NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(Autonomous Institution, Affiliated to VTU, Belgaum, Approved by AICTE & State Govt. of Karnataka)

Yelahanka, Bangaluru – 560064

DEPARTMENT OF MCA

.NET Laboratory

LAB MANUAL

2018-19

.NET Laboratory

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – II

Subject Code	16MCA27	CIE Marks	50
Number of Lecture Hours/Week	01 Hr Tutorial(Instructions) 02 Hrs Laboratory	SEE Marks	50
		SEE Hours	03

CREDITS – 02

Course Outcomes(CO):

This laboratory course enable students to get practical experience in design, develop, implement, analyze and evaluation/testing of

After studying this course, students will be able to:

CO1: Understand C# and client-server concepts using .Net Frame Work Components

CO2: Apply delegates, event and exception handling to incorporate with ASP, Win Form, ADO.NET

CO3: Analyze the use of .Net Components depending on the problem statement

CO4: Implement & develop a web based and Console based application with Database connectivity

Laboratory Experiments:

PART A

1.Write a Program in C# to demonstrate Command line arguments processing for the following.

a) To find the square root of a given number.

b) To find the sum & average of three numbers.

2.Write a Program in C# to demonstrate the following

a) Boxing and Unboxing b) Invalid Unboxing.

3.Write a program in C# to add Two complex numbers using Operator overloading .

4. Write a Program in C# to find the sum of each row of given jagged array of 3 inner arrays.

5. Write a Program in C# to demonstrate Array Out of Bound Exception using Try, Catch and Finally blocks.

6.Write a Program to Demonstrate Use of Virtual and override key words in C# with a simple program.

7. Write a Program in C# to create and implement a Delegate for any two arithmetic operations

8. Write a Program in C# to demonstrate abstract class and abstract methods in C#.

9. Write a program to Set & Get the Name & Age of a person using Properties of C# to illustrate the use of different properties in C#.

10. Write a Program in C# Demonstrate arrays of interface types (for runtime polymorphism).

PART B(Self Learning Activity)

Implement and develop a Suitable ASP.Net Web based application with Database connectivity:

- ***The Web Page should be implemented in ASP.Net by using Standard web and validation controls.***
- ***Display the database in a Grid view***
- ***The Student should be able to do the database connection with respect to the ASP.Net web page.***
- ***The Student should be able to do modification with the database like Insert, Delete and Updating of the database.***

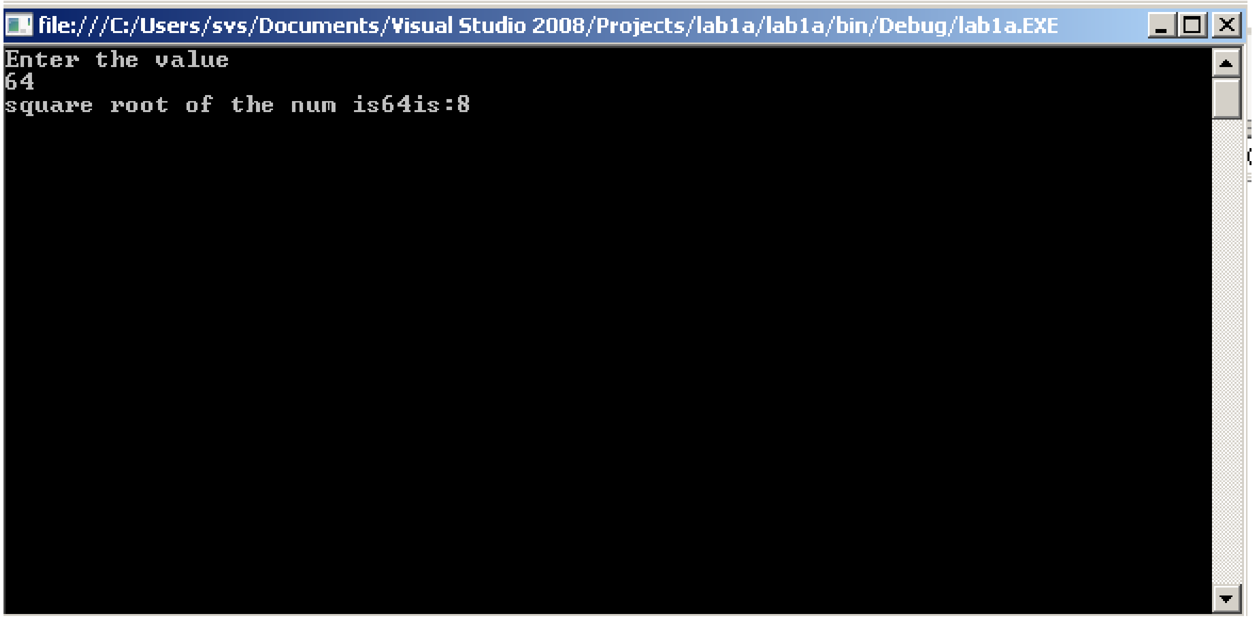
Note: In the examination each student should do one question from Part A followed by PART B demo.

1. Write a Program in C# to demonstrate Command line arguments processing for the following.
a) To find the square root of a given number.
b) To find the sum & average of three numbers.

Solution:
a) To find the square root of a given number.

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

namespace lab1a
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter the value");
            int i;
            i = int.Parse(Console.ReadLine());
            double sqr = Math.Sqrt(i);
            Console.WriteLine("square root of the num is{0}is:{1}", i, sqr);
            Console.ReadLine();
        }
    }
}
```

Output:


file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/lab1a/lab1a/bin/Debug/lab1a.EXE

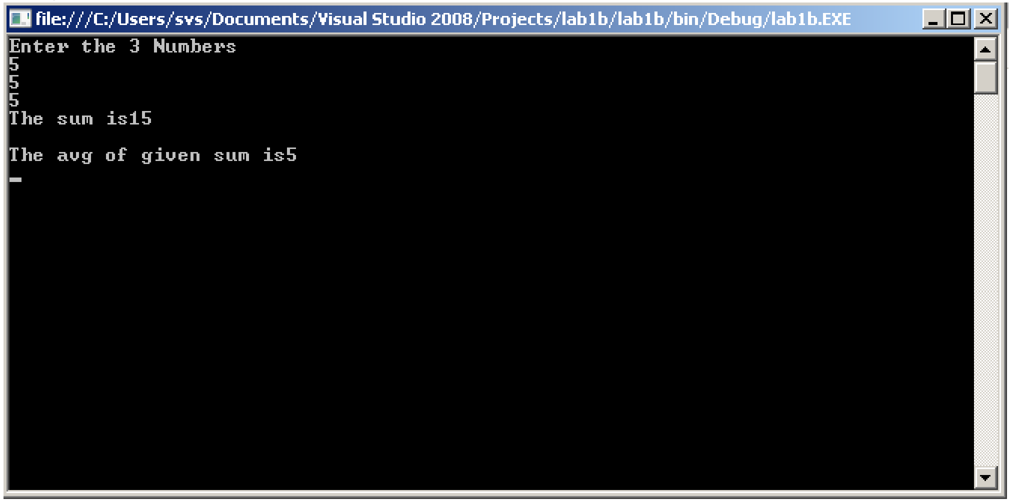
```
Enter the value
64
square root of the num is64is:8
```

Solution:

b) To find the sum & average of three numbers.

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


namespace lab1b
{
    class Program
    {
        static void Main(string[] args)
        {
            int a, b, c, sum;
            float avg;
            Console.WriteLine("Enter the 3 nums");
            a = int.Parse(Console.ReadLine());
            b = int.Parse(Console.ReadLine());
            c = int.Parse(Console.ReadLine());
            sum = a + b + c;
            Console.WriteLine("The sum is" + sum);
            Console.ReadLine();
            avg = sum / 3;
            Console.WriteLine("The avg of given sum is" + avg);
            Console.ReadLine();
        }
    }
}
```

Output:


file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/lab1b/lab1b/bin/Debug/lab1b.EXE

```
Enter the 3 Numbers
5
5
5
The sum is15

The avg of given sum is5
_
```

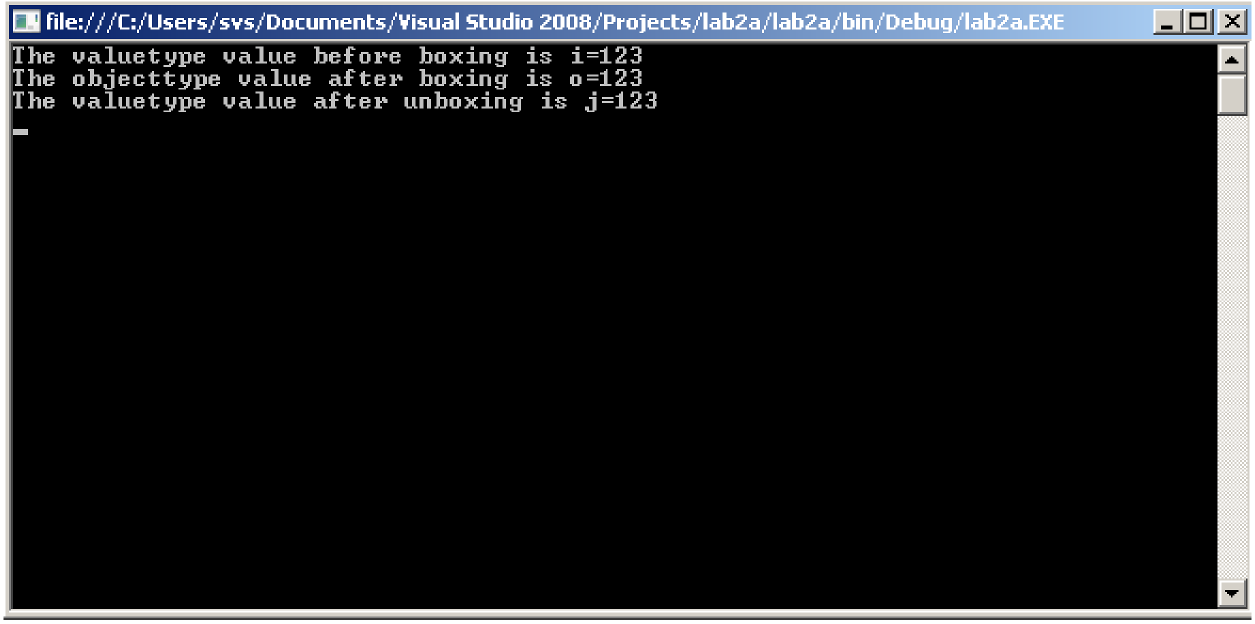
2. Write a Program in C# to demonstrate the following :
a) Boxing and Unboxing
b) Invalid Unboxing.

Solution:

a) Boxing and Unboxing

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

namespace lab2a
{
    class Program
    {
        static void Main(string[] args)
        {
            int i = 123;
            object o = i;
            int j = (int)o;
            Console.WriteLine("The valuetype value before boxing is i=" + i);
            Console.WriteLine("The objecttype value after boxing is o=" + o);
            Console.WriteLine("The valuetype value after unboxing is j=" +
j);
            Console.ReadLine();
        }
    }
}
```

Output:



file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/lab2a/lab2a/bin/Debug/lab2a.EXE

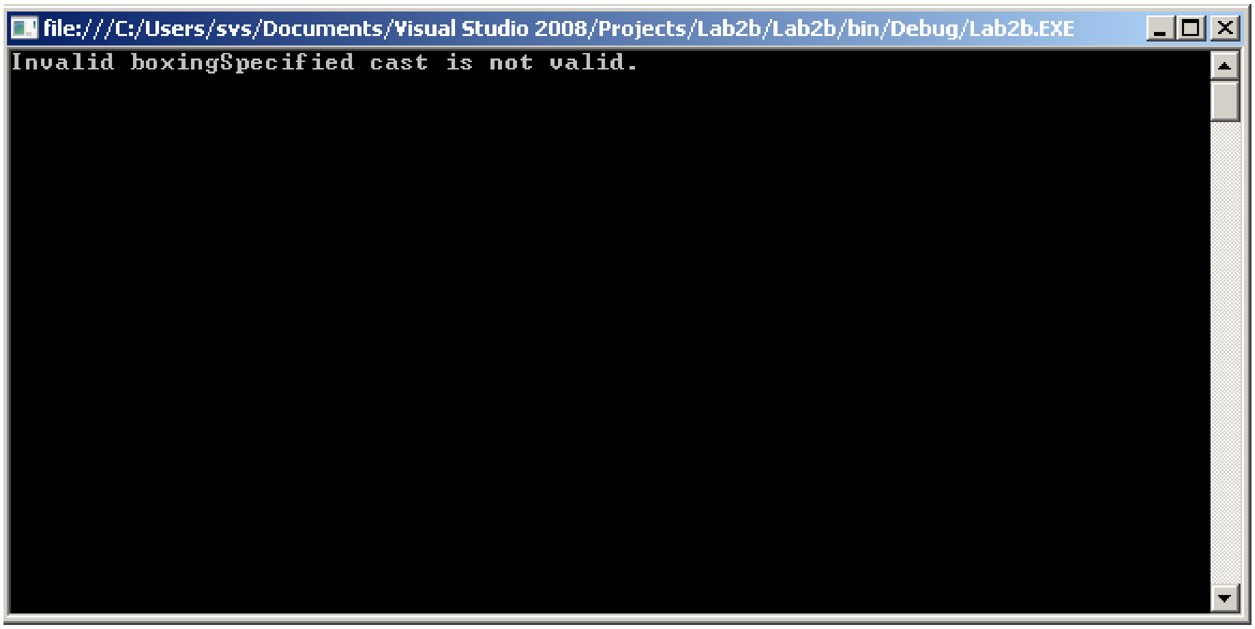
```
The valuetype value before boxing is i=123
The objecttype value after boxing is o=123
The valuetype value after unboxing is j=123
```

Solution:

b) Invalid Unboxing.

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

namespace lab2b
{
    class Program
    {
        static void Main(string[] args)
        {
            int i = 123;
            object o = i;
            try
            {
                int j = (Short)o;
                Console.WriteLine("unboxing is successful");

            }
            catch (System.InvalidCastException e)
            {
                Console.WriteLine("Invalid boxing" + e.Message);

            }
            Console.ReadLine();
        }
    }
}
```

Output:


file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/Lab2b/Lab2b/bin/Debug/Lab2b.EXE

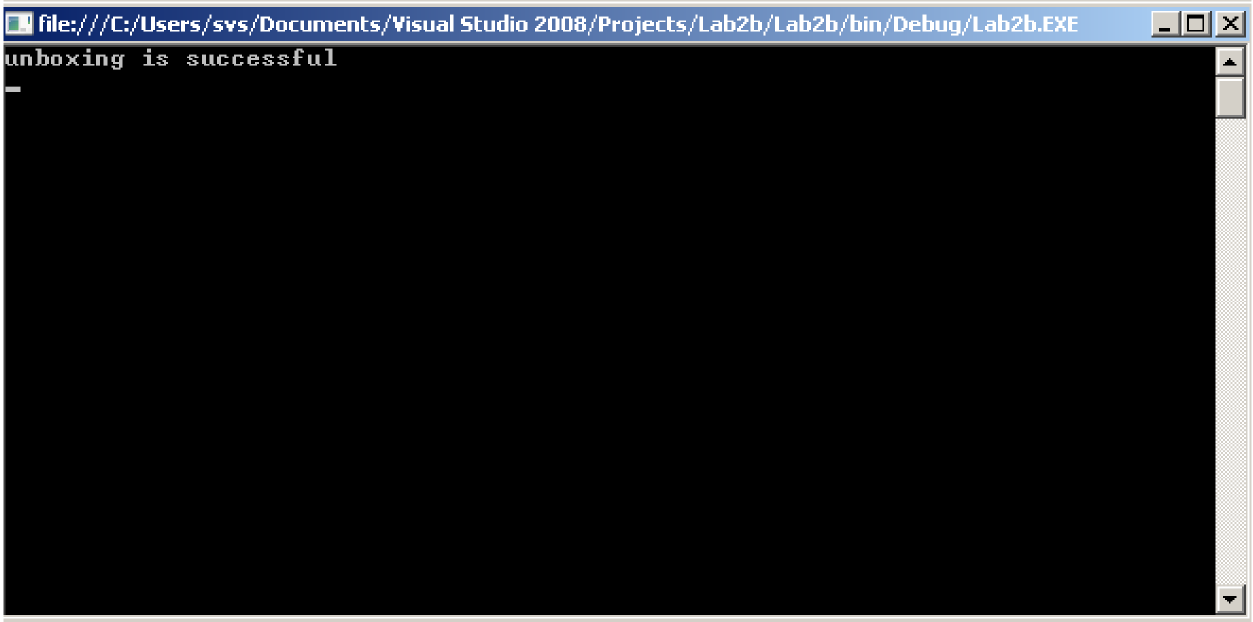
```
Invalid boxingSpecified cast is not valid.
```

Here, if we change the data type of o to int i.e

```
                    int j = (int)o;
```

Then we can achieve unboxing.

Output:



file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/Lab2b/Lab2b/bin/Debug/Lab2b.EXE

```
unboxing is successful
_
```

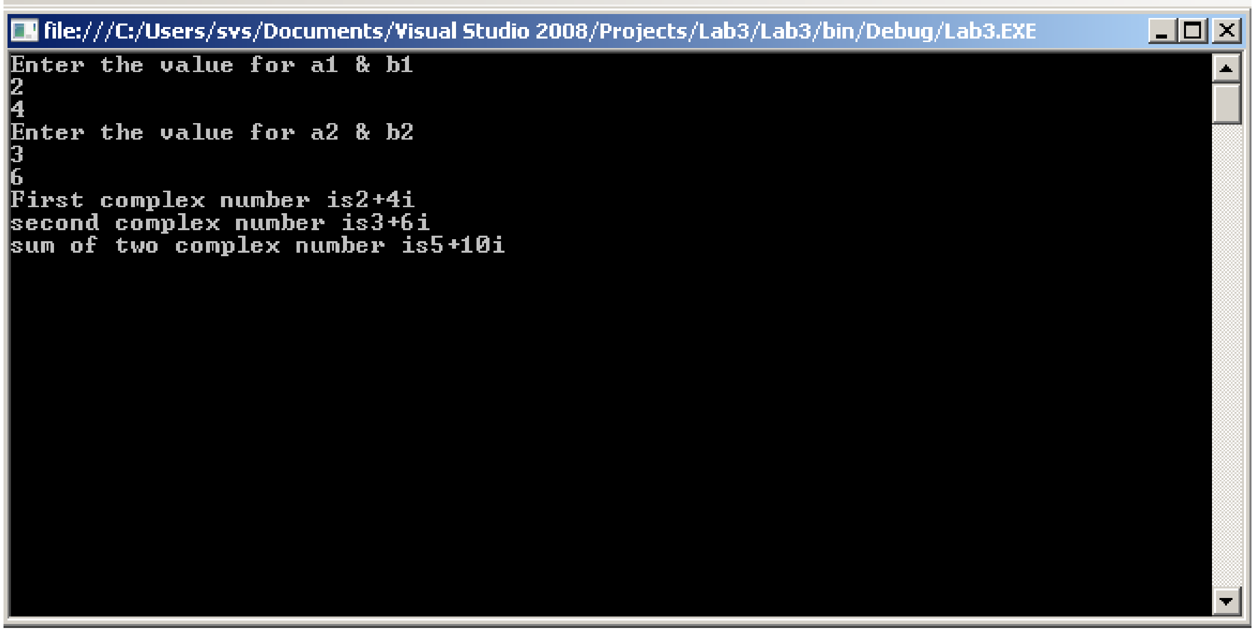
3.Write a program in C# to add Two complex numbers using Operator overloading .

Solution:

To add two complex numbers using Operator overloading:

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


namespace lab3
{
    class complex
    {
        int real;
        int imaginary;
        public complex(int r, int img)
        {
            real = r;
            imaginary = img;
        }
        public static complex operator +(complex c1, complex c2)
        {
            return new complex(c1.real + c2.real, c1.imaginary + 
c2.imaginary);
        }
        public override string ToString()
        {
            return (String.Format("{0}+{1}i", real, imaginary));
        }
    }
    class demo
    {
        static void Main(string[] args)
        {
            int a1, b1, a2, b2;
            Console.WriteLine("Enter the value for a1 & b1");
            a1 = Int32.Parse(Console.ReadLine());
            b1 = Int32.Parse(Console.ReadLine());
            complex num1 = new complex(a1, b1);
            Console.WriteLine("Enter the value for a2 & b2");
            a2 = Int32.Parse(Console.ReadLine());
            b2 = Int32.Parse(Console.ReadLine());
            complex num2 = new complex(a2, b2);
            complex sum = num1 + num2;
            Console.WriteLine("First complex number is"  + num1);
            Console.WriteLine("second complex number is"  + num2);
            Console.WriteLine("sum of two complex number is"  + sum);
            Console.ReadLine();
        }
    }
}
```

Output:



file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/Lab3/Lab3/bin/Debug/Lab3.EXE

```
Enter the value for a1 & b1
2
4
Enter the value for a2 & b2
3
6
First complex number is2+4i
second complex number is3+6i
sum of two complex number is5+10i
```

4. Write a Program in C# to find the sum of each row of given jagged array of 3 inner arrays.

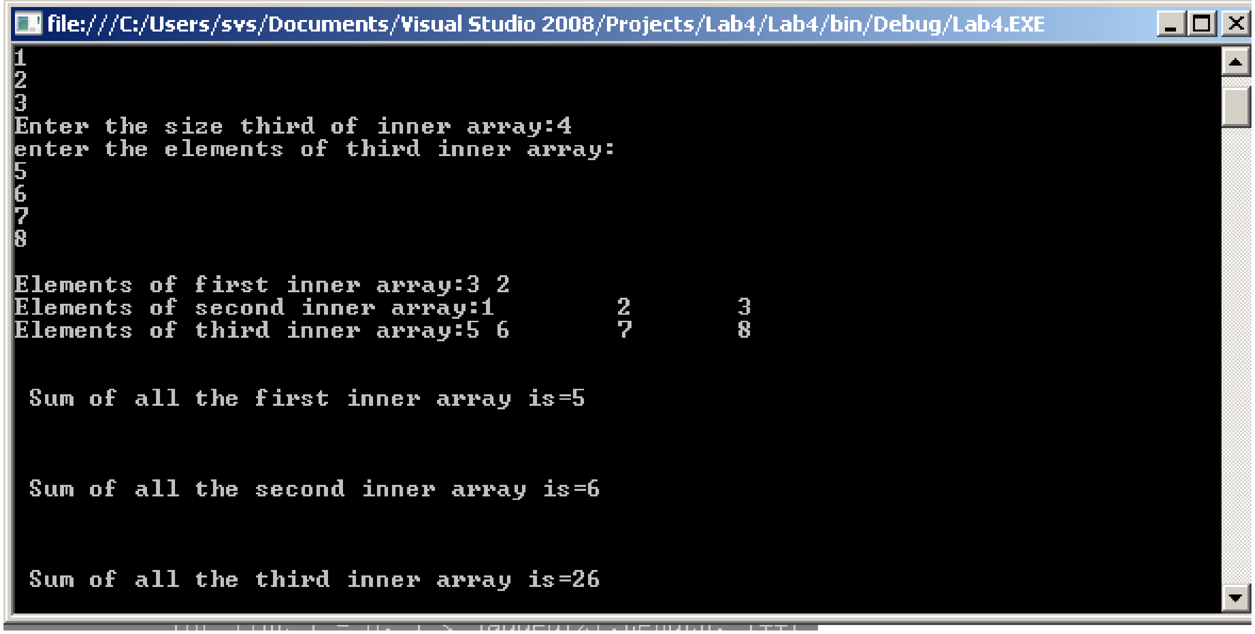
Solution:

To find the sum of each row of given jagged array of 3 inner arrays:

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


namespace lab4
{
    class jaggedarrays
    {
        int[][] jagged = new int[3][];
        public void ReadArrays()
        {
            Console.Write("Enter the size of first inner array:");
            jagged[0] = new int[int.Parse(Console.ReadLine())];
            Console.WriteLine("enter the elements of first inner array:");
            for (int i = 0; i < jagged[0].Length; i++)
                jagged[0][i] = int.Parse(Console.ReadLine());
            Console.Write("Enter the size second of inner array:");
            jagged[1] = new int[int.Parse(Console.ReadLine())];
            Console.WriteLine("enter the elements of second inner array:");
            for (int i = 0; i < jagged[1].Length; i++)
                jagged[1][i] = int.Parse(Console.ReadLine());
            Console.Write("Enter the size third of inner array:");
            jagged[2] = new int[int.Parse(Console.ReadLine())];
            Console.WriteLine("enter the elements of third inner array:");
            for (int i = 0; i < jagged[2].Length; i++)
                jagged[2][i] = int.Parse(Console.ReadLine());

        }
        public void FindSum()
        {
            int sum = 0;
            for (int i = 0; i < jagged[0].Length; i++)
                sum = sum + jagged[0][i];
            Console.WriteLine("\n\n\n Sum of all the first inner array
is={0}", sum);


            sum = 0;
            for (int i = 0; i < jagged[1].Length; i++)
                sum = sum + jagged[1][i];
            Console.WriteLine("\n\n\n Sum of all the second inner array
is={0}", sum);


            sum = 0;
            for (int i = 0; i < jagged[2].Length; i++)
```

```
                sum = sum + jagged[2][i];
            Console.WriteLine("\n\n\n Sum of all the third inner array
is={0}", sum);
        }
        public void PrintArrays()
        {
            Console.Write("\nElements of first inner array:");
            for (int i = 0; i < jagged[0].Length; i++)
                Console.Write(jagged[0][i] + "\t");
            Console.Write("\nElements of second inner array:");
            for (int i = 0; i < jagged[1].Length; i++)
                Console.Write(jagged[1][i] + "\t");
            Console.Write("\nElements of third inner array:");
            for (int i = 0; i < jagged[2].Length; i++)
                Console.Write(jagged[2][i] + "\t");

        }
        class demo
        {

            static void Main(string[] args)
            {
                jaggedarrays ja = new jaggedarrays();
                ja.ReadArrays();
                ja.PrintArrays();
                ja.FindSum();
                Console.ReadLine();
            }
        }
    }
}
```

Output:


file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/Lab4/Lab4/bin/Debug/Lab4.EXE

```
1
2
3
Enter the size third of inner array:4
enter the elements of third inner array:
5
6
7
8

Elements of first inner array:3 2
Elements of second inner array:1        2       3
Elements of third inner array:5 6       7       8


 Sum of all the first inner array is=5



 Sum of all the second inner array is=6



 Sum of all the third inner array is=26
```

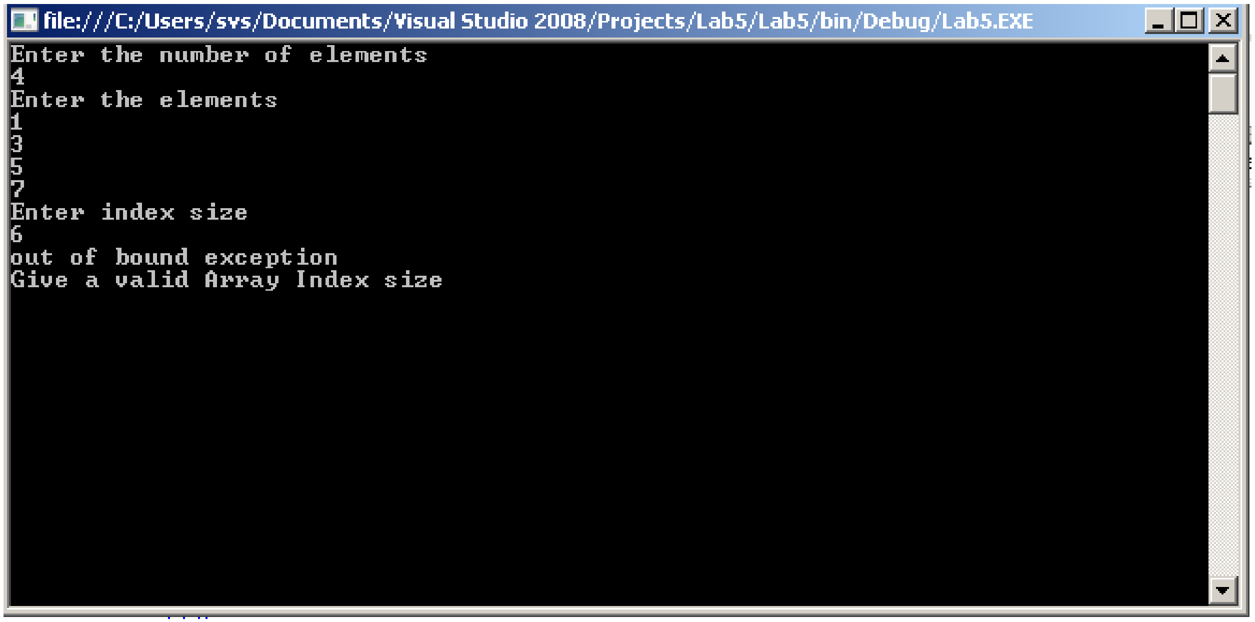
5. Write a Program in C# to demonstrate Array Out of Bound Exception using Try, Catch and Finally blocks.

Solution:

To demonstrate Array Out of Bound Exception using Try, Catch and Finally blocks:

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


namespace lab5
{
    class Program
    {
        public static void Main(string[] args)
        {
            int i, ele;
            Console.WriteLine("Enter the number of elements");
            ele = int.Parse(Console.ReadLine());
            int[] a = new int[ele];
            Console.WriteLine("Enter the elements");
            for (i = 0; i != ele; i++)
            {
                a[i] = int.Parse(Console.ReadLine());
            }
            Console.WriteLine("Enter index size");
            int element = int.Parse(Console.ReadLine());
            try
            {
                Console.WriteLine("element is" + a[element]);

            }
            catch (IndexOutOfRangeException e)
            {
                Console.WriteLine("out of bound exception");
            }
            finally
            {
                Console.WriteLine("Give a valid Array Index size");
            }
            Console.ReadLine();

        }
    }
}
```

Output:



file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/Lab5/Lab5/bin/Debug/Lab5.EXE

```
Enter the number of elements
4
Enter the elements
1
3
5
7
Enter index size
6
out of bound exception
Give a valid Array Index size
```

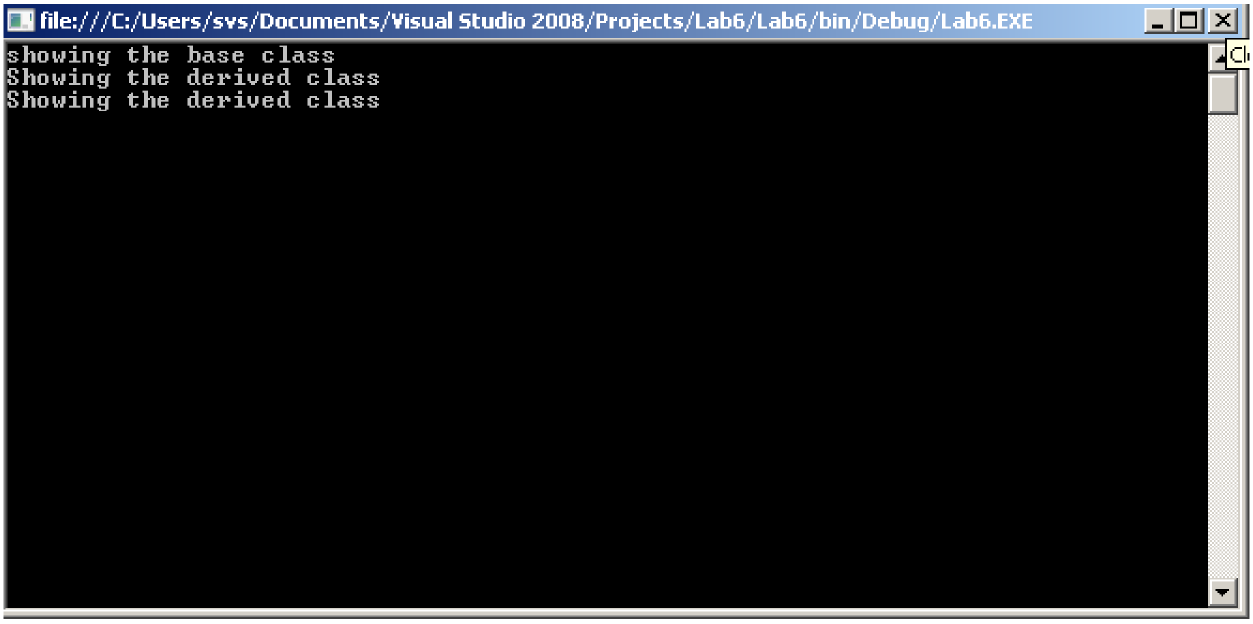
6.Write a Program to Demonstrate Use of Virtual and override key words in C# with a simple program.

Solution:

To Demonstrate Use of Virtual and override key words in C# with a simple program:

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


namespace lab6
{
    class Base
    {
        public virtual void show()
        {
            Console.WriteLine("showing the base class");

        }
    }
    class Derived : Base
    {
        public override void show()
        {
            Console.WriteLine("Showing the derived class");
        }
    }
    class demo
    {
        static void Main(string[] args)
        {
            Base b = new Base();
            b.show();
            Derived d = new Derived();
            d.show();
            Base b1 = new Derived();
            b1.show();
            Console.ReadLine();
        }
    }
}
```

Output:



file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/Lab6/Lab6/bin/Debug/Lab6.EXE

```
showing the base class
Showing the derived class
Showing the derived class
```

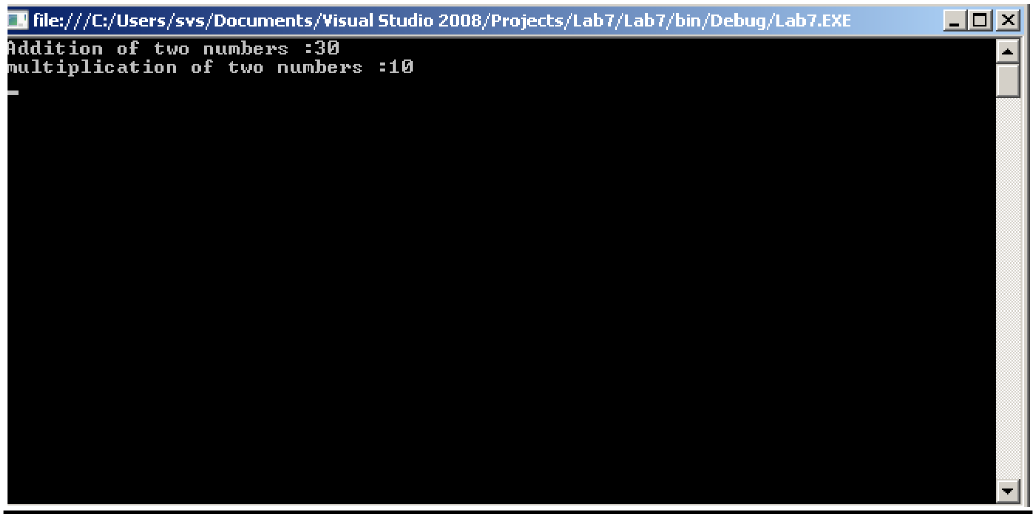
7. Write a Program in C# to create and implement a Delegate for any two arithmetic operations

Solution:

To create and implement a Delegate for any two arithmetic operations:

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

public delegate int operation(int p, int q);
namespace lab7
{
    class delegates
    {
        public static int AddNum(int p, int q)
        {
            return (p + q);
        }
        public int MultiNum(int p, int q)
        {
            return (p * q);
        }
        static void Main(string[] args)
        {
            operation op1 = new operation(AddNum);
            Console.WriteLine("Addition of two numbers :{0}", op1(10, 20));
            delegates d = new delegates();
            operation op2 = new operation(d.MultiNum);
           Console.WriteLine("multiplication of two numbers :{0}", op2(5,2));
            Console.ReadLine();
        }
    }
}
```

Output:


file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/Lab7/Lab7/bin/Debug/Lab7.EXE

```
Addition of two numbers :30
multiplication of two numbers :10
_
```

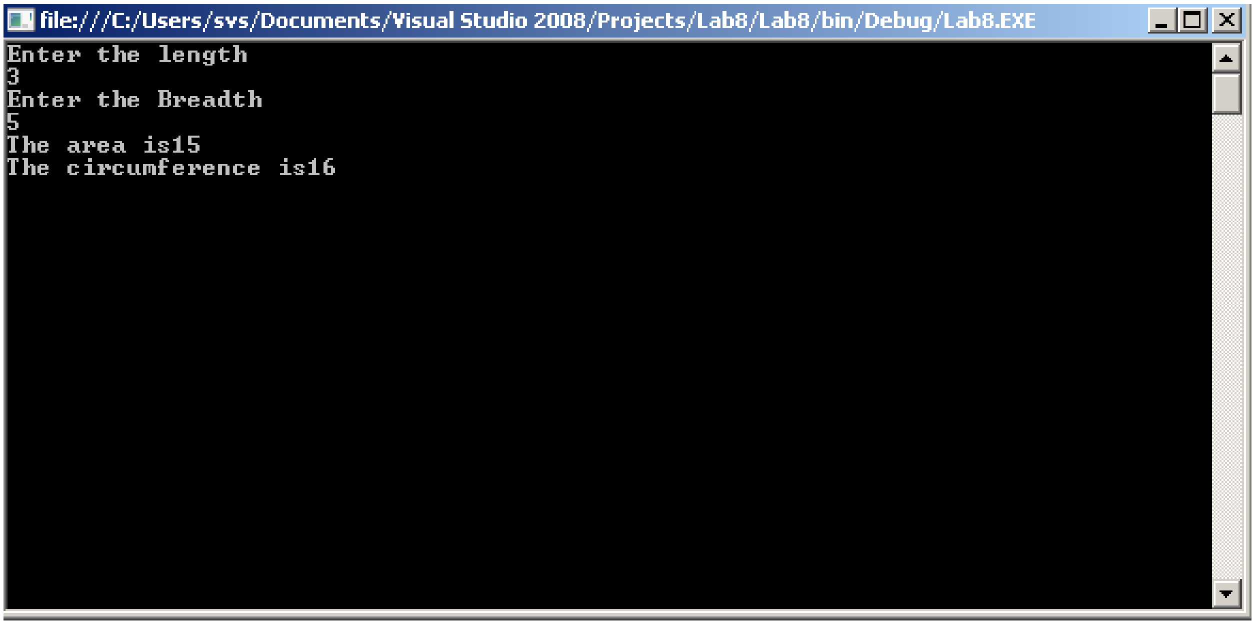
8. Write a Program in C# to demonstrate abstract class and abstract methods in C#.

Solution

To demonstrate abstract class and abstract methods in C#:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace lab8
{
    abstract class shape
    {
        protected float l, b;
        public abstract float Area();
        public abstract float Circumference();
    }
    class Rectangle : shape
    {
        public void getlb()
        {
            Console.WriteLine("Enter the length");
            l = float.Parse(Console.ReadLine());
            Console.WriteLine("Enter the Breadth");
            b = float.Parse(Console.ReadLine());
        }
        public override float Area()
        {
            return l * b;
        }
        public override float Circumference()
        {
            return 2 * (l + b);
        }
    }
    class demo
    {
        static void Main(string[] args)
        {
            Rectangle r = new Rectangle();
            r.getlb();
            Console.WriteLine("The area is" + r.Area());
            Console.WriteLine("The circumference is" + r.Circumference());
            Console.ReadLine();
        }
    }
}
```

Output:



file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/Lab8/Lab8/bin/Debug/Lab8.EXE

```
Enter the length
3
Enter the Breadth
5
The area is15
The circumference is16
```

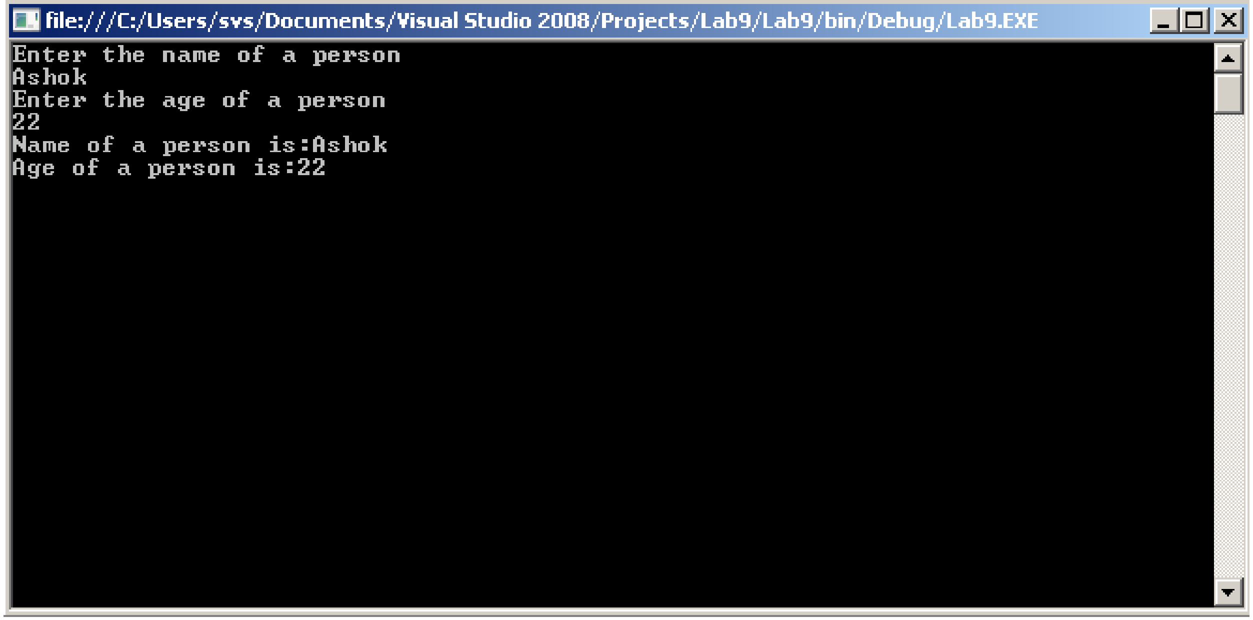
9. Write a program to Set & Get the Name & Age of a person using Properties of C# to illustrate the use of different properties in C#.

Solution:

To Set & Get the Name & Age of a person using Properties of C# to illustrate the use of different properties in C#:

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


namespace lab9
{
    class Person
    {
        string name;
        int age;
        public string pername
        {
            get { return name; }
            set { name = value; }
        }
        public int perage
        {
            get { return age; }
            set { age = value; }
        }
    }
    class demo
    {
        static void Main(string[] args)
        {
            Person p1 = new Person();
            Console.WriteLine("Enter the name of a person");
            p1.pername = Console.ReadLine();
            Console.WriteLine("Enter the age of a person");
            p1.perage = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Name of a person is:" + p1.pername);
            Console.WriteLine("Age of a person is:" + p1.perage);
            Console.ReadLine();

        }
    }
}
```

Output:



file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/Lab9/Lab9/bin/Debug/Lab9.EXE

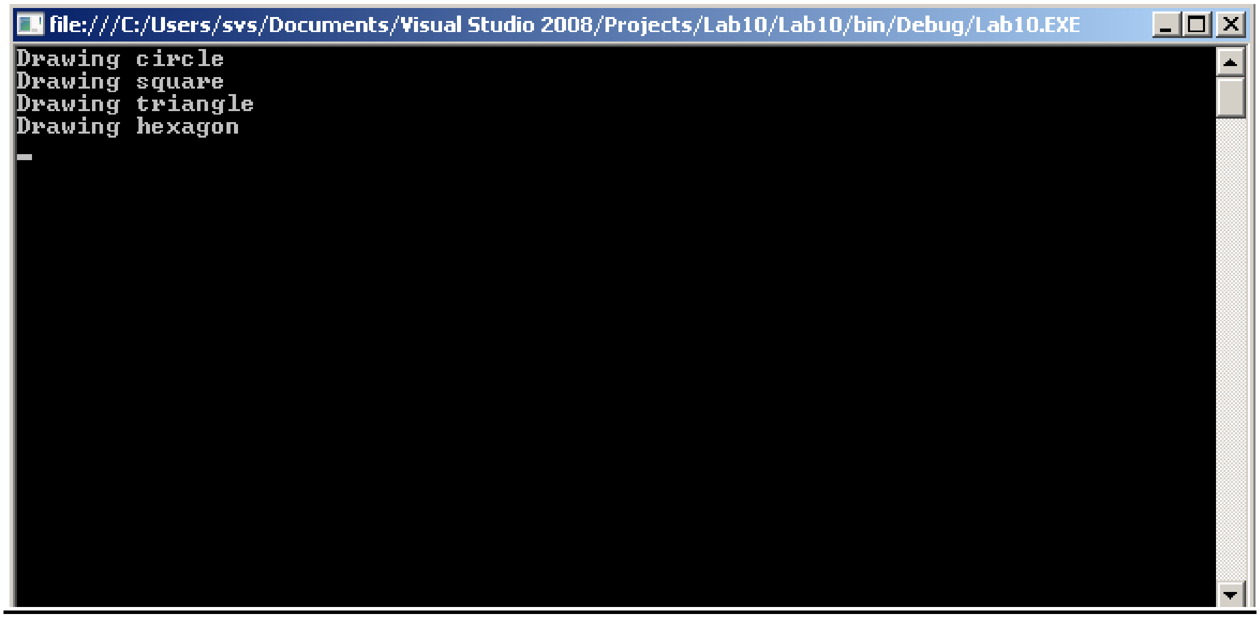
```
Enter the name of a person
Ashok
Enter the age of a person
22
Name of a person is:Ashok
Age of a person is:22
```

10. Write a Program in C# Demonstrate arrays of interface types (for runtime polymorphism).

Solution:
To demonstrate arrays of interface types (for runtime polymorphism):

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


namespace lab10
{
    interface ishape
    {
        void draw();
    }
    class circle : ishape
    {
        public void draw()
        {
            Console.WriteLine("Drawing circle");
        }
    }
    class square : ishape
    {
        public void draw()
        {
            Console.WriteLine("Drawing square");
        }
    }
    class triangle : ishape
    {
        public void draw()
        {
            Console.WriteLine("Drawing triangle");
        }
    }
    class hexagon : ishape
    {
        public void draw()
        {
            Console.WriteLine("Drawing hexagon");
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            ishape[] shape = { new circle(), new square(), new triangle(),
new hexagon() };
            foreach (ishape s in shape)
                s.draw();
            Console.ReadLine();

        }
    }
}
```

Output:


file:///C:/Users/svs/Documents/Visual Studio 2008/Projects/Lab10/Lab10/bin/Debug/Lab10.EXE

```
Drawing circle
Drawing square
Drawing triangle
Drawing hexagon
_
```