Lecture Two

Object-Oriented Class: A Fundamental Concept in Object-Oriented Programming

1. Definition of a Class

A class can be formally defined as a programming construct that encapsulates data for an object and methods to manipulate that data. It serves as a template from which multiple objects can be instantiated. Each object created from a class inherits its structure and behavior, but can maintain distinct values for its attributes.

2. Structure of a Class

A typical class includes three primary components:

a. *Attributes* (Data Members): variables that define the properties or state of an object. Some called them *fields* or *class feature*.

b. *Methods*: procedures that define the behaviors or actions of an object.

c. *Constructors*: special methods used to initialize objects upon creation. These elements together define how objects behave and interact within an object-oriented system.

3. Example Implementation

1. Write a C# complete code using a console application to create a class entitled **Convert2h** contains:

- ✓ **Fields**:
  - **seconds.**
- ✓ **Constructor.**
- ✓ **Methods:**
  - **con2h** (divide seconds by 3600) //method1
  - **con2m** (divide seconds by 60) //method2

Solution:

```
class Convert2h
    {
        // Field to store seconds
        private double seconds;

        // Constructor to initialize the field
        public Convert2h(double sec)
        {
            seconds = sec;
        }

        // Method 1: Convert seconds to hours
        public double Con2h()
        {
            return seconds / 3600;
        }

        // Method 2: Convert seconds to minutes
        public double Con2m()
        {
            return seconds / 60;
        }
    }
```

```
public static void Main(string[] args)
    {
        Console.Write("Enter number of seconds: ");
        double sec = Convert.ToDouble(Console.ReadLine());

         // Create an object of Convert2h
         Convert2h converter = new Convert2h(sec);

         // Display results
         Console.WriteLine("Seconds in hours: " + converter.Con2h());
         Console.WriteLine("Seconds in minutes: " + converter.Con2m());
         Console.ReadLine();
    }
```

Run

```
Enter number of seconds: 3600
Seconds in hours: 1
Seconds in minutes: 60
```

2. Write a C# complete code using a console application to create a class entitled **Circle** contains:
   - ✓ **Fields:**
     - **r**
   - ✓ **Constructor.**
   - ✓ **Methods**:
     - **area** (r*r*3.14) //method1
     - **pre** (2*r*3.14)//method2

Solution:

```
class Circle
    {
        // Field to store radius
        private double r;

        // Constructor to initialize the radius
        public Circle(double radius)
        {
            r = radius;
 }

// Method 1: Calculate area of the circle
public double area()
 {
    return r * r * 3.14;
 }

// Method 2: Calculate perimeter (circumference) of the circle
public double pre()
 {
    return 2 * r * 3.14;
 }
}
```

```
1.  public static void Main(string[] args)
2.      {
3.          Console.WriteLine("First Object");
4.          Console.Write("Enter the radius of the circle: ");
5.          double radius = Convert.ToDouble(Console.ReadLine());
6.
7.          // Create a Circle object
8.          Circle c1 = new Circle(radius);
9.
10.         // Display results
11.         Console.WriteLine("Area of the circle = " + c1.area());
12.         Console.WriteLine("Perimeter of the circle = " + c1.pre());
13.
14.         // Create a Circle object
15.         Console.WriteLine("Second Object");
16.         Circle c2 = new Circle(100);
17.
18.         // Display results
19.         Console.WriteLine("Area of the circle = " + c2.area());
20.         Console.WriteLine("Perimeter of the circle = " + c2.pre());
21.     }
```

Run

```
First Object
Enter the radius of the circle: 200
Area of the circle = 125600
Perimeter of the circle = 1256
Second Object
Area of the circle = 31400
Perimeter of the circle = 628
```

3. Write a C# complete code using a console application to create a class entitled **Employee** contains:
   - ✓ **Fields**:
     - **name.**

- **birthYear**

✓ **Constructor.**

✓ **Methods:**

- **Age (2025-birthYear)** check if age greater than 60 write retired else in service. //method1
- **display** //method2

Solution

```
1.  public class Employee
2.      {
3.          // Fields
4.          public string name;
5.          public int birthYear;
6.
7.          // Constructor
8.          public Employee(string name, int birthYear)
9.          {
10.             this.name = name;
11.             this.birthYear = birthYear;
12.         }
13.
14.         // Method 1: Calculate age and check retirement status
15.         public void Age()
16.         {
17.             int age = 2025 - birthYear;
18.             Console.WriteLine($"Age: {age}");
19.
20.             if (age > 60)
21.                 Console.WriteLine("Status: Retired");
22.             else
23.                 Console.WriteLine("Status: In Service");
24.         }
25.
26.         // Method 2: Display employee information
27.         public void Display()
28.         {
29.             Console.WriteLine("---- Employee Information ----");
30.             Console.WriteLine("Name: "+name);
31.             Console.WriteLine("Birth Year: "+birthYear);
```

لاحظ عند استخدام نفس المتغيرات بالحقول نستخدم الكلمة المفتاحية this

```
32.         }
33.     }
```

```
1.  public static void Main(string[] args)
2.      {
3.          // Create an Employee object
4.          Console.Write("Enter employee name: ");
5.          string name = Console.ReadLine();
6.
7.          Console.Write("Enter birth year: ");
8.          int year = Convert.ToInt32(Console.ReadLine());
9.
10.         // Create employee instance
11.         Employee emp = new Employee(name, year);
12.
13.         // Display info and check age
14.         emp.Display();
15.         emp.Age();
16.
17.         Console.ReadKey(); // Wait for user input before closing
18.     }
```

Run

```
Enter employee name: Ali Bader
Enter birth year: 1995
---- Employee Information ----
Name: Ali Bader
Birth Year: 1995
Age: 30
Status: In Service
```