Outcomes:
- Basics of (Java) procedural programming
  - Usage of assignments, conditionals, and loops.
  - Ability to write recursive programs.
- Basics of (Java) object-oriented programming
  - Ability to write programs using classes.
- Fluency with Java and the Java API
  - Knowledge of base classes and specifications.

Website:
- www.cs.cornell.edu/courses/cs1110/2012sp/

Helping You Succeed: Class Resources
- "Interludes." Discussion of some aspect of computing, internet, or CS to help you understand the computing world we live in.
- AEW Workshops. 1 credit, 2 hours. No homework. Small, collaborative classes parallel to course. No class first week. See link on course website, talk to advisors in Olin 167.
- iClickers. Everyone: get your own clicker. By Thursday. We use them to judge the sense of understanding of the class, to encourage staying alert. Part of your participation grade.
- Piazza. Our "town square", a place to ask and answer questions.
- AEW Workshops. 1 credit, 2 hours. No homework. Small, collaborative classes parallel to course. No class first week. See link on course website, talk to advisors in Olin 167.
  - Tuesday: 12:20, 1:25, 2:30, 3:35
  - Wednesday: 12:20, 1:25, 2:30, 3:35

Assignments
- Larger programming tasks (every two weeks)
- First assignment requires mastery
  - Submit, get feedback, resubmit, … until correct
  - Everyone eventually scores 10/10
- Later assignments designed to be fun
  - A4: Color models
  - A5: Graphics and drawing
  - A6: Image processing (jpeg, png…)
  - A7: Breakout game

Reading for This and Next Lecture
- Reading: Sections 1.1, 1.2, 1.3.
  - Lab 1 will give you practice with concepts in 1.2.
  - May not understand all the reading because a lot of new terms, but doing the reading will enhance next lecture.
- Plive (optional): Lesson 0; also Page 1.3, Activity 1-4.1
- Summary of lectures all on course website
- Today and Tuesday:
  - Introduce our IDE, DrJava
  - Introduce types, expressions, and assignments
  - Learn to use DrJava as a “fancy calculator"

Academic Integrity
- We ask you not to cheat, in any way, shape, or form. In return, we try to be fair about the amount of work, in grading the work, and in giving you a course grade. See website for more information
- Do Quiz 0 on Course CMS.
Type: Set of values and the operations on them

- **Type integer**:
  - values: …, –3, –2, –1, 0, 1, 2, 3, 4, 5, …
  - operations: +, –, *, /, unary –
- **Type int**: A FINITE set of integers
  - values: –2147483648, –2147483647, …, –3, –2, –1, 0, 1, 2, 3, 4, 5, …, 2147483646, 2147483647
  - operations: +, –, *, % (7 % 3 = remainder when dividing 7 by 3), /, unary –
  - Principal: These int operations must yield an int.
  - Example: 1 / 2 rounds toward 0
- **Bounds**: Integer.MIN_VALUE - 1, Integer.MAX_VALUE + 1

Memorize this definition!
Write it down several times.

Type: Set of values and the operations on them

- **Type double**:
  - values: –22.51E6 or \(-22.51 \times 10^6\)
    or \(22.51 \times 10^{-6}\)
  - An approximation to the real numbers.
  - operations: +, –, *, /, unary –
  - 1.0/2.0 is 0.5 not 0. Operation / behaves differently for double
- **Double.MIN_VALUE** \(4.9 \times 10^{-324}\)
- **Double.MAX_VALUE** \(1.7976931348623157 \times 10^{308}\)

Smallest POSITIVE value
exponent
mantissa

Casting: Converting Value Types

- **Basic form**: (type) value
  - (double) 2 casts 2 to type double. Value is 2.0
  - (int) 2.56 casts 2.56 to type int. Value is 2
  - Java never does it automatically because it might lose information.
- **Other examples**:
  - (double) (int) 2.56 Value is 2.0
  - (double) 2.56 Value is 2.56

Type: Set of values and the operations on them

- **Type boolean**:
  - values: true, false
  - operations: ! (not) && (and) || (or)
  - ! \(b\) read “not \(b\)”
    true if \(b\) is false and false if \(b\) is true
  - \(b\) && \(c\) read “\(b\) and \(c\)”
    true if both \(b\) and \(c\) are true, false otherwise
  - \(b\) || \(c\), read “\(b\) or \(c\)”
    true if \(b\) is true or \(c\) is true, false otherwise
  - \(i < j\) \(i <= j\) \(i == j\) \(i >= j\) \(i > j\) \(i != j\)
    evaluate to true or false
  - Cannot cast to or from the other types

Precedence of Java Operators (p. 23)

- **Unary operators**: + – !
- **Binary arithmetic**: * / %
- **Binary arithmetic**: + –
- **Arithmetic comparisons**: < > <= >=
- **Equality relations**: == !=
- **Logical and**: &&
- **Logical or**: ||

You will practice all of these in Lab 1.