**Topic 2: Inheritance (OOP)**

Top of Form

Bottom of Form

**Content**

**Reading**

[Reading](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286756_1)

Object-Oriented Programming C# -- Inheritance

* + Microsoft:docs.microsoft.com
    - [Inheritance C# and .NET](https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/inheritance)
  + GeeksForGeeks: geeksforgeeks.com
    - [C# Inhertiance](https://www.geeksforgeeks.org/c-sharp-inheritance/)
  + TutorialsPoint: tutorialspoint.com
    - [C# - Inheritance](https://www.tutorialspoint.com/csharp/csharp_inheritance.htm)
  + W3Schools: w3schools.com
    - [C# Inheritance](https://www.w3schools.com/cs/cs_inheritance.asp)

**Connecting Classes: Composition vs Inheritance**

[Connecting Classes: Composition vs Inheritance](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286756_1)

**Composition**

Composition is using other classes in your class, the relationship is "has-a". This applies to properties. So if a class should represent a property objecgt, you use composition. This was covered in the previous topic (Using classes in classes). Now you know the technical term to describe it, composition.

* + Student class
    - A student object has a first name
    - A student object has a last name
    - A student object has an id
    - A student object has an address
  + Building class
    - A building object has a size
    - A building object has an address

**Inheritance**

Inheritance is another way to connect classes. This linking is more an "is-a" relationship and the when a class is a derived class, it inherits all the properties and methods from the base class. So the derived class "is-a" base class. The age old examples

* + A car object is a vechicle
  + A dog object is an animal
  + A bicycle object is a cycle
  + and the list goes on....

You can make your own base and derived classes.

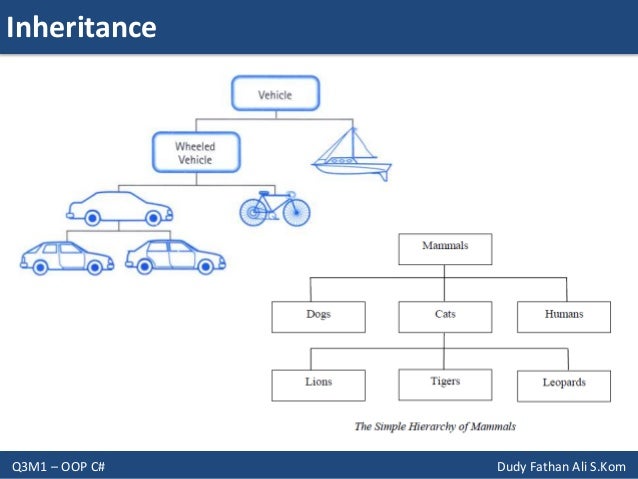
You may be asking, when do I use composition and when do I use inheritance?

How closely the classes are connected gives you the clues. Also, it may depend on your program and what you are trying to accomplish in your program. You can ask how they are related. Ask if one class is another class OR if one class has another class.

In the previous topic, you applied composition. Now you can learn about the Object-Oriented Principle Inhertiance.

**Inheritance**

[Inheritance](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286756_1)



**Example of Inheritance**

[Example of Inheritance](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286756_1)

https://youtu.be/-0bJZR7CYI0

Base (Parent, Super) Class Building:

public class Building

{

private string \_address;

private double \_size; // in square feet

// Constructors

public Building()

{

Address = "unknown";

Size = 0.0;

}

public Building(string address, double size)

{

Address = address;

Size = size;

}

// Properties

public string Address { get => \_address; set => \_address = value; }

public double Size { get => \_size; set => \_size = value; }

// Methods

public override string ToString()

{

return "Buidling at " + Address + " is " + Size.ToString() + " ft^2.";

}

}

Derived (Child, Sub) Class House:

public class House : Building

{

private int \_numberBedrooms;

private int \_numberBathrooms;

// Constructors

public House() : base()

{

NumberBedrooms = 0;

NumberBathrooms = 0;

}

public House(string address, int size, int numberBedrooms, int numbrBathrooms) : base(address, size)

{

NumberBedrooms = numberBedrooms;

NumberBathrooms = numbrBathrooms;

}

// New Properties, House still can access Building Properties Size and Address

public int NumberBedrooms { get => \_numberBedrooms; set => \_numberBedrooms = value; }

public int NumberBathrooms { get => \_numberBathrooms; set => \_numberBathrooms = value; }

public override string ToString()

{

return base.ToString() + " It also has " + \_numberBedrooms + " bedrooms " + \_numberBathrooms + " baths.";

}

}

**How to override methods in Derived Class**

[How to override methods in Derived Class](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286756_1)

Previous examples show that ToString() is overriden. THis means the base class has method ToString(). This base class is the [Object Class(System)](https://docs.microsoft.com/en-us/dotnet/api/system.object?view=net-5.0), it is "ultimate base class of all .NET classes; it is the root of the type hierarchy." This means that any class you write inherits from this base class. Thus, when write your ToString() method in your own base classes, you still need the override keyword to signify the base Object Class ToString() is being overridden. If you look at the [ToString()](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/override#:~:text=An%20override%20method%20provides%20a,as%20the%20overridden%20base%20method.), which returns a string representing the current object, in the Object Class, you see the following:

public virtual string? ToString ();

 Notice the virtual keyword. When you write your own base class will need to have the keyword virutal to indicate it can be overridden. In the derived class, there are some options. You can extend the method and use override in the derived class or you can hide the base the class method using the new keyword.

* + Base class: virtual method-if you plan to override in derived class
  + Derived class
    - Extend the base method: override
    - Hide the base method: new

This might be clearer with a few examples

**Base class:**

**https://youtu.be/tbjY8KSOuhU**

**Files:** [**WritingUtensil.cs**](https://dmacc.blackboard.com/bbcswebdav/pid-7286772-dt-content-rid-101466181_1/xid-101466181_1)[**Program.cs**](https://dmacc.blackboard.com/bbcswebdav/pid-7286772-dt-content-rid-101466182_1/xid-101466182_1)

**https://youtu.be/1Pt8TD4tS8w**

**Files:** [**Marker.cs**](https://dmacc.blackboard.com/bbcswebdav/pid-7286772-dt-content-rid-101466183_1/xid-101466183_1)[**Program.cs**](https://dmacc.blackboard.com/bbcswebdav/pid-7286772-dt-content-rid-101466184_1/xid-101466184_1)