**Using Objects**

Class vs. Object

* Class: template for building objects
  + Defines the methods and variables an object has
  + Defines how an object is created
  + Ex: Student class
    - Defines *how* students:
      * Learn
      * Sleep
      * Weep
      * Study
    - Defines *what* a student learns:
      * Name
      * Student ID
      * Age
      * Gender
      * GPA
* Object: the “physical” entity/thing that preforms actions (methods) and stores data (variable)
  + \*An object is an instance of class
  + Objects have 3 things:
    - Behavior: methods
    - State: instance variables
    - Identity
  + Ex: Student Object
    - Can do:
      * Sleep
      * Learn
      * Weep
      * Study
    - Know:
      * Name
      * Age
      * Student ID
      * GPA
  + \*Encapsulation\* -- the idea that an object “owns” it’s own variables and can do things to those variables

Primitive vs Objects

* Primitives contain a value
* Objects can do things and store values

Variables vs. Reference Variables

* Variables just store a direct value
  + int x = 4;
* Reference values store objects which store different data
  + The object can also do stuff
  + The reference variable is like a remote control
  + The variable is just the name of the remote

Primitive Declaration

* type name = value;

Object Declaration

* object instantiation

ClassName name = new ClassName(parameters 🡪 initialize the unknown);

Math Class: math functions

* pow()
* round()
* sqrt()

String Class: methods to manipulate words

Scanner Class: console input

API (Application Programming Interface)