**Readings:**

[**https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/inheritance**](https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/inheritance)

[**https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/inheritance**](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/inheritance)

[**https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/base**](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/base)

**https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/override**

[**https://docs.unrealengine.com/latest/INT/GettingStarted/FromUnity/index.html**](https://docs.unrealengine.com/latest/INT/GettingStarted/FromUnity/index.html)

[**https://docs.unrealengine.com/latest/INT/Gameplay/Framework/\**](https://docs.unrealengine.com/latest/INT/Gameplay/Framework/\)

[**https://docs.unrealengine.com/latest/INT/Gameplay/Framework/QuickReference/index.html**](https://docs.unrealengine.com/latest/INT/Gameplay/Framework/QuickReference/index.html)

[**https://docs.unrealengine.com/latest/INT/Gameplay/index.html**](https://docs.unrealengine.com/latest/INT/Gameplay/index.html)

**Part 1 – System Diagrams**

* Diagram all of the required components for the following game types.
  + You should have an image document for each game type.
    - Use a UML diagram tool to help build this.
    - A good tool is: https://www.draw.io/
  + Simplify your diagram where you can by collecting things into categories.
    - Example: You don’t need to list all weapon types, a “weapon” class to abstract all of them is fine
* DeathMatch First Person Shooter
* Capture The Flag

**Part 2 – Unreal Engine Structure**

Several pages from the Unreal Engine Documentation has been given to you. Review them and other documentation. Provide a diagram that shows the basic relationships of the core gameplay framework classes in Unreal Engine.

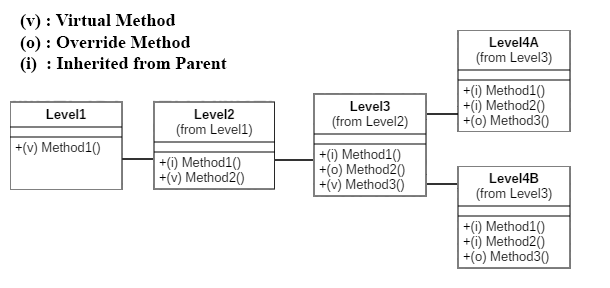
*Hint: If you do the readings, you might just find the answer you need…*

**PART 3 – Inheritance and Overrides Example:**

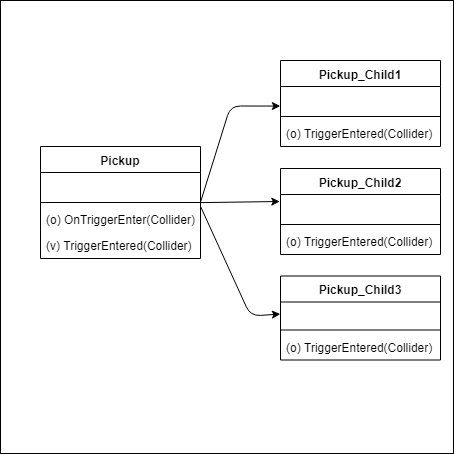
* This Assignment is C# Console Application
  + Name the project <LastName>\_Lab1\_InheritanceExample
* Create a set of 5 classes that inherit using the structure
  + For this assignment, write all 5 classes in one CS File called “Levels.cs”

Level1  
 |->Level2  
 |-> Level3   
 |-> Level4A  
 |-> Level4B

* Write the following methods in the classes
  + Have each method that prints a string that indicates the method was called.
  + Have the override methods indicate that it is an override
  + In class ‘Level1’
    - Add a new virtual method Method1
  + In class ‘Level2’
    - Add a new virtual method Method2
  + In class ‘Level3’
    - Add a new virtual method Method3
    - Add an override of Method2
  + In class ‘Level4A’
    - Add an override of Method3
  + In class ‘Level4B’
    - Add an override of Method3



**PART 4 – Unity Example**

****

* Create a simple Unity project
* Create the scripts using the Diagram above
  + Create a Script “Pickup”
    - Add “TriggerEntered”
      * Public
      * Virtual
    - OnTriggerEnter
      * will call “TriggerEntered”
      * Destroy the Game Object
  + Create Three Child Classes of “Pickup”
    - This will override “TriggerEntered”
    - Each Child will do something different.
* Create 3 Prefabs using the child classes
* Create a simple scene to test.