C# INTERMEDIATE

Access Modifiers

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Intermediate\_11\_Access\_Modifiers

{

//\*\*\*\*\*\*\*\*\*\*\*\*

// NOTES

//\*\*\*\*\*\*\*\*\*\*\*\*

// THE TYPES OF ACCESS MODIFFIERS

// PUBLIC

// PRIVATE

// PROTECTED

// INTERNAL

// PROTECTED INTERNAL

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// WHAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// An Access Modifier is a way to control access to a class and/or its members.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// WHY

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// To create safety in our programs.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// OBJECT ORIENTED PROGRAMMING

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// There are 3 major concepts in Object Oriented Programming

// Encapsulation (Information Hiding)

// Inheritance

// Polymorphism

// NOTE: When designing Classes you want to design them so that they only do one thing

// and they do that thing perfectly.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// ENCAPSULATION

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// When you want to hide some details of a class from the other classes

// Define fields as private

// EXAMPLE

// Below, the Name field of the Person class is set to private

// so that no other classes have access to it.

// The only way to change the value of that Name field is to

// use the public methods SetName and GetName

/\*

public class Person

{

private string Name;

public void SetName(string name)

{

if (!String.IsNullOrEmpty(name))

this.Name = name;

}

public string GetName()

{

return Name;

}

}

\*/

public class Person

{

// The convention is that class field names usually start with an underscore

// and in camel case.

// \*\*\* NOTE \*\*\*:

// One of the major reasons for creating methods that manipulate the

// \_birthdate property is because of ENCAPSULATION. We need to hide the details of

// the class from external access.

private DateTime \_birthdate;

public void SetBirthdate(DateTime birthdate)

{

\_birthdate = birthdate;

}

public DateTime GetBirthDate()

{

return \_birthdate;

}

}

class Program

{

static void Main(string[] args)

{

var person = new Person();

person.SetBirthdate(new DateTime(1982, 1, 1));

Console.WriteLine(person.GetBirthDate());

}

}

}