Practical No. 2

Working with Object Oriented C# and ASP.NET (Console Application).

a) Create a simple application to perform following operations:

1. Finding Factorial of No.

Program.cs

```
using System;
namespace Fact
{
   class FactorialProgram
   {
      static void Main(string[] args)
      {
         int i, fact = 1, num;
         Console.WriteLine("Enter any number:");
         num = int.Parse(Console.ReadLine());
         for (i = 1; i <= num; i++)
         {
            fact = fact * i;
         }
         Console.Write("Factorial of " + num + " is: " + fact);
         Console.ReadKey();
      }
   }
}
```

Output:

```
Enter any number:
5
Factorial of 5 is: 120
```

2. Money Conversion.

Program.cs

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

```
using System.Text;
using System.Threading.Tasks;

namespace MoneyConversion
{
   class Program
   {
      static void Main(string[] args)
      {
         int ch;
         double rupe;
         Console.WriteLine("Enter your Choice :\n 1: Dollar to Rupee \n 2: Euro to Rupee \n 3:
British pounds to Rupee \n 4: Yen to Rupee ");
         Console.WriteLine("\nEnter your choice: ");
         ch = int.Parse(Console.ReadLine());
         switch (ch)
         {
            case 1:
               Double dollar, val = 83.96;
               Console.WriteLine("Enter the amount in Dollar :");
               dollar = Double.Parse(Console.ReadLine());
               rupe = dollar * val;
               Console.WriteLine("Amount in rupees: " + rupe);
               break;
            case 2:
               Double Euro, value = 92.67;
               Console.WriteLine("Enter the amount in Euro :");
               Euro = Double.Parse(Console.ReadLine());
               rupe = Euro * value;
               Console.WriteLine("Amount in rupees " + rupe);
               break;
            case 3:
               Double bpound, valu = 109.93;
               Console.WriteLine("Enter the amount in British Pound :");
               bpound = Double.Parse(Console.ReadLine());
               rupe = bpound * valu;
               Console.WriteLine("Amount in rupees " + rupe);
               break;
            case 4:
               Double jyen, valuee = 0.58;
               Console.WriteLine("Enter the amount in Japanese Yen :");
               jyen = Double.Parse(Console.ReadLine());
               rupe = jyen * valuee;
               Console.WriteLine("Amount in rupees: " + rupe);
               break;
            default:
               Console.WriteLine("Invalid choice, Please enter a number between 1 to 4.");
               break;
         }
         Console.ReadKey();
```

```
    }
  }
}
```

Output:

```
Enter your Choice :
 1: Dollar to Rupee
 2: Euro to Rupee
 3: British pounds to Rupee
 4: Yen to Rupee

Enter your choice:
1
Enter the amount in Dollar :
5
Amount in rupees: 419.8
```

```
Enter your Choice :
 1: Dollar to Rupee
 2: Euro to Rupee
 3: British pounds to Rupee
 4: Yen to Rupee

Enter your choice:
2
Enter the amount in Euro :
5
Amount in rupees 463.35
```

```
Enter your Choice :
 1: Dollar to Rupee
 2: Euro to Rupee
 3: British pounds to Rupee
 4: Yen to Rupee

Enter your choice:
3
Enter the amount in British Pound :
5
Amount in rupees 549.65
```

```
Enter your Choice :
 1: Dollar to Rupee
 2: Euro to Rupee
 3: British pounds to Rupee
 4: Yen to Rupee

Enter your choice:
4
Enter the amount in Japanese Yen :
5
Amount in rupees: 2.9
```

3. Temperature Conversion.

i. Celsius to Fahrenheit.

Program.cs

```
using System;
namespace TemperatureConversion
{
   class Program
   {
      static void Main(string[] args)
      {
         double farenheit, Celsius;
         Console.Write("Enter temperature in Celsius:");
         Celsius = Convert.ToDouble(Console.ReadLine());
         Console.WriteLine("Celcius: " + Celsius);
         farenheit = (Celsius * 9) / 5 + 32;
         Console.WriteLine("Farenheit: " + farenheit);
         Console.ReadKey();
      }
   }
}
```

Output:

```
Enter temperature in Celsius:35
Celcius: 35
Farenheit: 95
```

ii. Fahrenheit to Celsius.

Program.cs

```
using System;
namespace TemperatureConversion
{
   class Program
   {
      static void Main(string[] args)
      {
         double Celsius, fahrenheit;
         Console.Write("Enter temperature in Fahrenheit: ");
```

```
            fahrenheit = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine("fahrenheit: " + fahrenheit);
            Celsius = (fahrenheit - 32) * 5 / 9;
            Console.WriteLine("Celsius: " + Celsius);
            Console.ReadKey();
        }
    }
}
```

Output:

```
Enter temperature in Fahrenheit: 96
fahrenheit: 96
Celsius: 35.5555555555556
```

b) Create a simple application to demonstrate the use of the following concept:

1. Function Overloading.

Program.cs

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

namespace FunctionOverload
{
    class addition
    {
        public void add(int a)
        {
            int sum = a + a;
            Console.WriteLine("The addition is: " + sum);
        }
        public void add(int a, int b)
        {
            int s = a + b;
            Console.WriteLine("The addition is: " + s);
        }
        public void add(int a, int b, int c)
        {
```

```
            int z = a + b + c;
            Console.WriteLine("The addition is: " + z);
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            addition n = new addition();
            n.add(10);
            n.add(10, 20);
            n.add(10, 20, 30);
            Console.ReadKey();
        }
    }
}
```

Output:

```
The addition is: 20
The addition is: 30
The addition is: 60
```

2. Inheritance.

i. Single Inheritance.

Program.cs

```
using System;
namespace SingleInheritance
{
    public class Animal
    {
        public void eat()
        {
            Console.WriteLine("Eating.....");
        }
    }
    public class Dog : Animal
    {
        public void bark()
        {
            Console.WriteLine("Barking.....");
```

```
        }
    }

    class Program
    {
        public static void Main(string[] args)
        {
            Dog d = new Dog();
            d.bark();
            d.eat();
            Console.ReadKey();
        }
    }
}
```

Output:

```
Barking.....
Eating.....
```

ii. Multi Level Inheritance.

Program.cs

```
using System;
namespace MultiLevelInheritance
{
    public class Animal
    {
        public void eat()
        {
            Console.WriteLine("Eating.....");
        }
    }
    public class Dog : Animal
    {
        public void bark()
        {
            Console.WriteLine("Barking.....");
        }
    }
    public class BabyDog : Dog
    {
```

```
        public void weep()
        {
            Console.WriteLine("Weeping...");
        }
    }

    class Program
    {
        public static void Main(string[] args)
        {
            BabyDog d = new BabyDog();
            d.weep();
            d.bark();
            d.eat();
            Console.ReadKey();
        }
    }
}
```

Output:

```
Weeping...
Barking.....
Eating.....
```

3. Constructor Overloading.

Program.cs

```
using System;

namespace Constructor_Overload
{
    public class Car
    {
        // constructor with no parameter
        public Car()
        {
            Console.WriteLine("Car constructor");
        }

        // constructor with one parameter
        public Car(string brand)
        {
```

```
            Console.WriteLine("Car constructor with one parameter");
            Console.WriteLine("Brand: " + brand);
        }
    }

    class Demo
    {
        static void Main(string[] args)
        {
            // call with no parameter
            Car car = new Car();
            Console.WriteLine();

            // call with one parameter
            Car car2 = new Car("Bugatti");
            Console.ReadKey();
        }
    }
}
```

Output:

```
Car constructor

Car constructor with one parameter
Brand: Bugatti
```

4. Interfaces.

Program.cs

```
using System;
namespace CSharpInterface
{
    interface IPolygon
    {
        void calculateArea(int l, int b);
    }
    class Rectangle : IPolygon
    {
        public void calculateArea(int l, int b)
        {
            int area = l * b;
            Console.WriteLine("Area of Rectangle: " + area);
```

```
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Rectangle r1 = new Rectangle();
            r1.calculateArea(100, 200);
            Console.ReadKey();
        }
    }
}
```

Output:

```
Area of Rectangle: 20000
```

c) Create a simple application to demonstrate the use of the following concept:

1. Using delegates & Events.

Program.cs

```
using System;
using System.Collections.Generic;
class Program
{
    static int calculateSum(int x, int y)
    {
        return x + y;
    }
    public delegate int myDelegate(int num1, int num2);
    static void Main()
    {
        myDelegate d = new myDelegate(calculateSum);
        int result = d(5, 5);
        Console.WriteLine(result);
        Console.ReadKey();
    }
}
```

Output:

```
10
```

2. Exception Handling.

i. Index Out of Range Exception.

Program.cs

```
using System;

namespace ExceptionExample
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] colors = { "Red", "Blue", "Green" };
            try
            {
                Console.WriteLine(colors[5]);
            }
            catch (IndexOutOfRangeException e)
            {
                Console.WriteLine("An exception occurred: " + e.Message);
            }
            Console.ReadKey();
        }
    }
}
```

Output:

```
An exception occurred: Index was outside the bounds of the array.
```

ii. Divide by Zero Exception.

Program.cs

```
using System;
using System.Collections.Generic;
```

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

namespace ExcepHand
{
   public class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("Enter first number: ");
         int fno = int.Parse(Console.ReadLine());
         Console.WriteLine("Enter second number: ");
         int sno = int.Parse(Console.ReadLine());
         try
         {
            int divRes = fno / sno;
            Console.WriteLine("Division of two numbers is: " + divRes);
         }
         catch (Exception e)
         {
            Console.WriteLine("An exception occurred: " + e.Message);
         }
         finally
         {
            Console.WriteLine("Sum of two numbers is: " + (fno + sno));
         }
         Console.ReadKey();
      }
   }
}
```

Output:

```
Enter first number:
10
Enter second number:
0
An exception occurred: Attempted to divide by zero.
Sum of two numbers is: 10
```