Lesson 20 - Classes & Objects I

Classes

We have seen a lot of built-in or *primitive* type variables, which we use to store single values, and later we used Arrays, which were collections, and had some methods and properties.

In Object Oriented Programming, a `class` is a datatype that defines a set of variables and methods for a declared object. A `class` is a blueprint, and an `object` is an instance generated from that blueprint. The blueprint defines data and behavior for a type.

General syntax for a class defintion is as follows:

```
class Blueprint
{
        // variables, properties, methods.
}
```

Objects

The `type` is the class name, and when we create a variable of a class `type`, then we call that variable an `object`.

```
NOTE:
A class is NOT a variable. However, based on a class you can create an object of
that class and save it to a variable. Objects are also called instances of a
class.
```

Each object has its own characteristics, and these are called properties. Values of these properties describe the current state of the object.

Class Example

```
class Student
{
    public int Age;
    public string Name;
    public int[] Grades = new int[3];
```

```
    public void DisplayGrades()
    {
        Console.WriteLine("Student: " + Name);
        foreach (var grade in Grades)
            Console.WriteLine(grade);
    }
}
```

```
NOTE: class names and class variables use UpperCamelCase
```

Object initialization is as follows:

```
Student student = new Student();
```

You can skip the type on right hand side if var isn't used. We can also access the variables and methods that have been defined as public.

```
Student student = new();
student.Name = "Talha";
student.Age = 21;
student.Grades = new int[] { 5, 6, 10 };
```

This can also be further simplified as,

```
Student student = new()
{
    Name = "Talha",
    Age = 21,
    Grades = new int[] { 5, 6, 10 }
};
```

We can also call its method using the `dot` operator just like how we used to do for arrays.

```
student.DisplayGrades();
```

```
OUTPUT:
Student: Talha
5
6
10
```

```
public class Letter
{
```

```
        public string Author;
        public string Recipient;
        public string Body;
        public int Date;
        public int Month;
        public int Year;

        public void DisplayLetter()
        {
                Console.WriteLine("FROM: " + Author);
                Console.WriteLine("TO: " + Recipient);
                Console.WriteLine($"{Date}/{Month}/{Year}");
                Console.WriteLine("BODY: \n" + Body);
        }
}
```

```
public class Calculator
{
        private double Mem1;
        private double Mem2;
        private char Operator;

        private double CalculateFromMemory()
        {
                double result = Operator switch
                {
                        '+' => Mem1 + Mem2,
                        '-' => Mem1 - Mem2,
                        '*' => Mem1 * Mem2,
                        '/' => Mem1 / Mem2,
                        '%' => Mem1 % Mem2,
                        _ => 0
                };
                Console.WriteLine($"{Mem1} {Operator} {Mem2} = {result}");
                return result;
        }

        public double Calculate(double num1, char op, double num2)
        {
                Mem1 = num1;
                Mem2 = num2;
                Operator = op;

                return CalculateFromMemory();
        }
}
```

```
Letter coverLetter = new()
{
    Author = "Talha",
    Recipient = "Amad",
    Body = "Please give me the job.",
    Date = 27,
    Month = 11,
    Year = 2022,
};
coverLetter.DisplayLetter();

Console.WriteLine();

Letter responseLetter = new()
{
    Author = "Amad",
    Recipient = "Talha",
    Body = "Sorry, try to write a more convincing letter next time.",
    Date = 28,
    Month = 11,
    Year = 2022
};
responseLetter.DisplayLetter();

Student dullStudent = new()
{
    Name = "Amjad",
    Age = 14,
    Grades = new int[] { 1, 3, 2 }
};

Student brightStudent = new()
{
    Name = "Talal",
    Age = 12,
    Grades = new int[] { 10, 9, 10 }
};

dullStudent.DisplayGrades();
brightStudent.DisplayGrades();

Calculator calc = new();

calc.Calculate(10, 'r', 4);
```