Announcements

- Read for next time Chap. 13.5-13.7, 7.4
- Reading Quiz for next time

Abstract Class

- A class that cannot be instantiated
- Defines specifications for a class, but doesn’t have all the implementations
- Forces users to redefine methods
- Subclasses extend it
- May have abstract methods
  - There is no code
  - Subclasses that are not abstract must have a method of the same name with code

Inheritance hierarchy

- Mover and Bouncer are abstract classes
- Bouncer extends Mover
- BouncingBall and BouncingSmiley extend Bouncer
**Polymorphism**

- Ability to refer to objects of multiple types with varying behavior
- `ArrayList<Mover>` contains `BouncingBalls` and `BouncingSmileys` and `RaceCars`
- They are all subclasses of `Mover`
- Can’t create a new `Mover`. Why?

**Classwork Today**

- Finish `Bouncing Smileys`
- Write the `RaceCar` Class. Extends `Mover`.
  - `RaceCars` only move to the right
  - `RaceCars` are rectangular in shape
- Modify the `Canvas` class to create `RaceCars`