C#

Part 3

Agenda

- Class
- Inheritance
- Polymorphism
- Interface
- Exception
- File

Class

- Classes and objects are the two main aspects of object-oriented programming.
- Class is a template for objects, and an object is an instance of a class.

```
public class Computer
{                                                   1
    public string Brand = string.Empty;

    0 references
    public Computer()                               2
    {
    }
                                                    3
    0 references
    public void Details(string BrandName, string RamSize, string HarddiskSize)
    {
        Console.WriteLine("Brand Name :  {0}, Ram Szie : {1}, Hard Disk Size  : {2}",
                                                BrandName, RamSize, HarddiskSize);
    }

    private string ProcessName;

    2 references
    public string DisplaySize                       4
    {
        get { return DisplaySize; }
        set { DisplaySize = value; }
    }
}
```

```
1 --> Access Modifier
2 --> Constructor
3 --> Method
4 --> Property
```

Inheritance

- Inherit fields and methods from one class to another.
- Two categories in inheritance
- Derived Class
- Base class

```
0 references
public class Dell : Computer
{
    public string BrandBName = "Dell";
}
```

Polymorphism

- Poly" means many and "morph" means forms
- There are two types of polymorphism in C#:
- Static / Compile Time Polymorphism.
- Dynamic / Runtime Polymorphism.

Abstract

- Data **abstraction** is the process of hiding certain details and showing only essential information to the user.
- The `abstract` keyword is used for classes and methods:
- **Abstract class:** is a restricted class that cannot be used to create objects (to access it, it must be inherited from another class).
- **Abstract method:** can only be used in an abstract class, and it does not have a body. The body is provided by the derived class (inherited from).

Interfaces

- An `interface` is a completely "**abstract class**", which can only contain abstract methods and properties
- To access the interface methods, the interface must be "implemented" by another class

Files

- The `File` class from the `System.IO` namespace, allows us to work with files:
- `using System.IO;  // include the System.IO namespace`

Exceptions

- When an error occurs, C# will normally stop and generate an error message. The technical term for this is: C# will throw an **exception**
- **Syntax**

```
try
{
  //  Block of code to try
}
catch (Exception e)
{
  //  Block of code to handle errors
}
```

Thanks!

Contact us:

training@apps2fusion.com

+44 207 101 9262

+ 1 212 404 1735

www.apps2fusion.com