

Practical 2A): Write the program for the following

Aim : Create a simple application to demonstrate the concepts of boxing and unboxing.

Code :

```
using System;
namespace awd_32
{
    class Program
    {
        static void Main()
        {
            int num = 23;
            object obj = num; //boxing
            int i = (int)obj; //unboxing
            Console.WriteLine("Value of object : "+obj);
            Console.WriteLine("Value of i : "+i);
        }
    }
}
```

Output :

```
Value of object : 23
Value of i : 23
```

Practical 2B)

Aim: Create a simple application to perform addition, subtraction, multiplication and division using delegate.

Code :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace awd_32
{
    delegate void MyDelegate(int x, int y);
    public class CalculatedDelegate
    {
        public static void Add(int x, int y){
            Console.WriteLine("Addition = "+(x+y)); }
        public static void Sub(int x, int y){
            Console.WriteLine("Subtraction = "+(x-y)); }
        public static void Mul(int x, int y){
            Console.WriteLine("Multiplication = "+(x*y));}
        public static void Div(int x, int y){
            Console.WriteLine("Division = "+(x/y));}
    }
    class Program{
        public static void Main(string[] args){
            MyDelegate obj = new MyDelegate(CalculatedDelegate.Add);
            obj(6,4);
            MyDelegate obj1 = new MyDelegate(CalculatedDelegate.Sub);
            obj1(2,5);
            MyDelegate obj2 = new MyDelegate(CalculatedDelegate.Mul);
            obj2(6,4);
            MyDelegate obj3 = new MyDelegate(CalculatedDelegate.Div);
            obj3(6,4);
            Console.WriteLine("\n") }}}}
```

Output :

```
Addition = 10
Subtraction = -3
Multiplication = 24
Division = 1
```

Practical 2C):

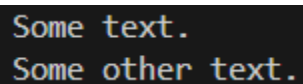
Aim: Create a simple application to demonstrate the use of the concepts of interfaces.

Code :

```
using System;
namespace awd_32
{
    interface IFirstInterface
    {
        void myMethod(); // interface method
    }
    interface ISecondInterface
    {
        void myOtherMethod(); // interface method
    }
    // implement multiple interfaces
    class DemoClass : IFirstInterface, ISecondInterface
    {
        public void myMethod()
        {
            Console.WriteLine("Some text.");
        }
        public void myOtherMethod()
        {
            Console.WriteLine("Some other text.");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            DemoClass myObj = new DemoClass();
            myObj.myMethod();
            myObj.myOtherMethod();
        }
    }
}
```

Output :



```
Some text.
Some other text.
```