A Picture is worth $2^{10}$ words

PFTD and most of the week
- Be able to explain what a class is, what the parts of a class are, how classes are used in Java Programs
  - Methods, Constructors, Instance Variables, Keyword: this
- Be able to explain the differences between an array and an ArrayList
  - Includes creating, access, update, auto-boxing/unboxing
- Be able to create a Java class starting with nothing, be able to modify a Java class
  - Using Eclipse, but also using IDE-independent concepts
- Be able to test methods and classes
  - Using supplied test files/cases, toward your own testing

Simulating planets or molecules
- Each object has IRL state
  - Bonds between atoms, types of elements, mass, size, ...
  - Atmosphere, mass, distance from star, ...
- Each object has state in simulation of motion
  - Coordinates in space ($x,y$) or ($x,y,z$) or ($x,y,z,t$) or ...
  - Motion, mass, size
- Objects also have behavior: bond, move, bounce, ...

Class encapsulates state and behavior
- Class captures commonalities, each object has different characteristics
  - Dogs have fur, size, speed, temperament, ...
- I will not typically use examples like this, but they are often useful and in common practice
3.5 Class in Java (other OO languages)

- Define class named Foo in Foo.java
- Create object by calling new Foo(..)
- Access object by calling methods: obj.doSomething()

3.6 Class MolecularBall

- Name of file?
- State?
- Constructor?
- Behavior?

- Examine use of class
  - Simulation
  - Leads to NBody

3.7 Ball motion simulation

https://git.cs.duke.edu/201fall16/bouncing-balls/tree/master
See video at: http://www.youtube.com/watch?v=V57vHOMx0BE

3.8 Understanding the Simulation

http://bit.ly/201fall16-sep6-1

- As you read the code, notice conventions used in naming local variables, instance variables, methods, classes
- How is the class StdDraw used in simulation?
  - You’ll use this class in NBody simulation as well
**Class**

- **Object-oriented way to combine state and behavior**
  - State is: instance variables, behavior is methods
  - Examples: Point, String, MolecularBall, ArrayList

- **Objects are instances of a class. Class is blueprint**
  - Create objects by calling new – invokes *constructor*
  - A constructor is like a method, but initializes object
  - Typically you put code in the constructor

- **Objects communicate via methods (parameters)**
  - Object can pass itself: *this* is a keyword for that

**Code Review: Simulation**

- **Walk through both BallWorld.java and MolecularBall.java**
  - Attentive to both Java and to Eclipse/IDE for creating and updating Java

- **Where is program launch point?**
  - *public static void main* blah blah blah
  - What does public mean? What does static mean? What does launch mean?

- **Examine primitive types, Object types, control flow, arrays, method calls**
- **Be attentive to use of this as reference to self**

https://git.cs.duke.edu/201fall16/bouncing-balls/tree/master

**Eclipse Particulars**

- **Creating new Java project, new Java classes**
  - Creating class in src folder
  - How to run a program within Eclipse, apropos of APT's

- **How to use starter code from ... several sources?**
  - Copy/paste can be your friend, but doesn't scale
  - In the past we've used Eclipse plugin: Ambient
  - Transitioning to Git, very widely used/industry "standard"

- **Refactoring code, when find-replace doesn't work**
  - Let's change MolecularBall to simply Molecule

**Arrays and ArrayLists**

- **String[], int[], double[], how to read?**
  - Type to the left of [], so what is String[]?

- **An array is an object, so created by calling new**
  - String[] s = new String[100];
  - How to access # elements? .length, but not a method!
  - Cannot grow, but minimal memory overhead and stores primitives or objects (contrast ArrayList)
  - Some methods that work on arrays,
    - see Arrays.java in java.util
    - How to read an API, what is Arrays.sort(.) Java 8
ArrayList in java.util

- Collection of objects, cannot store primitives
  - Primitives are autoboxed and unboxed, so can .add
  - Access with .get, change with .set(__, __)
  - Query with .contains, many more methods
  - See also Collections.java, e.g., analog to Arrays.java

```java
ArrayList<Integer> il = new ArrayList<Integer>();
il.add(7);
il.add(8);
int sz = il.size();
int value = il.get(0);
il.set(1, 10);
value = il.get(3);  // exception thrown!
```

javarepl.com Check Understanding

- Revisit questions now that you've seen arrays, but do NOT use javarepl.com in answering them now


- Object variables are points to the object
- Strings are immutable
- int value converted to Integer in ArrayList (and vice versa) as needed

---

Sergey Brin

- Simple ideas sometimes can change the world [wikipedia]
  - Works because of scale
  - Co-created pagerank (Larry Page), which evaluates links to a page and the importance of those links, based on the importance of the page from which the links come which ...!