**Statics:**

* Static fields and static methods do not belong to a single instance of a class
* Common explanation is that statics belong to the entire class, not just one instance
* Main is static because the class needs to access that
  + Main is also global, and apparently globals are bad
* Static fields and methods are properties of a class
  + Static methods are good for utility i.e. it will be used from multiple classes, multiple instances
* Non-static fields and methods are properties of an instance
  + Non-static or instance is the same thing
  + Every time new is used, it’s a new instance of a method i.e. new int[4];
* Example, every math method is static because it makes much more sense to have a static method available to every class than to force a new instance of math class every single time there needs to be a math call. Math.sqrt(25.0); instead of Math myMath = new Math(); myMath.sqrt(25.0);
* You can access statics anywhere, and since global is bad, don’t have static unless its for utilitarian stuff
* Primitive static fields are initialized to 0 if no initialization is performed
* All instances of class shares the same static fields
  + I feel like that would explain some bad behavior in some of my code
* Ask yourself, does it make sense to call this method without constructing an Object?
* Static can’t access instance fields
  + Reason why I add static to everything but apparently that’s bad
* Reasons why statics are bad:
  + Static is global, and the memory exists the entire runtime of the program
  + No thread safety
  + Not good security

**Strings:**

**Inheritance:**

* Every single Java class inherits from Object, which provides toString() and equals()