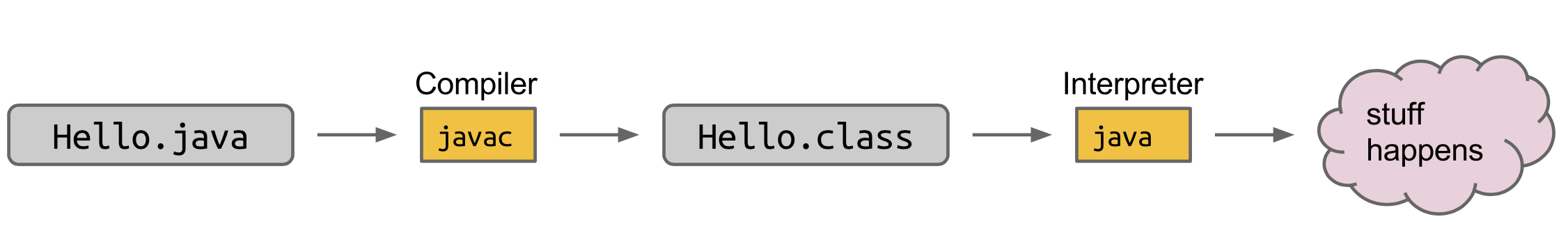
CS 61B Data Structures

Practice

Self-efficient

* Java is compiled language rather than interpreted language (python)
* Java is statically typed
* Variable, parameter and function must have a defined type, and it’s permanent.
* The types are checked before run.
* Good:
  + Faster.
  + Check error before run.
  + Easier to reason about.
* Bad:
  + Less general, more verbose.
* Java is object oriented
  + Every java file must define a class, and that class must have the same name with the file.
  + To run java program, must create “main”.
  + not all class must have main method, can create at test drive and call the method.
* Compile java program
  + javac Hello.java
  + java Hello



* elements of java program
  + instance variable

public int size;

* + constructor

public Dog(int s) {

size = s;

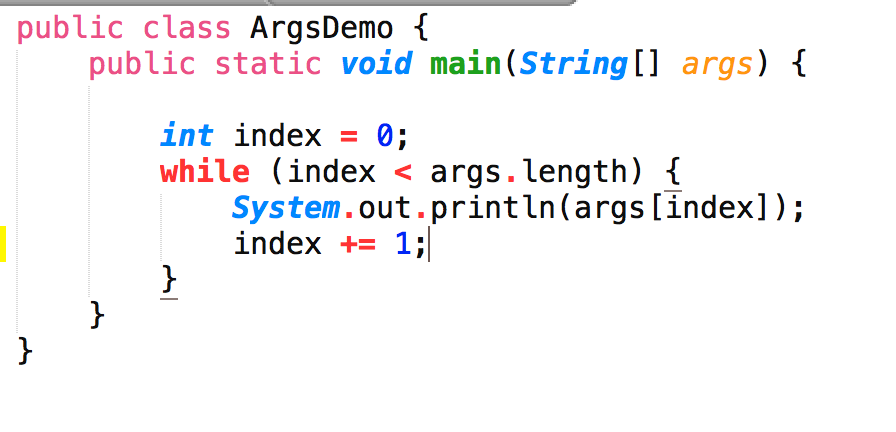
}

* + declaration, instantiation, and assignment
    - declaration: Dog maya
    - instantiation: new Dog(15)
    - assignment: Dog maya = new Dog(15);
    - all in one: Dog maya = new Dog(30)
    - Create an array of instances
      * Dog[] someDogs = new Dog[2];
      * someDogs[0] = new Dog(10);
  + invocation of method
    - maya.makeNoise();
  + Define function

public static int larger(int x, int y)

Return type and input type

* Non-static method
  + if the method is going to invoked by an instance of the the class, it should be non-static.
  + static methods (members: variables, parameters…) are invoked using the class name, non-static methods are invoked using the instance name.
* Args
  + command line input



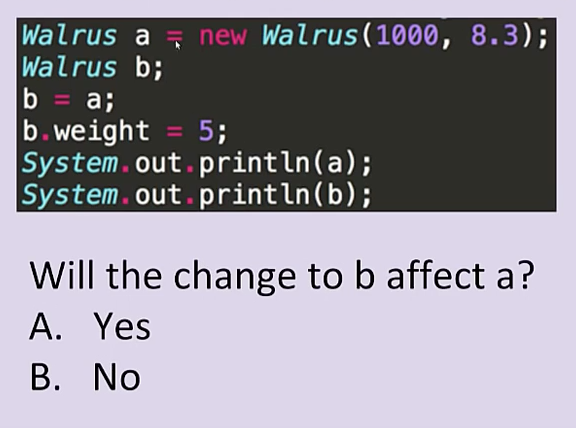
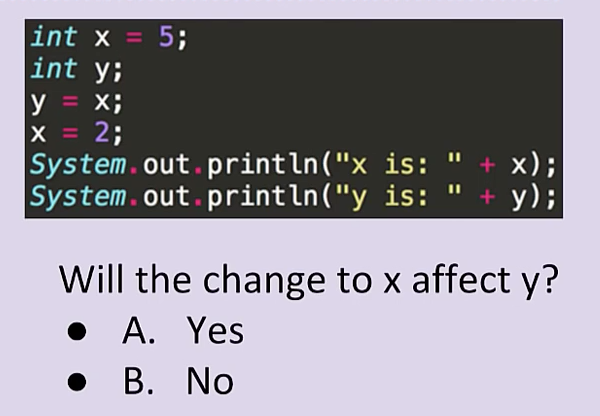
* + $ java ArgsDemo abcd bs bcd



* + Integer.parseInt()



* public: other class can access this method
* private: user does not know, safe for you to change
* variable

Yes No

* + primitive types: byte, int, short, long, double, float, char, Boolean

b = a, copies all the bits from a into b.

box, either 32 bits (int) or 64 bits (double)

un- instantiation, error

* + reference types: everything else

64 bits addresses, null or arrow