**CSE210 Week 05 - Inheritance**

Inheritance is the ability for one class to obtain the attributes and methods of another class directly without having to type them. This is kind of like a child inheriting a trait from a parent. For example, it there are two classes a Person and a Student. A Person has attributes or methods that all people share such as GetName():. A Student is a Person and will also have all the properties of a Person Class, however, a Student may have extra items such as a student ID, major, courses, etc. The Student Class could inherit all Person Class functionality and add to it. So a Person Class is a Parent Class and a Student Class is a Child Class. They are called base and derived class or Super and Sub classes.

In C# defining names of class always with a colon followed by the name of the parent class. Some of the benefits of having a parent class is you are able to call GetName() method on an instance of Student even though it is not designed in that class. The program can access private data but because it is private, you can’t access say \_name directly in the methods defined in the Student class. It is necessary to write code GetStudentInfo() that will return both the name and ID number for the student.

When writing code, it is important to separate attributes out that are common among all of the classes. If all the classes require the program to retrieve the same information, it is important to create a base or Parent class. The other classes can derive that information from the base or Parent class which will eliminate the need to write as much code for each of the child or derived classes. These Sub classes will inherit the attributes and methods of its Super/base or Parent class. Here is a graph of a Parent Class with the Sub or Child classes.

Assignment

\_studentName: string

\_topic: string

GetSummary(): string

Writing Assignment

\_title: string

GetWritingInfo(): string

Math Assignment

\_textbookSection: string

\_problem: string

GetHomeworkList(): string

This demonstration is showing that the Parent class is Assignment. This holds the studentName, Topic, and the call or GetSummary strings. Both the children classes or derived classes will draw from the parent class because both will need the students name and the topic of the assignment. The both will need the summary called. The Math class is going to hold information such as the textbook the math problem will come from. It will also hold the math problem and the call will be GetHomeworkList. The Writing class needs to know the title of the article and the call is GetWritinginfor(). Both of these need their own classes because they need different information, however, they both need the same information from the parent or Assignment class.

A Parent class could have code written like this,

using System;

namespace assignment\_class

{

public class Assignment

{

private string \_studentName;

private string \_topic;

public Assignment(string studentName, string topic)

{

\_studentName = studentName;

\_topic = topic;

}

public string GetStudentName()

{

return \_studentName;

}

public string GetTopic()

{

return \_topic;

}

public string GetSummary()

{

return $"{\_studentName} - {\_topic}";

}