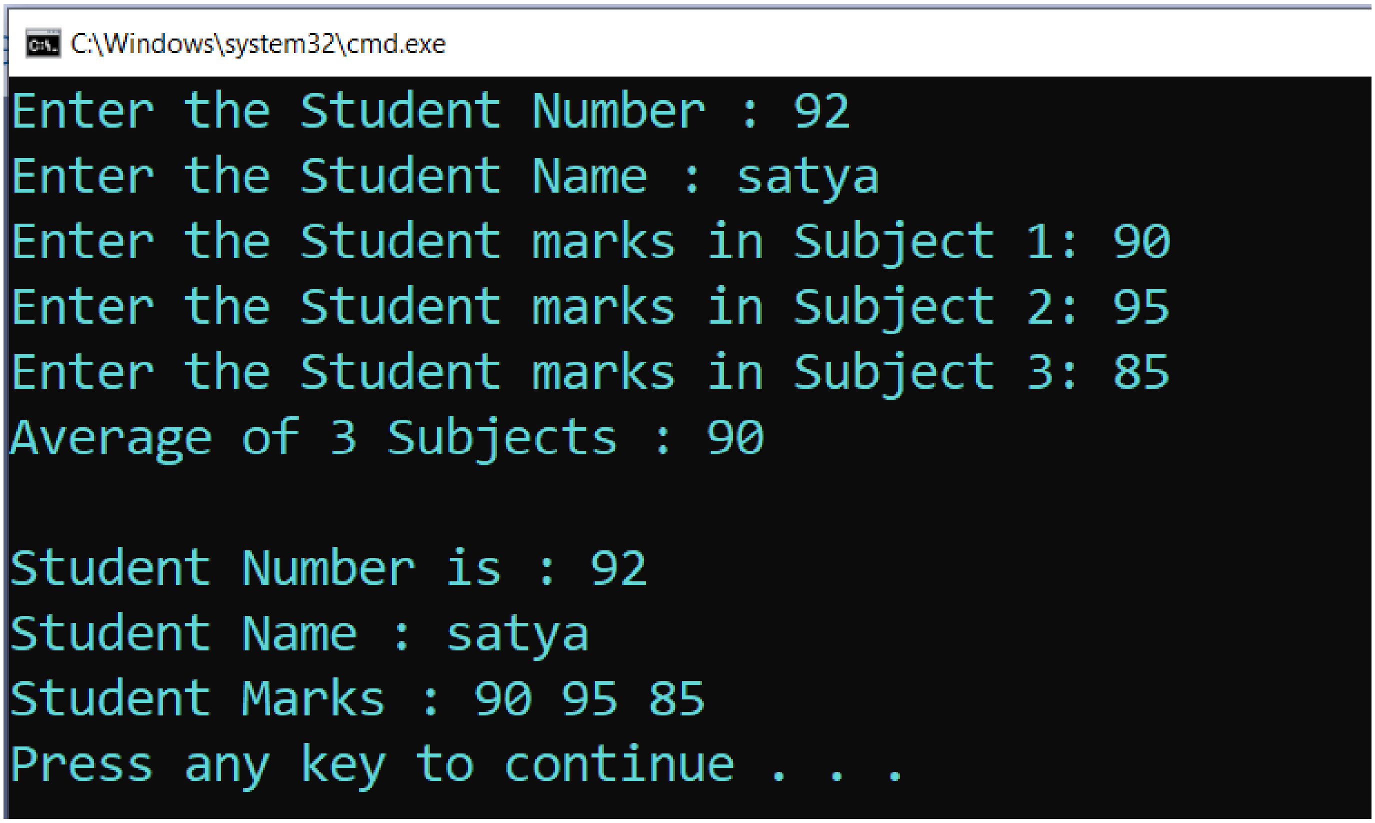
Result : Sucessfully Executed the program.

Aim : Create a class Student with the filed sno, sname, marks in three subjects. Create a method SetStudentDetails() which will read the information from the student. Create another method GetStudentDetails() which will display the information. Also, another method to compute the average mark of the student in three subject. Display the information.

Program :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Student_Class
{
    public class Program
    {
        static void Main(string[] args)
        {
            Student obj = new Student();
            obj.SetStudentDetails();
            obj.average();
            Console.WriteLine();
            obj.GetStudentDetails();
        }
    }
    class Student
    {
        public int sno;
        public string sname;
        public Decimal m1, m2, m3;
        public void SetStudentDetails()
        {
            Console.Write("Enter the Student Number : ");
            sno = Convert.ToInt32(Console.ReadLine());
            Console.Write("Enter the Student Name : ");
            sname = Console.ReadLine();
            Console.Write("Enter the Student marks in Subject 1: ");
            m1 = Convert.ToDecimal(Console.ReadLine());
            Console.Write("Enter the Student marks in Subject 2: ");
            m2 = Convert.ToDecimal(Console.ReadLine());
            Console.Write("Enter the Student marks in Subject 3: ");
            m3 = Convert.ToDecimal(Console.ReadLine());
        }
        public void GetStudentDetails()

        {

            Console.WriteLine("Student Number is : " + sno);

            Console.WriteLine("Student Name : " + sname);

            Console.WriteLine("Student Marks : " + m1 + " " + m2 + " " + m3);

        }

        public void average()

        {

            Console.WriteLine("Average of 3 Subjects : " + ((m1 + m2 + m3) / 3));

        }

    }

}
```

Output :



```
Enter the Student Number : 92
Enter the Student Name : satya
Enter the Student marks in Subject 1: 90
Enter the Student marks in Subject 2: 95
Enter the Student marks in Subject 3: 85
Average of 3 Subjects : 90

Student Number is : 92
Student Name : satya
Student Marks : 90 95 85
Press any key to continue . . .
```

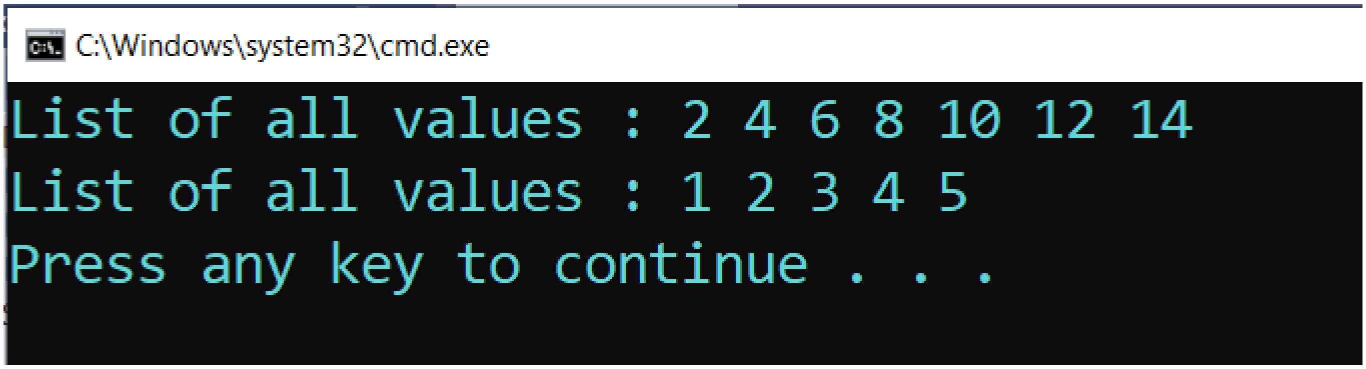
Result : Sucessfully Executed the Program .

Aim : Write a C# program which takes variable number of parameters i.e using params.

Program :

```
using System;
namespace Params_Keyword
{
    internal class Program
    {
        public void Show(params int[] val) // Params Paramater
        {
            Console.Write("List of all values : ");
            for (int i = 0; i < val.Length; i++)
            {
                Console.Write(val[i] + " ");
            }
            Console.ReadLine();
        }
        static void Main(string[] args)
        {
            Program p = new Program(); // Creating Object
            p.Show(2, 4, 6, 8, 10, 12, 14); // Passing arguments of variable length
            p.Show(1, 2, 3, 4, 5);
        }
    }
}
```

Output :



```
List of all values : 2 4 6 8 10 12 14
List of all values : 1 2 3 4 5
Press any key to continue . . .
```

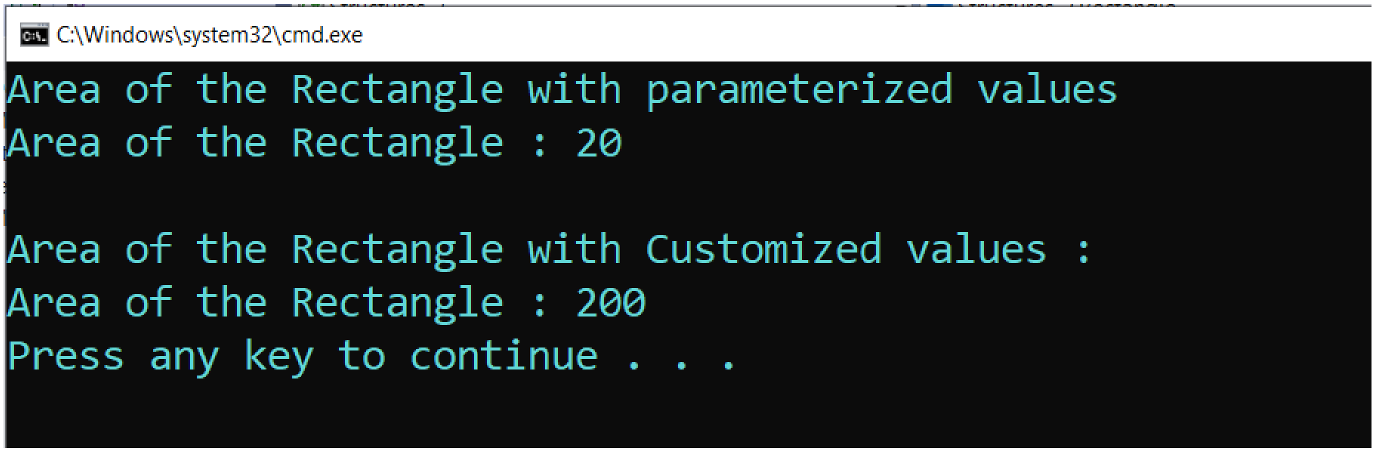
Result : Sucessfully executed the program.

Aim : Write a C# program to create a struct Rectangle which has two data members width and height, where we are using constructor to initialize data and method to calculate area of rectangle.

Program :

```
using System;

namespace Structures_2

{

    internal class Program

    {

        public static void Main(string[] args)

        {

            Rectangle r = new Rectangle(4,5);

            Console.WriteLine("Area of the Rectangle with parameterized values");

            r.Area_Rectangle();

            Console.WriteLine();

            Console.WriteLine("Area of the Rectangle with Customized values : ");

            r.width = 10;

            r.height = 20;

            r.Area_Rectangle();
        }
    }
    public struct Rectangle
    {
        public int width, height;
        public Rectangle(int w, int h)
        {
            this.width = w;
            this.height = h;
        }
        public void Area_Rectangle()
        {
            Console.WriteLine("Area of the Rectangle : " + (width * height));
        }
    }
}
```

Output :



```
Area of the Rectangle with parameterized values
Area of the Rectangle : 20

Area of the Rectangle with Customized values :
Area of the Rectangle : 200
Press any key to continue . . .
```

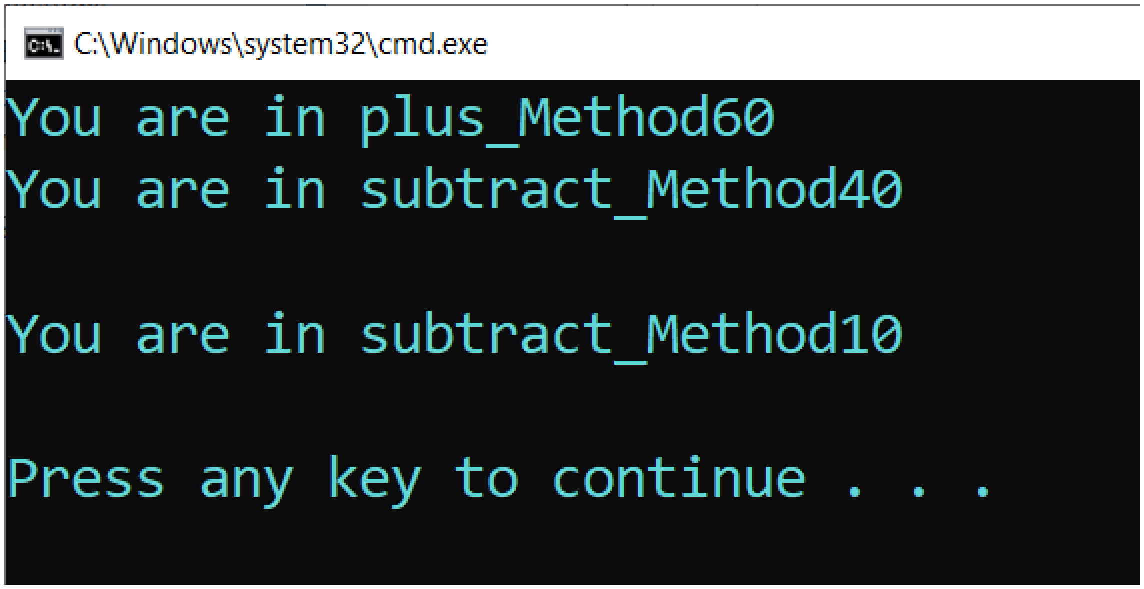
Result : Sucessfully Executed the Program.

Aim : Write a C# program on Multicast Delegates.

Program :

```
using System;
namespace Multicast_Delegates
{
    internal class Program
    {
        public delegate void delmethod(int x, int y);
        public void plus_Method1(int x, int y)
        {
            Console.Write("You are in plus_Method");
            Console.WriteLine(x + y);
        }

        public void subtract_Method2(int x, int y)
        {
            Console.Write("You are in subtract_Method");
            Console.WriteLine(x - y);
        }
        static void Main(string[] args)

        {

            Program obj = new Program();

            delmethod del = new delmethod(obj.plus_Method1);

            // Here we have multicast

            del += new delmethod(obj.subtract_Method2);

            // plus_Method1 and subtract_Method2 are called

            del(50, 10);

            Console.WriteLine();

            //Here again we have multicast

            del -= new delmethod(obj.plus_Method1);

            //Only subtract_Method2 is called

            del(20, 10);

            Console.ReadLine();

        }

    }

}
```

Output :



```
You are in plus_Method60
You are in subtract_Method40

You are in subtract_Method10

Press any key to continue . . .
```

Result : Sucessfully Executed the program.

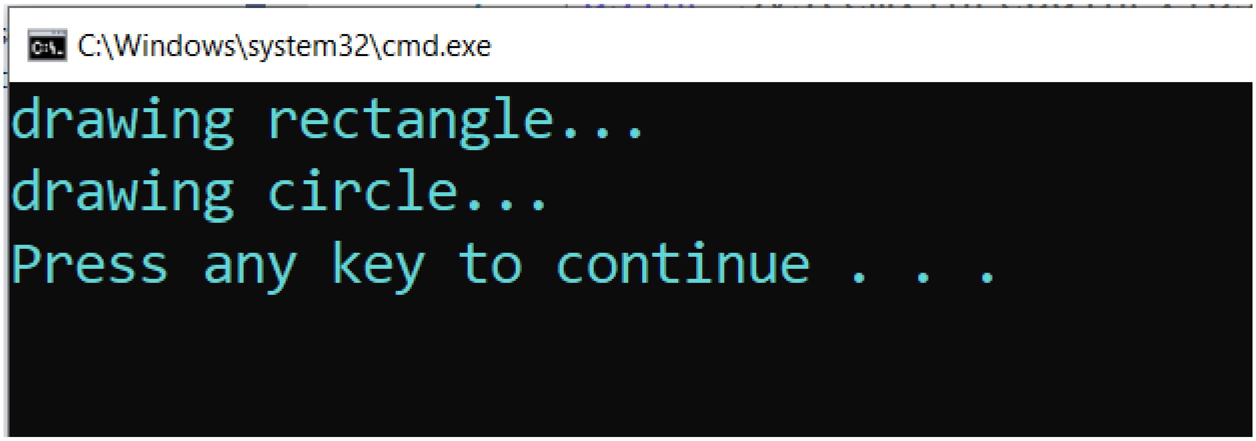
Aim : Write a C# program to create an interface Drawable which has draw() method. Its implementation is provided by two classes: Rectangle and Circle.

Program :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Interfaces
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Drawable d;
            d = new Rectangle();
            d.draw();
            d = new Circle();
            d.draw();
        }
    }
    public interface Drawable
    {
        void draw();
    }
    public class Rectangle : Drawable
    {
        public void draw()
        {
            Console.WriteLine("drawing rectangle...");
        }
    }
    public class Circle : Drawable
    {
        public void draw()
        {
            Console.WriteLine("drawing circle...");
        }
    }
}
```

208W1A12A0

Output :



C:\Windows\system32\cmd.exe

```
drawing rectangle...
drawing circle...
Press any key to continue . . .
```

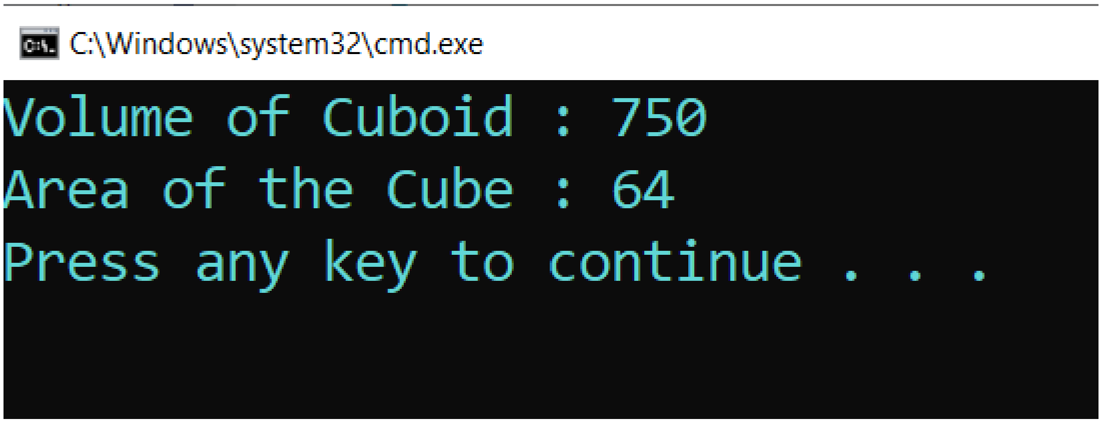
Result : Sucessfully Executed the Program.

Aim : Create a Class Polygon with three variables l,b,h. Create a parameterised method to set the values to l,b,h. Create an abstract method compute().

Program :

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

namespace Abstract_Class
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Cuboid c1 = new Cuboid(5, 10, 15);
            c1.Compute();

            Cube c2 = new Cube(2, 8, 4);
            c2.Compute();
        }
    }

    abstract class Polygon
    {
        public int l, b, h;
        public Polygon(int l, int b, int h)
        {
            this.l = l;
            this.b = b;
            this.h = h;
        }

        public abstract void Compute();
    }

    class Cuboid : Polygon
    {
        public Cuboid(int l, int b, int h) : base(l, b, h)
        {
            //base(l, b, h);
        }

        public override void Compute()
        {
            Console.WriteLine("Volume of Cuboid : " + (l * b * h));
        }
    }

    class Cube : Polygon
    {
        public Cube(int l, int b, int h) : base(l, b, h)
        {
            //base(l, b, h);
        }
        public override void Compute()
        {
            Console.WriteLine("Area of the Cube : " + (l * b * h));
        }
    }
}
```

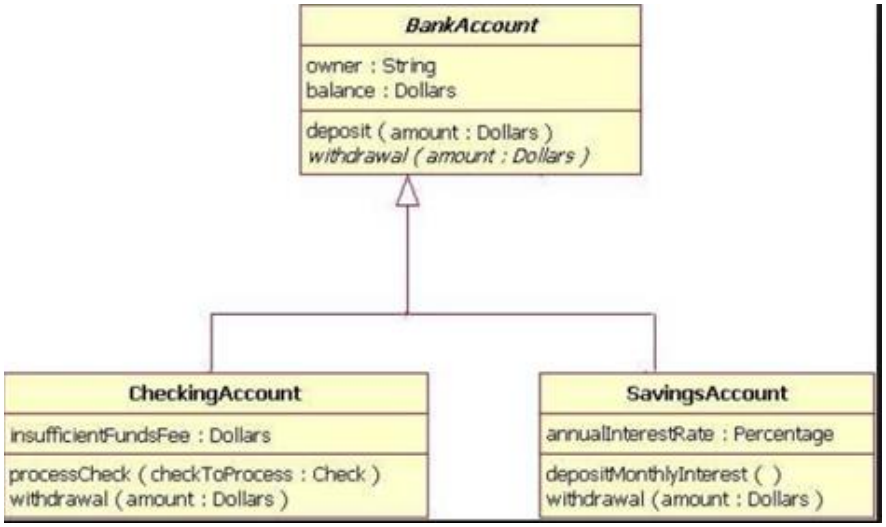
Output :



```
Volume of Cuboid : 750
Area of the Cube : 64
Press any key to continue . . .
```

Result : Sucessfully executed the program.

Aim : Implement the following inheritance :


BankAccount
owner : String
balance : Dollars
deposit (amount : Dollars)
withdrawal (amount : Dollars)
CheckingAccount
insufficientFundsFee : Dollars
processCheck (checkToProcess : Check)
withdrawal (amount : Dollars)
SavingsAccount
annualInterestRate : Percentage
depositMonthlyInterest ( )
withdrawal (amount : Dollars)

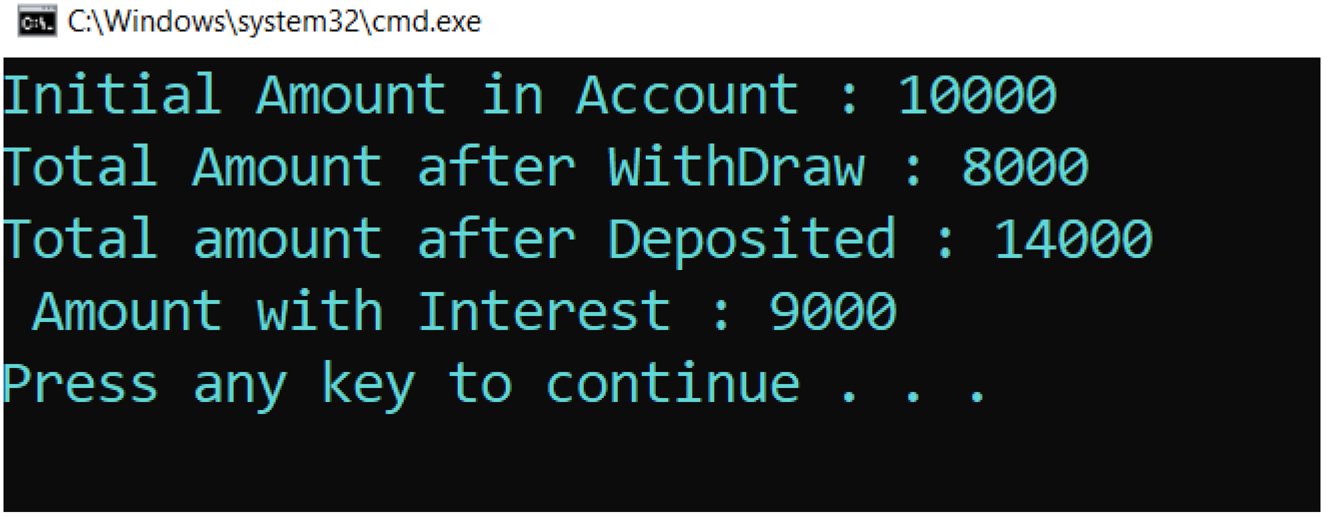
Program :

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

namespace Bank_Console
{
   internal class Program
   {
      static void Main(string[] args)
```

```
    {
        CheckAccount ca = new CheckAccount();
        Console.WriteLine("Initial Amount in Account : " + ca.amount);


        ca.ProcessCheck(2000);
        Console.WriteLine("Total Amount after WithDraw : " + ca.amount);


        ca.Deposit(6000);
        Console.WriteLine("Total amount after Deposited : " + ca.amount);


        SavingsAccount sa = new SavingsAccount();
        sa.DepositMonthlyInterest();
        Console.WriteLine(" Amount with Interest : " + sa.amount);


    }
}


class BankAccount
{
    public string owner;
    public float balance;
```

```
        public float amount = 10000;
        public void Deposit(float new_amount)
        {
            this.amount = this.amount + new_amount;
        }

        public void WithDrawl(float old_amount)
        {
            this.amount = this.amount - old_amount;
        }
    }
    class CheckAccount : BankAccount
    {
        public float InsufficentFee;

        public void ProcessCheck(float check_amount)
        {
            if (check_amount > base.amount)
            {
                Console.WriteLine(" You Don't Have Sufficent Amount in Account");
            }
            else
            {
                base.WithDrawl(check_amount);
            }
        }
    }
    class SavingsAccount : BankAccount
    {
        public float AnnualInterest = 0.9f;

        public void DepositMonthlyInterest()
        {
            base.amount = base.amount * AnnualInterest;
        }
    }
}
```

Output :



```
Initial Amount in Account : 10000
Total Amount after WithDraw : 8000
Total amount after Deposited : 14000
 Amount with Interest : 9000
Press any key to continue . . .
```

Result : Sucessfully Executed the program.