MODULE 1: INTRODUCTION TO PROGRAMMING

Managing Inheritance

TECH ELEVATOR
A Stride Company

Yesterday

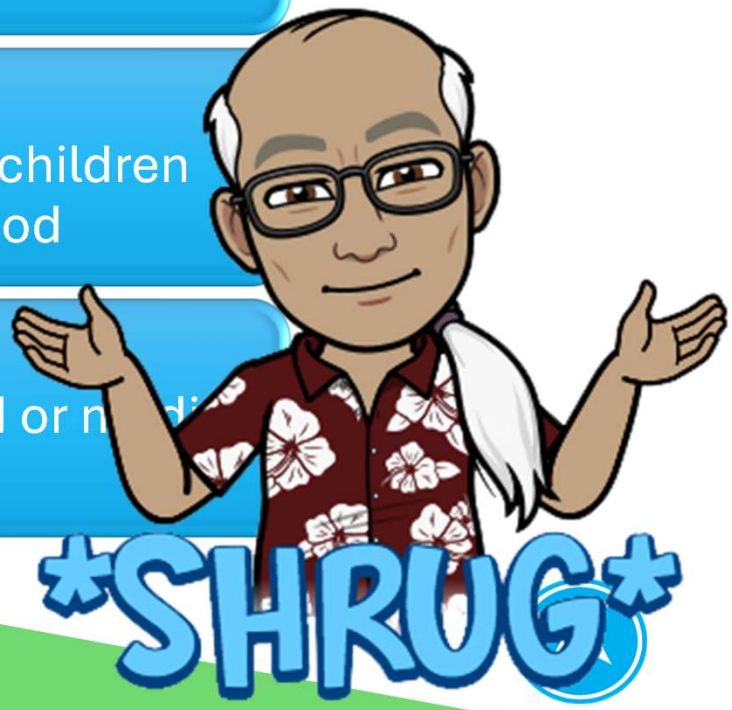
- What is polymorphism?
- What is one way to have polymorphism?
- What's the other?
- What is an interface?

Access Modifiers

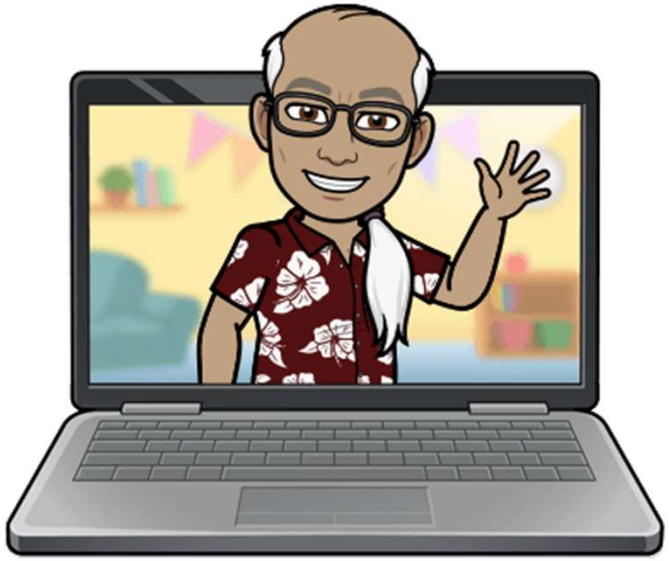
Public
Any code with access to the class can add or modify the property or access the method

Protected
The object where the item is declared and all their children can add or modify the property or access the method

Private
Only the object where the item is declared can add or m di the property or access the method



LET’S CODE!



Final

- Final classes are used to restrict the inheritance feature of object oriented programming. Once a class is defined as final class, this class cannot be inherited.
- Final methods prevent overriding a method of a class (default functionality in C#)

Abstract

- Abstract classes can not have objects created from them, but they can provide logic and structure to their subclasses.
- Abstract methods are methods with no logic that must be implemented by concrete subclasses
- If a class has an abstract method, it must be an abstract class
- If a class does not override an abstract method from its parent, it must also be an abstract class

Abstract vs Interface

- Abstract classes can still have implemented methods. Interfaces can have no implemented methods.
- A class can only inherit from one other class, but can implement as many interfaces as you want it to.
- Inheriting from a super class is kind of like making a more specialized version of that class. Implementing just means that you can be used in the context the interface is for. There is a difference in being a kind of Book versus being Sellable or Readable.

WHAT QUESTIONS DO YOU HAVE?

Code Reviews

- Friday 4/4
- 5 minutes
- Sign up for slots
- Make sure no Pathway conflicts

Reading for tonight:

Unit Testing