C# essentials

Like JAVA, C# is a strictly object-oriented language, so all the codes will be contained within class files.

namespace enables you to manage your code within worrying about class name collisions.

The first thing to know about variables in C sharp is language is strongly typed, meaning you must declare the data type that will go into that variable and you are allowed to use only that type for the duration of the variables lifecycle.

C# variables or primitive data types:

var, bool, string, int, uint (unsigned int, meaning only positive int), short, long, ushort, ulong, float, double

Eg: float num1 = 3.14f; var num2 = 3.14d; int num3 = int.MaxValue

If we look at the last example, we will notice that we treat everything as an object in C#, even including very primitive data types. .MaxValue is just a **method** or **property** of int object.

The above primitive data types just like atoms in Chemistry class and cannot be dis-constructed into smaller pieces. These primitive data types are what makes the more complicated structures and objects.

Create a constant number: const float pi = 3.14fl;

In a C# class, there could be properties, methods/functions, constructors and so on.

Properties:

For properties, we normally use getters and setters to get and set the value of the property.

Access modifiers:

public variable means that it can be seen outside the class, while private variable means that is cannot be seen outside the class.

Constructor:

Normally add the constructor at the end of the class???

Methods:

Methods in a class are actually functions in a class. It is more professional to call it a method than to call it function. Similar with the properties of a class, we need to return something in a method in order to exit the method.

Function bodied expression:

A function bodied expression is a different way of writing a method, very similar with Javascript, which used an arrow function.

Eg: public static float avgThreeScores(float 1, float b, float c) => (a + b + c) / 3;

If we use set a method to be static, then we can see the method outside the class without instantiating it.

Inherent a class in another class means that the another class can use the properties in the inherented class.

Async in C#

So our program, a collection of statements packaged in an EXE after it's complied, we take a program like that, load it into memory, and then we want to execute the statements in that program and we typically going to start in C# with static void main. Now, the thing that's running that code, we usually can call it a thread for now, so we have the operating system running our hardware within that operating system.

Asynchronos programing means more than one the code will be excuting. That means we need more than one thread. So async programing requires multi-threading.

Thread safety when two threads can run through a piece of code safely at the same time, without interfering with each other.

We want to use Async when we don’t want to wait for my code to be excuted.

The work still needs doing, so what we typically find is that, we can do things in parallel when it makes sense to do things at the same time especially when we have multiple processors, we have subsystems that can do more than one thing at a time, but it might not always be faster.

When we talk about blocking, or blocking I/O, or blocking calls, it just means that we're making a synchronous call. So we call into a method, our thread goes off, executes the statements in that method and comes back when it's done. So from my perspective as the caller, I'm blocked, I'm waiting. Method A calling method B, method A is blocked while method B is executing. Non-blocking thus, is what we're after in this course. **So method A calls method B, method A continues to run, cuz method B will deal with running it's code some other way, almost certainly on another thread.**

Delegate

Delgate is a key word in C#. Using delgate, we get some methods, which helps us to use asyc code.

Thread.Sleep(n) means block the current thread for at least the number of timeslices (or thread quantums) that can occur within n milliseconds.

Use the async modifier to specify that a method, [lambda expression](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/statements-expressions-operators/lambda-expressions), or [anonymous method](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/delegate-operator) is asynchronous. If you use this modifier on a method or expression, it's referred to as an *async method*.

namespace hardm\_vr703\_design\_pattern // namespace is like a folder where all codes stored

{

// Singleton design pattern (single object implementation)

public class GalilControl // class definition

{

private static GalilControl galilctrl; // class member variable

gclib gclib; // class member variable

public bool flag\_noGalilCommandSending; // class member variable

private GalilControl() // constructor is for initialization, like \_\_init\_\_ in python

{

gclib = new gclib();

flag\_noGalilCommandSending = false;

}

public static GalilControl GetInstance() // class member function

{

if (galilctrl == null)

{

galilctrl = new GalilControl();

}

return galilctrl;

}

}

}

A class or a Struct can implement one or more interfaces using colon (:).

Syntax: <Class or Struct Name> : <Interface Name>

We need to implement all the functions in every class that inherent the interface.