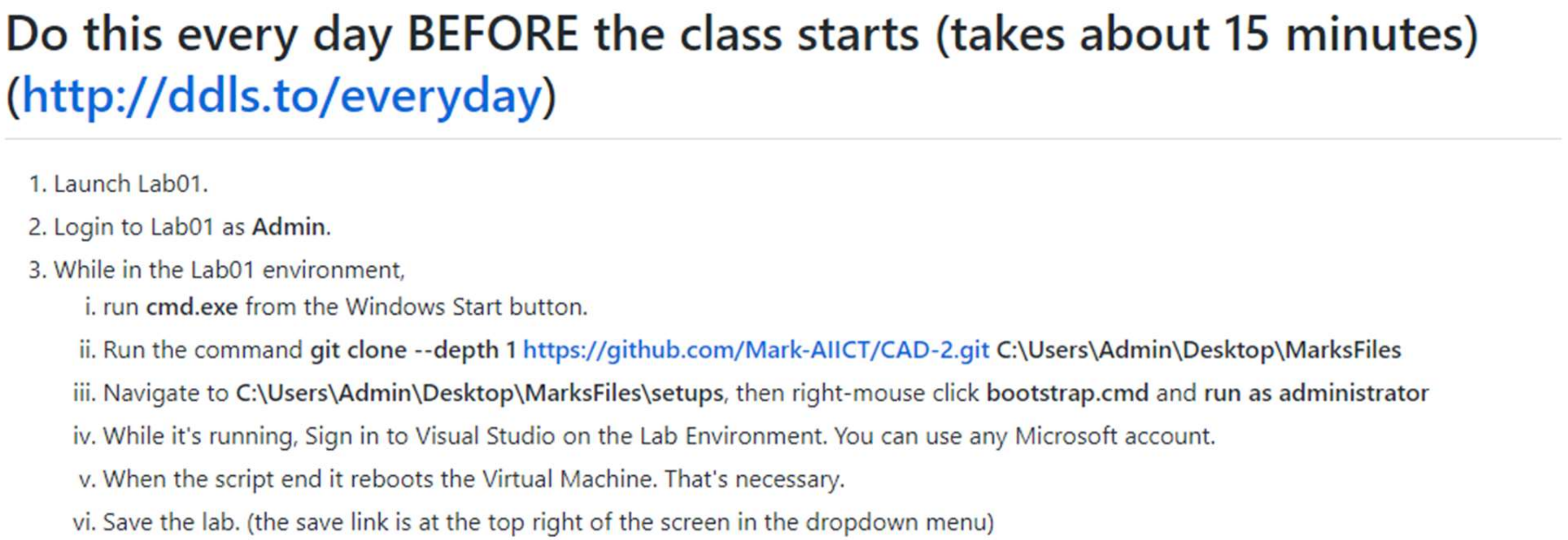
Day 6

The one link you need to recall

https://ddls.to/20483

Ready?



Do this every day BEFORE the class starts (takes about 15 minutes) (http://ddls.to/everyday)

1. Launch Lab01.
2. Login to Lab01 as **Admin**.
3. While in the Lab01 environment,
    i. run **cmd.exe** from the Windows Start button.
    ii. Run the command **git clone --depth 1** https://github.com/Mark-AIICT/CAD-2.git **C:\Users\Admin\Desktop\MarksFiles**
    iii. Navigate to **C:\Users\Admin\Desktop\MarksFiles\setups**, then right-mouse click **bootstrap.cmd** and **run as administrator**
    iv. While it's running, Sign in to Visual Studio on the Lab Environment. You can use any Microsoft account.
    v. When the script end it reboots the Virtual Machine. That's necessary.
    vi. Save the lab. (the save link is at the top right of the screen in the dropdown menu)

Course Outline

- Module 1: Review of Visual C# Syntax
- Module 2: Creating Methods, Handling Exceptions, and Monitoring Applications
- Module 3: Basic Types and Constructs of Visual C#
- Module 4: Creating Classes and Implementing Type-Safe Collections
- Module 5: Creating a Class Hierarchy by Using Inheritance
- Module 6: Reading and Writing Local Data
- Module 7: Accessing a Database

Module 3 Key Concepts

- What is an enum?

Module 3 Key Concepts

- What is a struct?

Module 3 Key Concepts

- What is a constructor?

Module 3 Key Concepts

- What is a property?

Module 3 Key Concepts

- What is an indexer?

Module 3 Key Concepts

- What is linq?

Module 3 Key Concepts

- What is a delegate?

Module 4 Key Concepts

- What is *object-oriented*?
- What is the benefit?

Module 4 Key Concepts

- What is a *class*?
- What is an *object*?

Module 4 Key Concepts

- What does this do?

```
var b = new BankAccount(){number=222};
```

Module 4 Key Concepts

- What does this do?

```
var b = new BankAccount(100){number=222};
```

Module 4 Key Concepts

- What do these mean?
  - *Public*
  - *Private*
  - *Internal*

Module 4 Key Concepts

- What is the difference between a *reference type* and a *value type*?

Module 4 Key Concepts

- What is *static*?

Module 4 Key Concepts

- What is an *interface*?

Module 4 Key Concepts

- What is a *generic*?

Module 5 Key Concepts

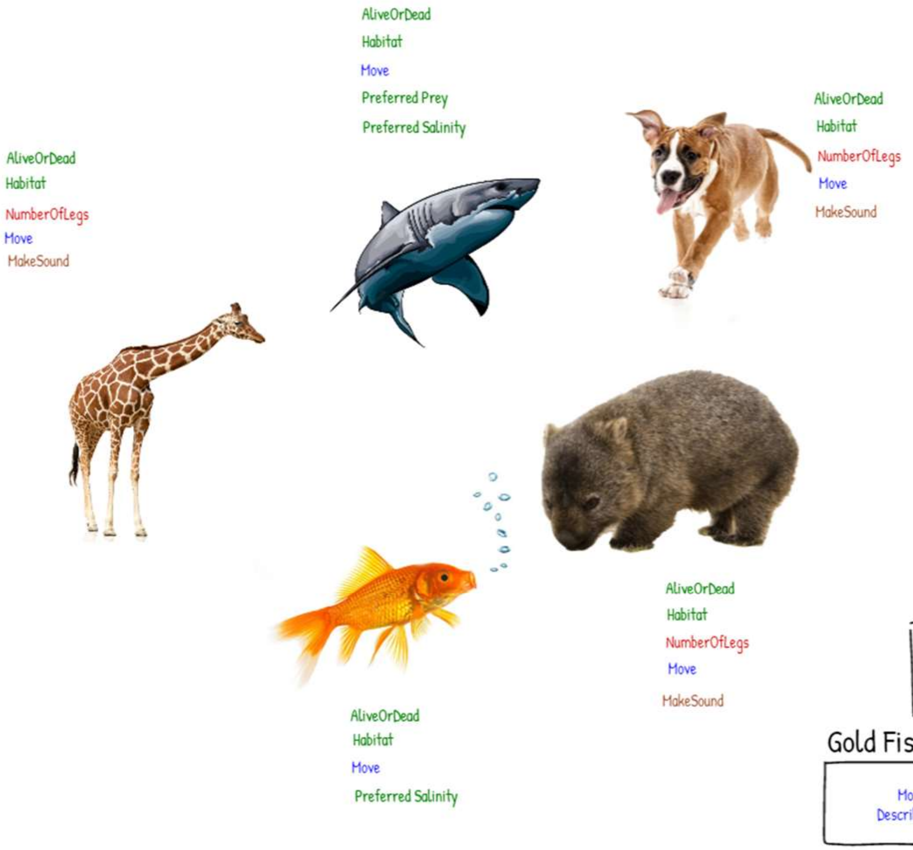
- What in OO is *inheritance*?
- What is the benefit?

Module 5 Key Concepts

- What is a *base class*?
- What is another term for a base class?

Module 5 Key Concepts

- What is a *derived class*?
- What is another term for a derived class?



AliveOrDead
Habitat
NumberOfLegs
Move
MakeSound

AliveOrDead
Habitat
Move
Preferred Prey
Preferred Salinity

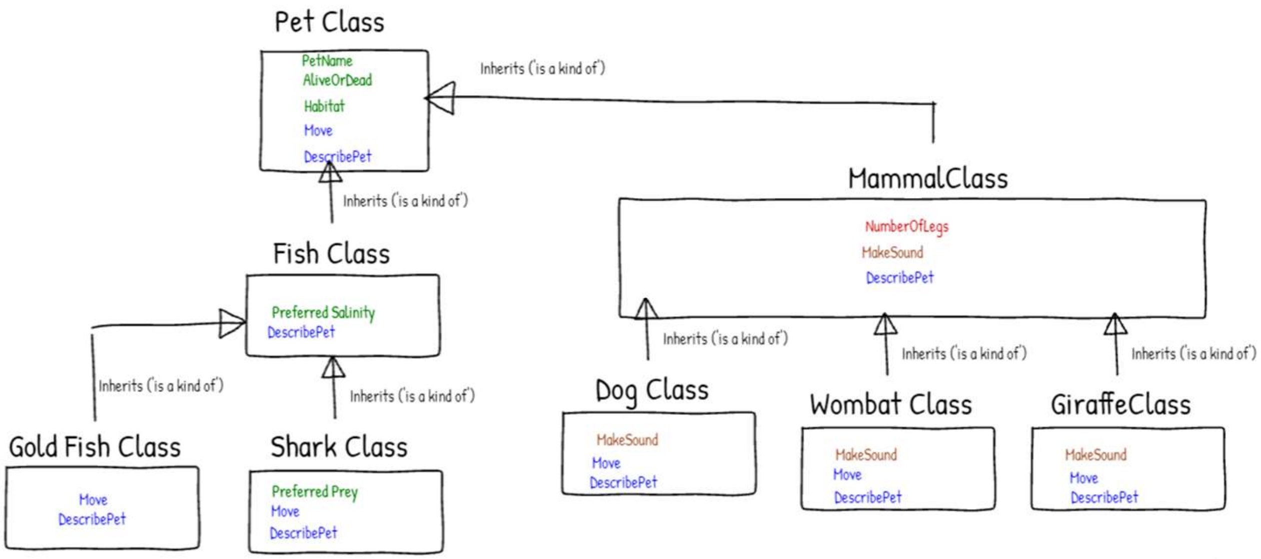
AliveOrDead
Habitat
NumberOfLegs
Move
MakeSound

AliveOrDead
Habitat
NumberOfLegs
Move
MakeSound

AliveOrDead
Habitat
Move
Preferred Salinity

These are my Pets

- * What attributes do these have?
- * What can these pets do?
- * What do these have in common?



Pet Class
PetName
AliveOrDead
Habitat
Move
DescribePet

Inherits ('is a kind of')

Fish Class
Preferred Salinity
DescribePet

MammalClass
NumberOfLegs
MakeSound
DescribePet

Inherits ('is a kind of')

Gold Fish Class
Move
DescribePet

Shark Class
Preferred Prey
Move
DescribePet

Dog Class
MakeSound
Move
DescribePet

Wombat Class
MakeSound
Move
DescribePet

GiraffeClass
MakeSound
Move
DescribePet

Module 5 Key Concepts

- What do these mean and where would I see them?
  - *abstract*
  - *sealed*
  - *virtual*
  - *protected*
  - *override*

Module 5 Key Concepts

- What is going on here?

```
public Coffee(string name, bool isFairTrade, int temp)
   : base(name, isFairTrade, servingTemp)
```

Module 5 Key Concepts

- What is going on here?

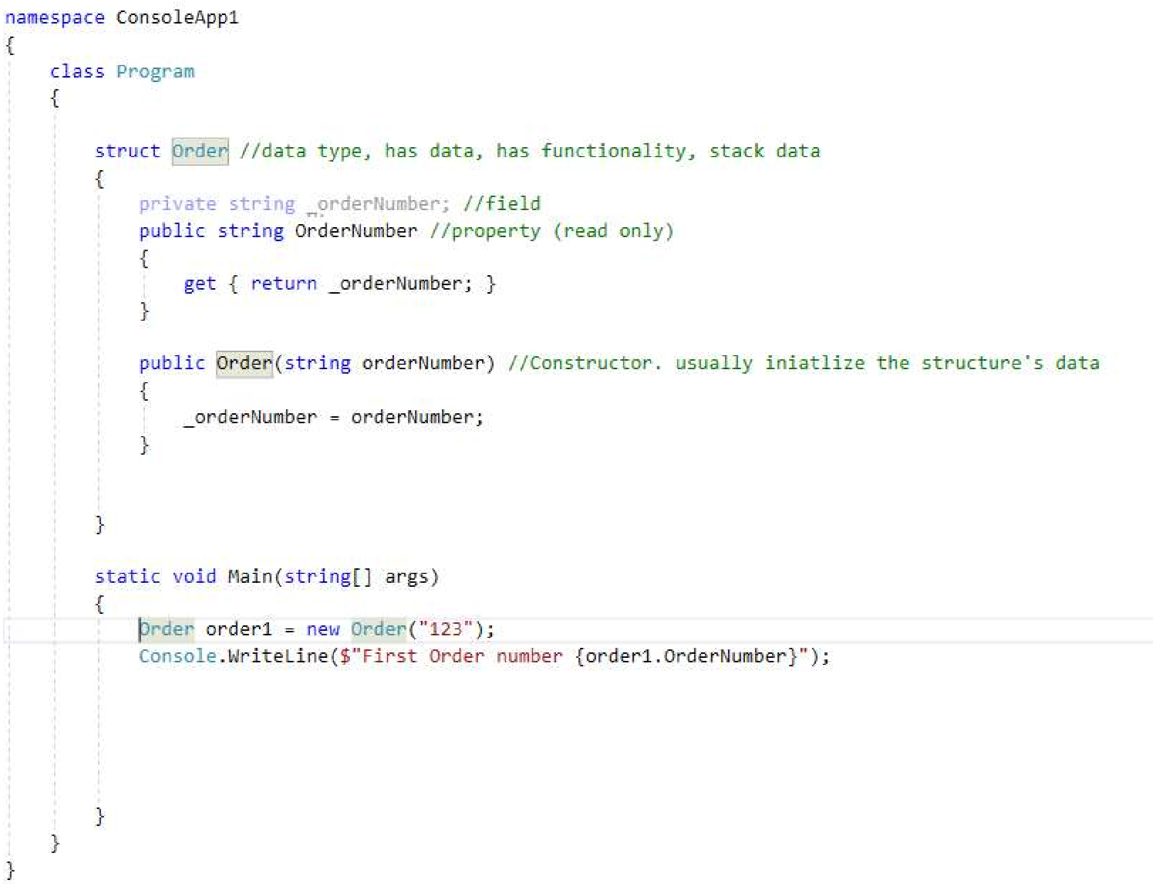
```
base.GetServingTemperature();
```

Module 5 Key Concepts

- What is an *extension method*?
- What is the benefit?

Course Outline

- Module 1: Review of Visual C# Syntax
- Module 2: Creating Methods, Handling Exceptions, and Monitoring Applications
- Module 3: Developing the Code for a Graphical Application
- Module 4: Creating Classes and Implementing Type-Safe Collections
- Module 5: Creating a Class Hierarchy by Using Inheritance
- Module 6: Reading and Writing Local Data
- Module 7: Accessing a Database



```
namespace ConsoleApp1
{
    class Program
    {

        struct Order //data type, has data, has functionality, stack data
        {
            private string _orderNumber; //field
            public string OrderNumber //property (read only)
            {
                get { return _orderNumber; }
            }

            public Order(string orderNumber) //Constructor. usually iniatlize the structure's data
            {
                _orderNumber = orderNumber;
            }


        }

        static void Main(string[] args)
        {
            Order order1 = new Order("123");
            Console.WriteLine($"First Order number {order1.OrderNumber}");




        }
    }
}
```

Infers the data type of 'b' from the right-hand-side of the assignment operator.|

Module 4 Key Concepts

variable (on the stack)

object is based on this class

- What does this do?

Sets the property 'number' to 222

```
var b = new BankAccount(){number=222};
```

assigns the address of the new object to 'b'

Creates an object

Calls the parameter-less constructor

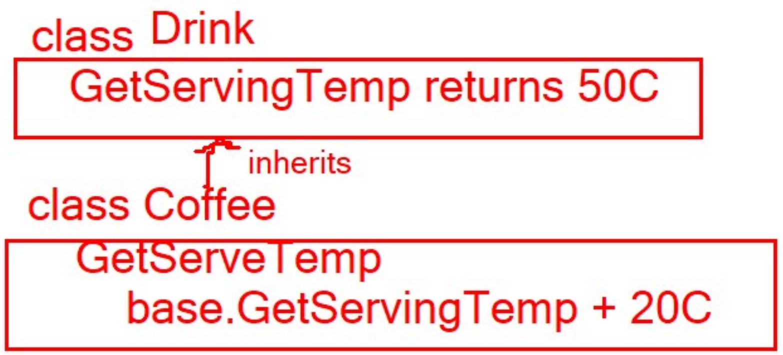
Module 5 Key Concepts

- What is going on here?

base.GetServingTemperature();

This is calling GetServingTemperature method in the base class.

We'd often see this in the method that overrides a base class method



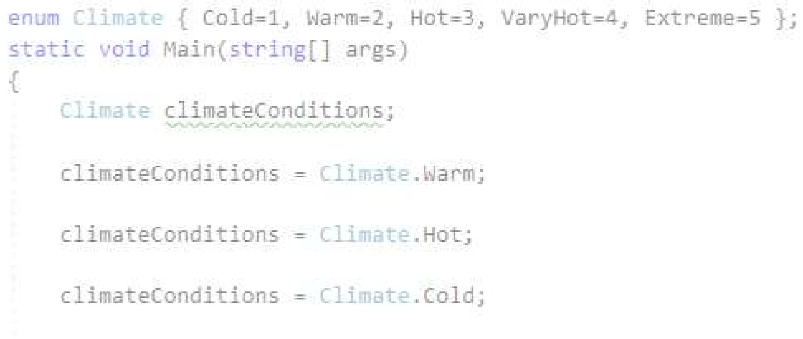
class Drink

GetServingTemp returns 50C

inherits

class Coffee

GetServeTemp
    base.GetServingTemp + 20C



```
enum Climate { Cold=1, Warm=2, Hot=3, VaryHot=4, Extreme=5 };
static void Main(string[] args)
{
    Climate climateConditions;

    climateConditions = Climate.Warm;

    climateConditions = Climate.Hot;

    climateConditions = Climate.Cold;
```