

## Practical No. 2.1

**Aim:** Implementation of Basic example of Console Application in c#.


**Code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    class emp
    {
        int eno; string nm; double sal;
        internal void getdata()
        {
            Console.WriteLine("Enter employee number, name and
salary:");
            eno = int.Parse(Console.ReadLine());
            nm = Console.ReadLine();
            sal = double.Parse(Console.ReadLine());

        }

        internal void putdata()
        {
            Console.WriteLine("Employee number:" + eno);
            Console.WriteLine("Employee name: {0}", nm);
            Console.WriteLine("Employee salary:" + sal);
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            emp e = new emp();
            e.getdata();
            e.putdata();
            Console.ReadLine();
        }
    }
}
```

## Output:



```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE
Enter employee number, name and salary:
101
rakesh
50000
Employee number:101
Employee name: rakesh
Employee salary:50000
```

The screenshot shows a Windows console window titled "file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE". The window contains the following text: "Enter employee number, name and salary:", "101", "rakesh", "50000", "Employee number:101", "Employee name: rakesh", and "Employee salary:50000". The text is displayed in a monospaced font on a black background.

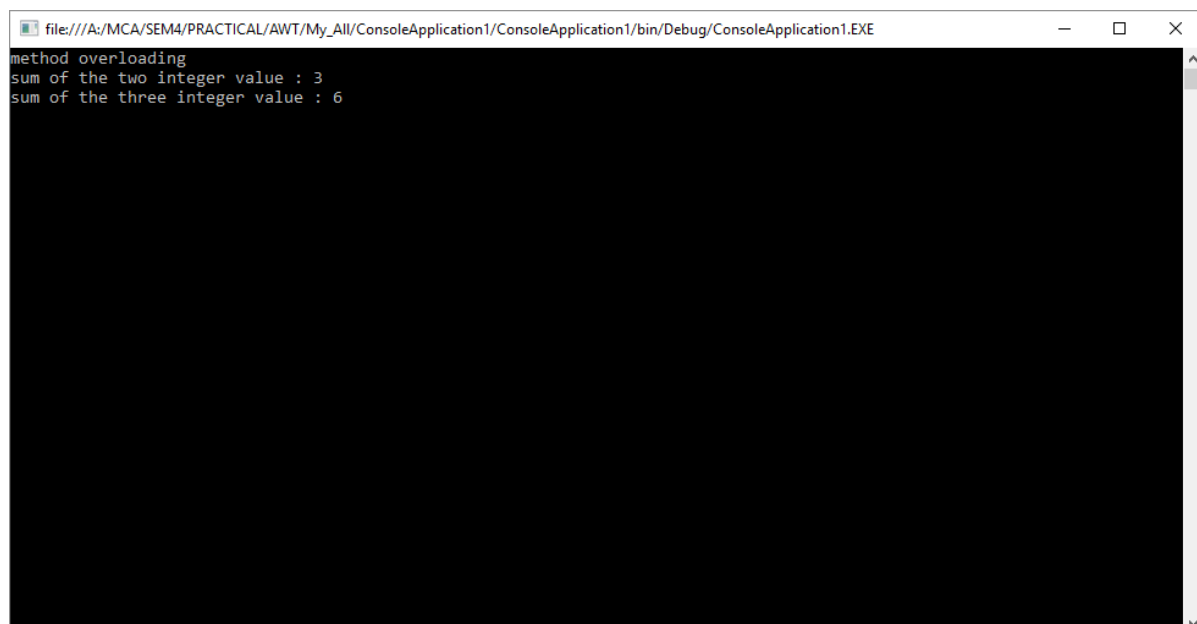
## Practical No. 2.2

**Aim:** Write a program of Method overloading.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    class demo
    {
        public int Add(int a, int b)
        {
            int sum = a + b; return sum;
        }
        public int Add(int a, int b, int c)
        {
            int sum = a + b + c; return sum;
        }
    }
    class Class1
    {
        static void Main(string[] args)
        {
            demo ob = new demo();
            Console.WriteLine("method overloading");
            int sum1 = ob.Add(1, 2);
            Console.WriteLine("sum of the two " + "integer value : " + sum1);
            int sum2 = ob.Add(1, 2, 3);
            Console.WriteLine("sum of the three " + "integer value : " + sum2);
            Console.ReadLine();
        }
    }
}
```

**Output:**

A screenshot of a Windows console application window. The title bar shows the file path: "file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE". The console output is as follows:

```
method overloading
sum of the two integer value : 3
sum of the three integer value : 6
```

### Practical No. 2.3

**Aim: Implementation of method overloading,**

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    class abc
    {
        int r, l, b;
        internal void area(int r)
        {
            double ar = 3.14 * r * r;
            Console.WriteLine("Area of circle:" + ar);
        }
        internal void area(int l, int b)
        {
            int ar = l * b;
            Console.WriteLine("Area of rectangle:" + ar);
        }
        internal void area(double b, double h)
        {
            double ar = (b * h) / 2;
            Console.WriteLine("Area of triangle:" + ar);
        }
    }
    class Class2
    {
        static void Main(string[] args)
        {
            abc ob = new abc(); ob.area(3);
            ob.area(5, 10);
            ob.area(2.5, 5.10);
            Console.ReadLine();
        }
    }
}
```

## Output:



A screenshot of a Windows console application window. The title bar shows the file path: file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE. The window contains three lines of text: "Area of circle:28.26", "Area of rectangle:50", and "Area of triangle:6.375". The console has a black background and white text. A vertical scrollbar is visible on the right side of the window.

```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE
Area of circle:28.26
Area of rectangle:50
Area of triangle:6.375
```

## Practical No. 2.4

**Aim:** Write a program for static variable.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    class test
    {
        int no;
        static int cnt;
        internal void get(int n)
        {
            no = n; cnt++;
        }
        internal void tnew()
        {
            Console.WriteLine("Number: {0}", no);
            Console.WriteLine("Count:" + cnt);
        }
    }
    class Class3
    {
        static void Main(string[] args)
        {
            test t1= new test();
            test t2= new test();
            test t3 =new test();
            t1.get(100);
            t2.get(200);
            t3.get(300);
            t1.tnew();
            t2.tnew();
            t3.tnew();
            Console.ReadLine();
        }
    }
}
```

## Output:



A screenshot of a Windows console window. The title bar at the top reads "file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE". The console area is black with white text. The output consists of six lines: "Number:100", "Count:3", "Number:200", "Count:3", "Number:300", and "Count:3". A vertical scrollbar is visible on the right side of the console window.

```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE
Number:100
Count:3
Number:200
Count:3
Number:300
Count:3
```

## Practical No. 2.5

**Aim:** Write a program for static method.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    class test1
    {
        int no;
        static int cnt;
        internal void get(int n)
        {
            no = n; cnt++;
        }
        internal static void showcount()
        {
            Console.WriteLine("count:" + cnt);
        }
        internal void tnew()
        {
            Console.WriteLine("Number: {0}", no);
            // Console.WriteLine("Count:" + cnt);
        }
    }
    class Class4
    {
        static void Main(string[] args)
        {
            test1 t1 = new test1();
            test1 t2 = new test1();
            test1 t3 = new test1();
            t1.get(100);
            t2.get(200);
            t3.get(300);
            t1.tnew();
            t2.tnew();
            t3.tnew();
            test1.showcount();
            Console.ReadLine();
        }
    }
}
```



## Output:



A screenshot of a Windows console window. The title bar shows the file path: `file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE`. The console output is as follows:

```
Number:100  
Number:200  
Number:300  
count:3
```

## Practical No. 2.6

**Aim:** Write a program of constructor.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    class emp1
    {
        string name; int eno; double sal; public emp1()
        {
            name = "Kumar"; eno = 121;
            sal = 90000;
        }
        internal void show()
        {
            Console.WriteLine("Name:" + name);
            Console.WriteLine("Eno:" + eno);
            Console.WriteLine("Salary:" + sal);
        }
    }
    class Class5
    {
        static void Main(string[] args)
        {
            emp1 e = new emp1();
            e.show();
            Console.ReadLine();
        }
    }
}
```

## Output:



```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE
Name: Kumar
Eno:121
Salary:90000
```

## Practical No. 2.7

**Aim:** Write a program of constructor overloading.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    class complex
    {
        double a, b; public complex()
        {
        }
        public complex(double x,double y)
        {
            a = x; b = y;
        }
        internal void show()
        {
            Console.WriteLine(a + "+i" + b);
        }
        public complex add(complex c2)
        {
            complex temp= new complex();
            temp.a = a + c2.a;
            temp.b = b + c2.b;
            return temp;
        }
    }
    class Class6
    {
        static void Main(string[] args)
        {
            complex c1 = new complex(2.5,3.6);
            complex c2 = new complex(4.3,8.9);
            complex c3 = new complex();
            c1.show();
            c2.show();
            c3=c1.add(c2);
            c3.show();
            Console.ReadLine();
        }
    }
}
```

## Output:



```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE
2.5+i3.6
4.3+i8.9
6.8+i12.5
```

## Practical No. 2.8

**Aim:** Write a program of properties.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    class stud
    {
        int roll; string nm1;
        public int rollno
        {
            get
            {
                return roll;
            }
            set
            {
                roll = value;
            }
        }
        public string name
        {
            get
            {
                return nm1;
            }
            Set
            {
                nm1 = value;
            }
        }
    }
    class Class7
    {
        static void Main(string[] args)
        {
            stud s = new stud();
            s.rollno = 1;
            s.name = "diyansh";
            Console.WriteLine("Roll number:" + s.rollno);
            Console.WriteLine("Name:" + s.name);
            Console.ReadLine();
        }
    }
}
```

## Output:



```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE
Roll number:1
Name:diyanish
```

## Practical No. 2.9

**Aim: Implementation of Single Inheritance.**

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    class A
    {
        internal int a;
        internal void geta()
        {
            a = 10;
        }
    }
    class B : A
    {
        int b;
        internal void getb()
        {
            b = 20;
        }
        internal void show()
        {
            Console.WriteLine("a=" + a);
            Console.WriteLine("b=" + b);
        }
    }
    class Class8
    {
        static void Main(string[] args)
        {
            B ob = new B();
            ob.geta();
            ob.getb();
            ob.show();
            Console.ReadLine();
        }
    }
}
```



## Output:



The screenshot shows a Windows console window with the title bar text: `file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE`. The window contains two lines of text: `a=10` and `b=20`. The console area is black, and the text is white. A vertical scrollbar is visible on the right side of the console area.

```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE
a=10
b=20
```

## Practical No. 2.10

### Aim: Implementation of Multi-Level Inheritance.

#### Source code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    public class student
    {
        int roll; string name;
        internal void getdata()
        {
            Console.WriteLine("Enter your details");
            Console.Write("Enter roll no.::");
            roll = int.Parse(Console.ReadLine());
            Console.Write("Enter your name::");
            name = Console.ReadLine();
        }
        internal void putdata()
        {
            Console.WriteLine("roll no:" + roll);
            Console.WriteLine("name is:" + name);
        }
    }
    public class test2 : student
    {
        protected int marks1, marks2;
        internal void getmarks()
        {
            Console.WriteLine("Enter your marks");
            marks1 = int.Parse(Console.ReadLine());
            marks2 = int.Parse(Console.ReadLine());
        }
        internal void putmarks()
        {
            Console.WriteLine("marks1:{0} marks2:{1}", marks1, marks2);
        }
    }
    public class Result2 : test2
    {
        int total;
        internal void totalm()
        {
            total = marks1 + marks2;
            Console.WriteLine("total is=" + total);
        }
    }
    class Class9
    {
        static void Main(string[] args)
        {
            Result2 r = new Result2();
        }
    }
}
```

```
r.getdata();
r.putdata();
r.getmarks();
r.putmarks();
r.totalm(); Console.ReadLine();
}
}
}
```

## Output:



```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE
Enter your details
Enter roll no.:121
Enter your name::shruti
roll no:121
name is:shruti
Enter your marks
80
90
marks1:80 marks2:90
total is=170
```

## Practical No. 2.11

**Aim: Implementation of Inheritance with constructor.**

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication2
{
    class alpha
    {
        int a;
        internal alpha(int x)
        {
            a = x;
        }
        internal void showa()
        {
            Console.WriteLine("a=" + a);
        }
    }
    class beta : alpha
    {
        double b;
        internal beta(int x, double y) : base(x)
        {
            b = y;
        }
        internal void showb()
        {
            Console.WriteLine("b=" + b);
        }
    }
    class gamma : beta
    {
        int c, d;
        internal gamma(int x, double y, int m, int n):base(x,y)
        {
            c = m; d = n;
        }
        internal void showed()
        {
            Console.WriteLine("c=" + c);
            Console.WriteLine("d=" + d);
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            gamma g = new gamma(10, 20.5, 30, 40);
            g.showa();
            g.showb();
            g.showed();
            Console.ReadLine();
        }
    }
}
```

}

### Output:

A screenshot of a Windows console application window. The title bar shows the file path: file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication2/ConsoleApplication2/bin/Debug/ConsoleApplication2.EXE. The window has standard minimize, maximize, and close buttons. The main area is black with white text. The output consists of four lines: a=10, b=20.5, c=30, and d=40. A vertical scrollbar is visible on the right side of the console area.

```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication2/ConsoleApplication2/bin/Debug/ConsoleApplication2.EXE
a=10
b=20.5
c=30
d=40
```

## Practical No. 2.12

**Aim: Implementation of method overriding.**

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication2
{
    class A
    {
        virtual internal void show2()
        {
            Console.WriteLine("This is class A");
        }
    }
    class B : A
    {
        internal override void show2()
        {
            Console.WriteLine("This is class B");
        }
    }
    class C : B
    {
        internal void show2()
        {
            Console.WriteLine("This is class C");
        }
    }
    class Class1
    {
        static void Main(string[] args)
        {
            //C x = new C();
            //x.show2();
            A r = new B();
            r.show2();
            Console.ReadLine();
        }
    }
}
```

## Output:



A screenshot of a Windows console window. The title bar at the top shows the file path: `file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication2/ConsoleApplication2/bin/Debug/ConsoleApplication2.EXE`. The window has standard Windows window controls (minimize, maximize, close) on the right. The main area of the window is black, and the text `This is class B` is displayed in the top left corner in a light gray font. A vertical scrollbar is visible on the right side of the console area.

```
This is class B
```

### Practical No. 2.13

**Aim:** Write a program for pass by value.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication2
{
    class Class2
    {
        static void swap(int x, int y)
        {
            int z = x; x = y;
            y = z;
        }
        static void Main(string[] args)
        {
            int a = 10; int b = 20;
            Console.WriteLine("Before swap:");
            Console.WriteLine("a=" + a);
            Console.WriteLine("b=" + b);
            swap(a, b);
            Console.WriteLine("After swap:");
            Console.WriteLine("a=" + a);
            Console.WriteLine("b=" + b);
            Console.ReadLine();
        }
    }
}
```

**Output:**

A screenshot of a Windows console application window. The title bar shows the file path: file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication2/ConsoleApplication2/bin/Debug/ConsoleApplication2.EXE. The console output is as follows:  
Before swap:  
a=10  
b=20  
After swap:  
a=10  
b=20  
The window has a standard Windows interface with minimize, maximize, and close buttons in the title bar.



## Practical No. 2.14

**Aim:** Write a program for pass by Reference.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication2
{
    class Class3
    {
        static void swap(ref int x, ref int y)
        {
            int z = x;
            x = y;
            y = z;
        }
        static void Main(string[] args)
        {
            int a = 10; int b = 20;
            Console.WriteLine("Before swap:");
            Console.WriteLine("a=" + a);
            Console.WriteLine("b=" + b);
            swap(ref a, ref b);
            Console.WriteLine("After swap:");
            Console.WriteLine("a=" + a);
            Console.WriteLine("b=" + b);
            Console.ReadLine();
        }
    }
}
```

**Output:**

A screenshot of a Windows console application window. The title bar shows the file path: file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication2/ConsoleApplication2/bin/Debug/ConsoleApplication2.EXE. The console output is as follows:  
Before swap:  
a=10  
b=20  
After swap:  
a=20  
b=10  
The window has a black background and white text. The output demonstrates that the variables a and b were swapped successfully using pass-by-reference.

## Practical No. 2.15

**Aim:** Write a program for out parameter.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication2
{
    class Class4
    {
        static void square(int x, out int y)
        {
            y = x * x;
        }
        static void Main(string[] args)
        {
            int a = 10, b;
            square(a, out b);
            Console.WriteLine("Program using out parameter: b=" + b);
            Console.ReadLine();
        }
    }
}
```

**Output:**

A screenshot of a Windows console application window. The title bar shows the file path: file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication2/ConsoleApplication2/bin/Debug/ConsoleApplication2.EXE. The window has standard minimize, maximize, and close buttons. The console output is displayed on a black background with white text, showing the line: "Program using out parameter: b=100". A vertical scrollbar is visible on the right side of the console area.

## Practical No. 2.16

**Aim:** Write a program for parameterized array.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication3
{
    class ParamArray
    {
        public int AddElements(params int[] arr)
        {
            int sum = 0;
            foreach (int i in arr)
            {
                sum += i;
            }
            return sum;
        }
    }
    class Class7
    {
        static void Main(string[] args)
        {
            ParamArray app = new ParamArray();
            int sum = app.AddElements(512, 720, 250, 567, 889);
            Console.WriteLine("The sum is: {0}", sum);
            Console.ReadKey();
        }
    }
}
```

**Output:**

A screenshot of a Windows console application window. The title bar shows the file path: file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication3/ConsoleApplication3/bin/Debug/ConsoleApplication3.EXE. The console output displays the text "The sum is: 2938". The rest of the console area is black.

## Practical No. 2.17

**Aim:** Write a program for abstract class.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication2
{
    abstract class shape
    {
        protected int x,y;
        internal shape(int a, int b)
        {
            x = a; y = b;
        }
        abstract internal void area();
    }
    class rectangle : shape
    {
        internal rectangle(int a, int b) : base(a, b)
        {
        }
        internal override void area()
        {
            int ar = x * y;
            Console.WriteLine("Area of rectangle:" + ar);
        }
    }
    class triangle : shape
    {
        internal triangle(int a, int b) : base(a, b)
        {
        }
        internal override void area()
        {
            int ar = (x * y) / 2;
            Console.WriteLine("Area of triangle:" + ar);
        }
    }
    class Class6
    {
        static void Main(string[] args)
        {
            shape s = new rectangle(10, 20);
            s.area();
            shape s1 = new triangle(20, 10);
            s1.area();
            Console.ReadLine();
        }
    }
}
```

## Output:



A screenshot of a Windows console application window. The title bar shows the file path: file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication2/ConsoleApplication2/bin/Debug/ConsoleApplication2.EXE. The window contains two lines of text: "Area of rectangle:200" and "Area of triangle:100". The rest of the window is black.

```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication2/ConsoleApplication2/bin/Debug/ConsoleApplication2.EXE
Area of rectangle:200
Area of triangle:100
```

## Practical No. 2.18

**Aim:** Write a program for Interface.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication2
{
    interface arithmetic
    {
        int add(int a, int b); int sub(int a, int b); void mul();
    }
    class abc : arithmetic
    {
        public int add(int a, int b)
        {
            return a + b;
        }
        public int sub(int a, int b)
        {
            return a - b;
        }
        public void mul()
        {
            int a = 10;
            int b = 20;
            int ar = a * b;
            Console.WriteLine("Product:" + ar);
        }
    }
    class Class7
    {
        static void Main(string[] args)
        {
            abc ob = new abc();
            Console.WriteLine("Total:" + ob.add(5, 10));
            Console.WriteLine("Result:" + ob.sub(5, 3));
            ob.mul();
            Console.ReadLine();
        }
    }
}
```

## Output:

A screenshot of a Windows console application window. The title bar shows the file path: file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication2/ConsoleApplication2/bin/Debug/ConsoleApplication2.EXE. The window has standard minimize, maximize, and close buttons. The main area is black with white text. The output consists of three lines: 'Total:15', 'Result:2', and 'Product:200'. A vertical scrollbar is visible on the right side of the console area.

```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication2/ConsoleApplication2/bin/Debug/ConsoleApplication2.EXE
Total:15
Result:2
Product:200
```

## Practical No. 2.19

**Aim: Write a program for Interface.**

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication2
{
    public interface ITransactions
    {
        // interface members void showTransaction(); double getAmount();
    }
    public class Transaction : ITransactions
    {
        private string tCode;
        private string date;
        private double amount;
        public Transaction()
        {
            tCode = " ";
            date = " "; amount = 0.0;
        }
        public Transaction(string c, string d, double a)
        {
            tCode = c; date = d;
            amount = a;
        }
        public double getAmount()
        {
            return amount;
        }
        public void showTransaction()
        {
            Console.WriteLine("Transaction: {0}", tCode);
            Console.WriteLine("Date: {0}", date);
            Console.WriteLine("Amount: {0}", getAmount());
        }
    }
    class Class8
    {
        static void Main(string[] args)
        {
            Transaction t1 = new Transaction("001", "8/1/2019", 78900.00);
            Transaction t2 = new Transaction("002", "9/2/2019", 451900.00);

            t1.showTransaction();
            t2.showTransaction();
            Console.ReadKey();
        }
    }
}
```



## Output:

A screenshot of a Windows console application window. The title bar shows the file path: file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication2/ConsoleApplication2/bin/Debug/ConsoleApplication2.EXE. The window has standard minimize, maximize, and close buttons. The console output is as follows:

```
Transaction: 001  
Date: 8/1/2019  
Amount: 78900  
Transaction: 002  
Date: 9/2/2019  
Amount: 451900  
_
```

The text is white on a black background. A vertical scrollbar is visible on the right side of the console area.

## Practical No. 2.20

**Aim:** Write a program for Delegates.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication2
{
    public delegate void del1();
    class abc2
    {
        internal void display1()
        {
            Console.WriteLine("This is method 1");
        }
        internal void display2()
        {
            Console.WriteLine("This is method 2");
        }
    }
    class Class9
    {
        static void Main(string[] args)
        {
            abc2 a = new abc2();
            del1 d = new del1(a.display1);
            d();
            Console.ReadLine();
        }
    }
}
```

**Output:**



## Practical No. 2.21

**Aim:** Write a program for Multicast Delegate.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication3
{
    public delegate int del1(int x, int y);
    class abc
    {
        internal int add(int x, int y)
        {
            return x + y;
        }
        internal int sub(int x, int y)
        {
            return x - y;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            abc ob = new abc();
            del1 d = ob.add;
            Console.WriteLine("Add:" + d(5, 8));
            d += ob.sub;
            Console.WriteLine("Sub:" + d(8, 5));
            Console.ReadLine();
        }
    }
}
```

## Output:



```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication3/ConsoleApplication3/bin/Debug/ConsoleApplication3.EXE
Add:13
Sub:3
```

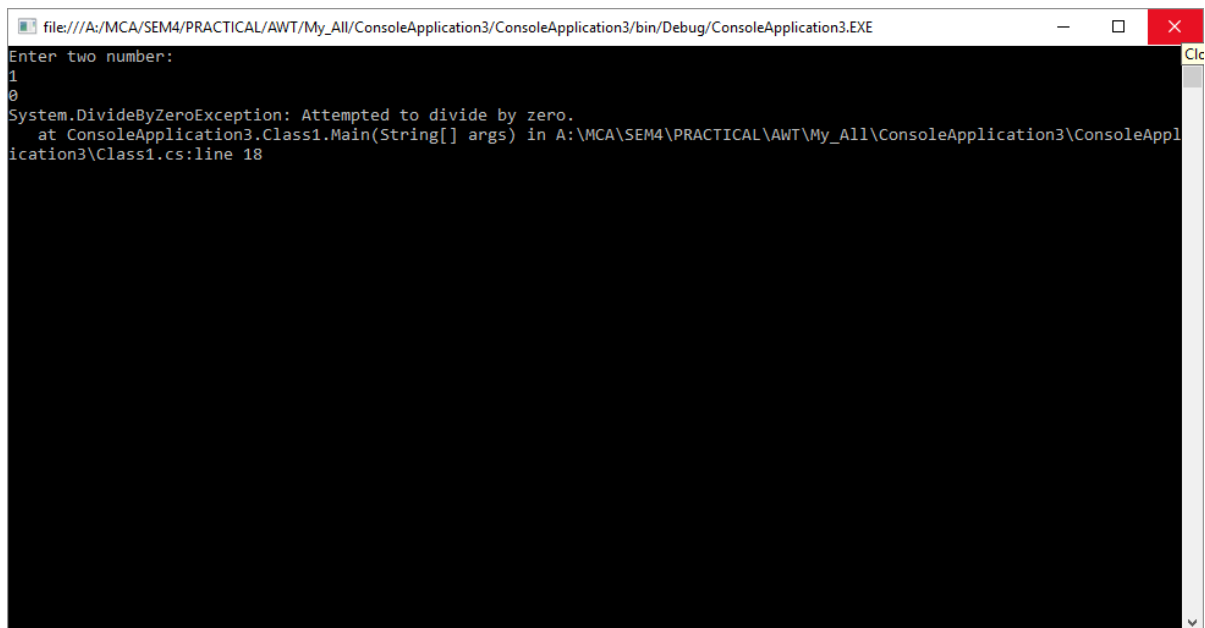
## Practical No. 2.22

**Aim:** Write a program for Exception Handling.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication3
{
    class Class1
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter two number:");
            int a = int.Parse(Console.ReadLine());
            int b = int.Parse(Console.ReadLine());
            try
            {
                int c = a / b;
                Console.WriteLine("c:" + c);
            }
            catch (Exception e)
            {
                Console.WriteLine(e);
            }
            Console.ReadLine();
        }
    }
}
```

**Output:**

A screenshot of a Windows console application window. The title bar shows the file path: file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication3/ConsoleApplication3/bin/Debug/ConsoleApplication3.EXE. The console output shows the prompt "Enter two number:" followed by the user input "1" and "0" on separate lines. Then, an exception is thrown: "System.DivideByZeroException: Attempted to divide by zero." followed by the stack trace: "at ConsoleApplication3.Class1.Main(String[] args) in A:\MCA\SEM4\PRACTICAL\AWT\My\_All\ConsoleApplication3\ConsoleApplication3\Class1.cs:line 18". The window has standard Windows controls (minimize, maximize, close) in the top right corner.

```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication3/ConsoleApplication3/bin/Debug/ConsoleApplication3.EXE
Enter two number:
1
0
System.DivideByZeroException: Attempted to divide by zero.
   at ConsoleApplication3.Class1.Main(String[] args) in A:\MCA\SEM4\PRACTICAL\AWT\My_All\ConsoleApplication3\ConsoleApplication3\Class1.cs:line 18
```

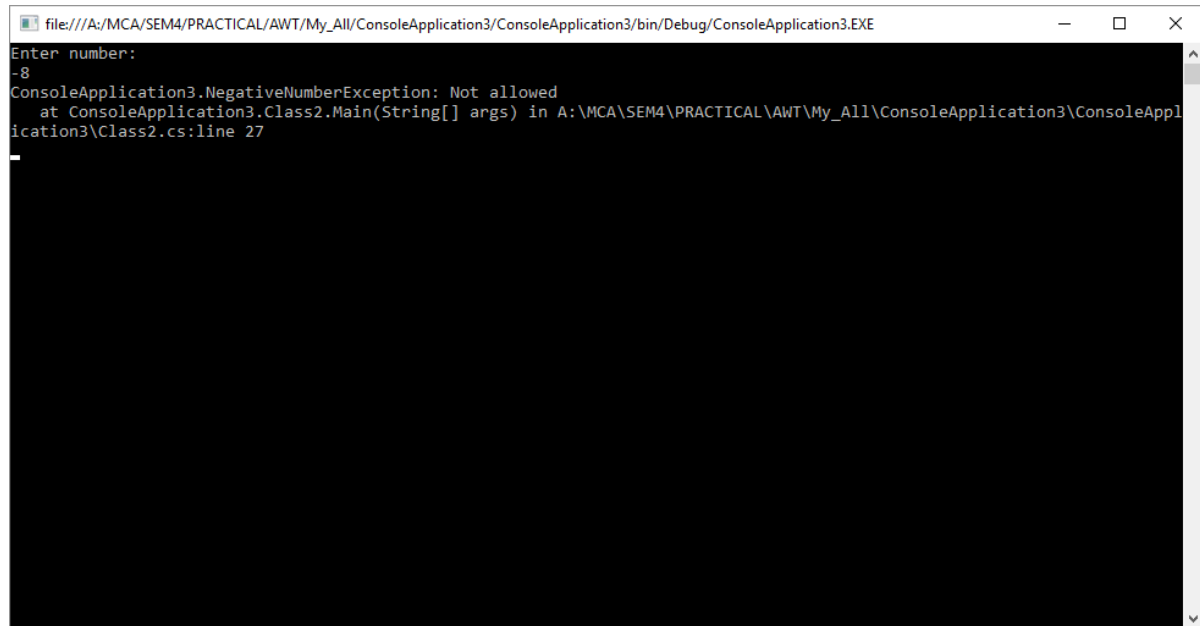
## Practical No. 2.23

**Aim:** Write a program for Exception Handling.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication3
{
    class NegativeNumberException: Exception
    {
    public NegativeNumberException(String s) : base(s)
    {
    }
    }
    class Class2
    {
    static void Main(string[] args)
    {
    Console.WriteLine("Enter number:");
    int a = int.Parse(Console.ReadLine());
    if (a > 0)
    Console.WriteLine("a:" + a);
    else
    try
    {
    allowed");          throw new NegativeNumberException("Not
    }
    catch (NegativeNumberException e)
    {
    Console.WriteLine(e);
    }
    Console.ReadLine();
    }
    }
}
```

## Output:



A screenshot of a Windows console application window. The title bar shows the file path: `file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication3/ConsoleApplication3/bin/Debug/ConsoleApplication3.EXE`. The window has standard minimize, maximize, and close buttons. The console output is as follows:

```
Enter number:
-8
ConsoleApplication3.NegativeNumberException: Not allowed
   at ConsoleApplication3.Class2.Main(String[] args) in A:\MCA\SEM4\PRACTICAL\AWT\My_All\ConsoleApplication3\ConsoleApp
lication3\Class2.cs:line 27
```

The console text is white on a black background. There is a small white cursor at the end of the first line of output.

## Practical No. 2.24

**Aim:** Write a program for ArrayList.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Collections;
namespace ConsoleApplication3
{
    class Class3
    {
        static void Main(string[] args)
        {
            ArrayList s = new ArrayList();
            s.Add("10");
            s.Add("abc");
            s.Add("90.5");
            foreach (string a in s)
                Console.WriteLine(a);
            s.Sort();
            Console.WriteLine("aftersort");
            foreach (string a in s)
                Console.WriteLine(a);
            s.Insert(1, "pqr");
            Console.WriteLine("after insert");
            foreach (string a in s)
                Console.WriteLine(a);
            s.RemoveAt(0);
            Console.WriteLine("after remove");
            foreach (string a in s)
                Console.WriteLine(a);
            Console.ReadLine();
        }
    }
}
```



## Output:



A screenshot of a Windows console window. The title bar shows the file path: file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_Alt/ConsoleApplication3/ConsoleApplication3/bin/Debug/ConsoleApplication3.EXE. The console output is as follows:

```
10
abc
90.5
after sort
10
90.5
abc
after insert
10
pqr
90.5
abc
after remove
pqr
90.5
abc
_
```

## Practical No. 2.25

**Aim:** Write a program for List.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    class student
    {
        int roll; string nm;
        public int rollno
        {
            get
            {
                return roll;
            }
            set
            {
                roll = value;
            }
        }
        public string
        name
        {
            get
            {
                name = value;
            }
            set
            {
                return nm;
            }
        }
        public student(int roll, string nm)
        {
            this.roll = roll;
            this.nm = nm;
        }
    }
    class Class4
    {
        static void Main(string[] args)
        {
            List<student> s = new List<student>();
            s.Add(new student(1, "Ankita"));
            s.Add(new student(2, "Nikita"));
            foreach (var e in s)
            {
                Console.WriteLine(e.rollno);
                Console.WriteLine(e.name);
            }
            Console.ReadLine();
        }
    }
}
```

```
}  
}  
}
```

## Output:



A screenshot of a Java console application window. The title bar shows the file path: `file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication3/ConsoleApplication3/bin/Debug/ConsoleApplication3.EXE`. The window contains the following output:

```
1  
Ankita  
2  
Nikita
```

## Practical No. 2.26

**Aim:** Write a program for stack.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication3
{
    class Test<T>
    {
        T _value;
        public Test(T t)
        {
            this._value = t;
        }
        public void write()
        {
            Console.WriteLine(this._value);
        }
    }
    class Class5
    {
        static void Main(string[] args)
        {
            Test<int> test1 = new Test<int>(5);
            test1.write();
            Test<String> test2 = new Test<string>("Ankita");
            test2.write();
            Console.ReadLine();
        }
    }
}
```

## Output:



```
5
Ankita

```

## Practical No. 2.27

**Aim:** Write a program for constraints.

**Source code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication3
{
    class abc<T> where T : class
    {
        T s;
        public abc(T a)
        {
            s = a;
        }
        public void show()
        {
            Console.WriteLine("Constraint:\n"+s);
        }
    }
    class Class6
    {
        static void Main(string[] args)
        {
            abc<String> ob = new abc<string>("ABC");
            ob.show();
            Console.ReadLine();
        }
    }
}
```

**Output:**

A screenshot of a Windows console application window. The title bar shows the file path: file:///A:/MCA/SEM4/PRACTICAL/AWT/My\_All/ConsoleApplication3/ConsoleApplication3/bin/Debug/ConsoleApplication3.EXE. The window has standard minimize, maximize, and close buttons. The console output is displayed on a black background with white text. The first line is "Constraint:" and the second line is "ABC". A vertical scrollbar is visible on the right side of the console area.

```
file:///A:/MCA/SEM4/PRACTICAL/AWT/My_All/ConsoleApplication3/ConsoleApplication3/bin/Debug/ConsoleApplication3.EXE
Constraint:
ABC
```