Girl Develop It, Class 1

- Object Oriented Programming (OOP)
  - System - complex set of objects interacting to create an outcome. Examples: rainwater filtration system or a car engine
  - Object
    - A piece of a system that has attributes (state) and abilities (behavior)
      - Examples: Penguin, Camera, Lens
    - Objects can interact with each other by passing messages.
    - Objects can be composed of other objects ("has a" or composition)
    - Objects can inherit traits from a base object ("is a" or inheritance).
    - Information hiding is keeping your attributes private, but advertising you abilities.
  - Class diagrams are used to show how objects relate to each other. The boxes have 3 parts: top section is name, middle section has attributes, and the bottom section has abilities.
  - There is no right or wrong way to break down a system into a set of objects. It may not turn out to be the best way to brake it down, but you can always change it later (aka refactor).
- Java
  - Java is an object oriented language.
  - Compliers
    - Compilers turn code (human readable) into machine code (computer readable). The input to the compiler is the code you write and the output is something that the computer can run.
    - Some languages are highly dependent on the machine (like C and C++) and some are not (scripting languages like python, ruby, perl).
    - Machine dependent languages are faster because the program it knows how to talk directly to the computer. The down side is that they aren't as portable.
    - Scripting languages are much slower because an "interpreter" is talking to the computer for you. The benefit is that you can run them from anywhere and get consistent results.
    - Java is somewhere in the middle. It complies to byte code and then is run on the virtual machine, which acts like an interpreter.
  - API
    - The "interface" part of refers what you can do with the library. It's publicized behaviors.
    - The Penguin's API would be eat(item), waddle(), and swim()
    - Twitter's API may contain validate(username, password) and getToken()
  - IntelliJ
    - Integrated Development Environment (IDE) - Putting all the tools to make programmers happy in one place.
  - Variables - A place to store a value. You give it a name so that it makes sense when you come back to your code a week later. There are a few different types, we will focus mainly on Strings and ints.
  - Operators - You can perform mathematical and comparison operations on variables and hard-coded values (aka. literals).
    - Example
      ```
      int weightOfPenguin = 25;
      int newWeight = weightOfPenguin + 10; // 35
      int anotherWeight = weightOfPenguin + newWeight; // 35 + 25 = 60
      ```
    - Example 2
      ```
      boolean value;
      value = anotherWeight >= weightOfPenguin; // true because 60 > 25
      ```
  - Control flow
    - "if" statement checks one condition. If it is true, then it does one thing, if it's false, it does another.
    - "while" loop does something repeatedly until the condition equates to false. While the condition is true, it continues to execute.
    - "for" loop does something for a certain amount of times. The for loop includes an initializer (the first section) which tells the loop what the starting value is. The second section is where the condition is checked each time through the loop. The final section is what should happen to the counter variable each time through the loop.
  - Method (function)
    - A method is a collection of related statements (think recipe or paragraph).
    - Input can be sent in to the function (aka. parameter), but it is not required.
    - A method may return a value using the "return" statement, but this is also not required.
    - A method operates on the input passed into to produce a result. That result is typically returned to the caller (the location in code where the the function was used)
  - Class
    - This is where your objects you created earlier in the design phase live in code.
    - An object can have attributes (variables) and abilities (methods).
    - Methods typically operate on the variables inside a class to change the state.
    - A class is like a blueprint - it tells you how to construct a new object of this type. (Ex. Penguin)
    - An instance is when you use the object blueprint to create a particular object (Ex. babyTux)