

OOP – Lesson 5

Class

- **Class** : It is the basic unit of the OOP which is used to wrap up (تغليف) the data and the codes (encapsulation).
- The code and data that form the class are called the members of the class.
- A class is defined as a data structure used to create object.

Declaring and Defining Classes

To define a new type or class, first declare it, and then define its methods and fields.

- Declare a class using the **class** keyword.

The syntax is as follows:

```
[access-modifiers] class <classname> [:base-class [,interface(s)]]
{
    class-body //It contents
                field, properties, methods, event(members)
}
```

Member Definitions

All members have their own accessibility level, defined in all cases by one of the following keywords:

public: Member is accessible (يمكن الوصول اليه) from any code

private: Member is accessible only from code that is part of the class (the default if no keyword is used)

protected: Member accessible only from code that is part of either the class or a derived class

Example 1 :

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

```
public class Time
{
```

```
// private variables
private int Year;
private int Month;
private int Date;
private int Hour;
private int Minute;
private int Second;

// public accessor methods
public void DisplayCurrentTime( )
{
    System.Console.WriteLine( "{0}/{1}/{2} {3}:{4}:{5}",
        Month, Date, Year, Hour, Minute, Second );
}

static void Main( )
{
    Time t = new Time( );
    t.DisplayCurrentTime( );
}
}
```

Object

- An instances of a class are called objects (instantiated).
- Objects are created in memory when your program executes.

Creating Objects

-Objects can be created using the **new** keyword followed by the name of the class that the object will be based upon, like this:

Time t = new Time();

The Lifecycle of an Object (دورة حياة الكائن)

The lifecycle includes *two* important stages: **Construction** and **Destruction**.

Construction:

When an object is first instantiated it needs to be initialized. This initialization is known as construction and is carried out by a *constructor* function.

عند تكوين الكائن من الضروري أن يُهيئ بإعطائه قيمة ابتدائية، وهذه التهيئة تدعى بالبناء ويُنفَّذ من قبل دالة البناء.

Basic initialization of an object is automatic.

Table . Primitive types and their default values	
Type	Default value
numeric (int, long, etc.)	0
bool	false
char	'\0' (null)
enum	0
reference	null

1- All objects have a *default constructor*, which is a parameterless method with the same name as the class itself.

EX: *Time object1=new Time ();*

2- In addition, a class definition might include several constructor methods with parameters, known as *nondefault constructors*, that uses a parameter at instantiation:

Time object1=new Time(CurrentTime);

To define a constructor, declare a method whose name is the same as the class in which it is declared. Constructors have no return type and are typically declared public. If there are arguments to pass, define an argument list just as you would for any other method.

Construction

EX1 :

```
using System;
class xxx
{
    public int x; // int x; or private int x;
    public int y;
```

```

    public xxx(int p1, int p2)
    {
        x = p1;
        y = p2;
    }

    static void Main()
    {
        xxx mC = new xxx(1, 2);
        Console.WriteLine("x = {0}, y = {1}", mC.x, mC.y);
    }
}

```

EX2:

```

using System;
public class Test
{
    class xxx
    {
        public int x; // int x
        public int y;

        public xxx(int p1, int p2)
        {
            x = p1;
            y = p2;
        }
    }

    static void Main()
    {
        xxx mC = new xxx(11, 22);
        Console.WriteLine("x = {0}, y = {1}", mC.x, mC.y);
    }
}

```

Output:

x = 11, y = 22

EX3

```

using System;
class Circle
{

```

```

const double PI = 3.141592;
public Circle() // default constructor
{
    radius = 5;
}

public double Area()
{
    // return 3.141592 * radius * radius;
    return PI * radius * radius;
}

private double radius;

static void Main()
{
    Circle c;
    c = new Circle();
    double areaOfCircle = c.Area();
    Console.WriteLine(areaOfCircle);
}
}

```

اضف الـ **method** التالية إلى البرنامج السابق واستخدمها في الـ **main**.

```

public Circle(double initialRadius) // overloaded constructor
{
    radius = initialRadius;
}

```