(40 pts) Approx. 3 days

The next part of our unit is all about using interfaces to build in new connections between your classes. Interfaces let us setup a list of methods that must be overridden by any class that wants to use that interface. This helps classes of different types talk the same language when they try to interact with one another! The use of interfaces requires our next (and possibly last) java keyword for the year: *implements*

In order to implement an interface, we simply add to the class signature the word "implements" and then the name of any interfaces that the class implements. For example:

public class Panda extends ZooAnimal implements herbivore, cuddly, photogenic

In the above example, the "Panda" class **extends** the ZooAnimal class, inheriting relevant variables and methods. It then also **implements** the herbivore, cuddly, and photogenic interfaces. In order to do so, our Panda class will need to override every abstract method defined in each one of those interfaces. The advantage to us as the programmer is that then we'll have more guarantees about what the Panda can do and how it can interact with other herbivores, etc.

1. Watch the three videos on interfaces and take good notes
2. Now, complete the following 6 challenges, paying close attention to the creation and implementation of interfaces:
    a. JAVA TASK 65: Make a program that contains a "Student" class with at least 4 instance variables, all setters & getters, as well as constructor and a toString() method. (This class will implement the rest of the tasks)
    b. JAVA TASK 66: Many students are also great musicians! Make an interface called "Musician" and give it 3 abstract methods. Then, go back to your Student class and properly implement the Musician interface.
    c. JAVA TASK 67: Many students are also excellent athletes! Make an interface called "Athlete" and give it 3 abstract methods. Then, go back to your Student class and properly implement the Athlete interface.
    d. JAVA TASK 68: Many students are hard-working mathematicians! Make an interface called "Mathematician" and give it 4 abstract methods. Then, go back to your Student class and properly implement the Mathematician interface.
    e. JAVA TASK 69: Many students are also very talented artists! Make an interface called "Artist" and give it 5 abstract methods. Then, go back to your Student class and properly implement the Artist interface.
    f. JAVA TASK 70: Many students also work jobs outside of school! Make an interface called "Employee" and give it ***6 abstract methods***. Then, go back to your Student class and properly implement the Employee interface.

Part 2: Tasks	*10-7 points*	*6-4 points*	*3-0 points*
Interface Notes	+ Watch the three presentations on Interfaces + Take a full page of notes on the ideas, including details on the keyword "implements"	- Less than a full page of interface notes - No brainstorming present	- Very brief or no notes in your notebook
	20-10 points	*9-5 points*	*4-0 points*
Java Tasks 65-70	+ You completed all 6 Java Tasks from this section	- You did not complete all 6 tasks	- You did not complete any tasks
	10-7 points	*6-4 points*	*3-0 points*
Take Unit 8 Quiz	+ You took the Unit 8 Quiz on the website by the Quiz Due Date + Grade is based on number correct	N/A	(0 pts) You did not take the Unit 8 Quiz
Checkoff from Benshoof	+ Mr. Benshoof got to see your Java programs run successfully	N/A	N/A