

1.00 Lecture 1

Course Overview Introduction to Java

Reading for next time: Big Java: 1.1-1.7

Course information

- **Course staff (TA, instructor names on syllabus/FAQ):**
 - 2 instructors, 4 TAs, 2 Lab TAs, graders
 - **Course Web page contains all course materials**
 - **Grad students: register for 1.001 or 1.002, not 1.00**
 - Most should register for 1.001: 9 units of G credit
 - If you need 12 units, register for 1.002 and do 3 extra credits or project
 - **Prerequisite: 18.01**
 - Calculus is used in homework and some lectures
 - **Recitations. Sign up on Web starting 8pm tonight**
 - Recitations scheduled Monday and Tuesday 10am thru 4pm
 - **TA office hours. Schedule, room posted on Web**
 - Wednesday and Thursday 5pm-10pm
 - **Instructor office hours Mon, Wed 4:30-5:30pm and by appt**
 - **Text:**
 - Horstmann, Big Java, 4th ed. 3rd ed out of date but usable
 - Java online tutorials: <http://download.oracle.com/javase/tutorial/>
 - Press, Numerical Recipes in C: www.nrbook.com/a/bookcpdf.php
-

Academic honesty

- **You must write your homework Java code yourself. Problem sets are individual work**
 - You may ask for help on the approach to a problem set, but not directly view or copy anyone's answers.
- You may collaborate on active learning (in-class exercises) **except** :
 - **You must do them yourself if you don't do them in class**
- Quizzes and final exams are individual work
- Please read, sign and return the academic honesty form handed out today.

Course goals

- **Core concepts of software development**
 - Software program structure and introduction to design
 - Software development and debugging/testing
- **Programming in an interactive, object oriented environment in Java (C# is very similar):**
 - Control, methods, classes, objects, inheritance, recursion
 - Eclipse interactive development environment (IDE)
- **Use of computation for scientific, engineering, management problems**
 - Homework problems in engineering science and systems
 - Numerical methods: matrices, linear systems, integration, root finding

Course goals, p.2

- **Graphical user interfaces**
 - Java Swing, events, inner classes
 - Model-view-controller, drawing/transformations
- **Communication**
 - Streams, input and output
- **Sensors (Phidgets), concurrent processing**
 - USB/analog sensors: pressure, light, motion, etc.
 - Threads, multiprocessing, GUI interactions
- **Data structures, sorting**
 - Trees, hash tables, linked lists, stacks, queues, sorting

Computer options

- **Your own laptop or desktop computer**
 - Windows 7/Vista, Linux (Ubuntu) or MacOS X with 1GB of RAM
 - Install Java, Eclipse (instructions on Web site and handouts)
 - There may be some issues using Phidgets with Linux
- **Loaner laptop computers from IS&T**
 - Request online: <http://ist.mit.edu/services/hardware/lcp>
 - Install Java, Eclipse
- **Athena Linux workstations**
 - Eclipse, Java available; same as on laptops
 - Log on; type 'add eclipse -sdk'; type 'eclipse &' (no quotes)
- **Laptops will be used in every class starting Friday**
 - Active learning: exercises, labs in every lecture
 - If you don't have a laptop, you can share in lecture, recitation
- **Bring laptops to recitations and office hours**
- **Lecture on Friday, recitations next week cover Eclipse, Java**

Course materials

- **Lecture notes**
 - Posted without exercise solutions two classes ahead
 - Solutions to exercises posted after each class
 - Hardcopy handed out at each class
- **Recitation notes posted Friday for following week**
 - No hardcopy
- **Homework**
 - Hardcopy on Fridays
 - Next homework posted one week ahead of hardcopy

Recitations and homework

- **Recitations (6% of grade)**
 - Recitations start Monday and Tuesday
 - Come with laptop to all recitations
 - Review class materials, exercises, homework help
 - 10 students per recitation
 - Mandatory attendance
- **Homework (40% of grade)**
 - 10 homework sets (plus ungraded homework 0)
 - Turn in electronically as zip file
 - Homework due Friday at noon except quiz, holiday weeks
 - 30 points off if submitted late but before Monday noon
 - 0 points if submitted after Monday noon
 - One no-penalty late homework automatically given
 - If two or more late and you have a good reason, see instructor
 - All pairs of homework submissions checked for copying

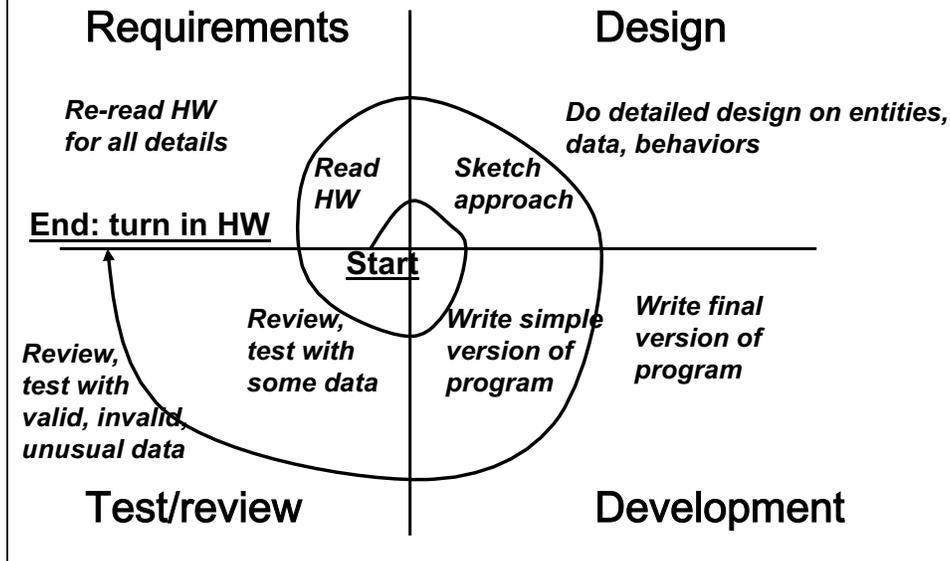
Lectures, quizzes, exam

- **Lectures are active learning with exercises (10%)**
 - Download Java files before almost every lecture
 - Turn in exercise solutions to lectures 3-35
 - Turn in zip file by 8pm that day (1/3 point each)
 - Can skip 3 turn-ins (turn in 30 out of the 33 lectures)
 - No late or excused turn-ins
 - Turn-ins (zip files) sampled by graders for completeness
- **Two quizzes, each 12% of the grade (24%)**
 - On Fridays at regular class time
 - Quiz starts at 3:05pm, ends at 4:25pm
 - Open book, open notes
- **Final exam during finals period (20%)**
 - Open book, open notes

Developing a Java program

- **Read the homework and understand the engineering content**
 - If you don't know what you have to do, you won't be able to do it.
 - Ask questions at recitation
- **Sketch out a design: entities (“things”), data, behavior**
 - Decide how to approach the problem
 - Sketch the approach, in words or pictures. Sketch in stages.
- **Write the program in Java, using Eclipse**
 - Create Java source code files in Eclipse explorer
 - Write Java code using Eclipse editor
 - Write only as much as you think will compile at each stage (e.g., reading the input). Use Eclipse Java compiler
 - Once one stage compiles, write and compile the next. Stage size will increase over the term.
- **Test, mostly by reading/reviewing code in Eclipse**
 - Use the Eclipse debugger to read code
- **Repeat the cycle again to pick up details**

Spiral model of development



Spiral model for 1.00 cont.

- Spiral model is the fastest development method when you haven't written something very similar before (as in 1.00)
- Catching bugs:
 - Code review catches 60-70% of bugs
 - Tests catch 20-30% of bugs
 - Review or read code by using the Eclipse debugger

Java Data Types

- **8 primitive or built-in data types**
 - Boolean (boolean): true or false, not 0 or 1
 - Character (char): 2 bytes long
 - 4 integer types (byte, short, int, long): 1 to 8 bytes long
 - 2 floating point types (float, double)
- These are not objects, unlike everything else in Java
- These are defined (almost) identically on every machine on which Java runs, unlike other programming languages
- Java is a strongly typed language:
 - Every variable in a program must have a declared type

Java Data Types

| | Type | Size (bits) |
|----------|---------|-------------|
| | boolean | 1 |
| | char | 16 |
| Integers | byte | 8 |
| | short | 16 |
| | int | 32 |
| | long | 64 |
| Reals | float | 32 |
| | double | 64 |

Java Data Types

| Type | Size (bits) | Range |
|---------|-------------|---|
| boolean | 1 | true or false |
| char | 16 | ISO Unicode character set |
| byte | 8 | -128 to 127 |
| short | 16 | -32,768 to 32,767 |
| int | 32 | -2,147,483,648 to 2,147,483,647 |
| long | 64 | -9,223,372,036,854,775,808L to 9,223,372,036,854,775,807L |
| float | 32 | 1.4E-45F to 3.4E+38F (6-7 significant digits) |
| double | 64 | 4.9E-324 to 1.8E+308 (15 significant digits) |

Note the F and L

What data type would you use?

- What would you use to store:
 - Speed of light
 - Your grade in this course
 - Your grade point average this term
 - Number of refrigerators in a room
 - Location of a point on a screen
 - 2^{65}
 - \$234.77
 - Half of \$234.77
 - Bits per second transmitted by modem

A Java program

```
public class welcome {
    // main method called when program starts, by definition
    // Lines starting with // are comments
    public static void main(String[] args) {
        // For text, we use Strings, which are sets of char
        // A String is an object, not a primitive data type
        System.out.println("Welcome to 1.00");
        int students= 120;
        int grads= 12;
        double fractionGrads= grads/students;
        System.out.println("Fraction grads: " + fractionGrads);
    }
}

// what will this program output?
```

Things to do

- **Complete and hand in two items today at class:**
 - Academic honesty form (must sign)
 - Info form, to add you to Web site if not preregistered
- **Computers**
 - Install Java, Eclipse by Friday's class
 - Help session Thu 7-9pm in room TBA (see Web)
 - Request loaner laptop online if needed
 - If approved, pick up your laptop tomorrow or Friday at N42
- **Bring your laptop to lecture Friday**
 - We will learn to use Eclipse, Java
- **Do problem set 0:**
 - Log in to 1.00 Web site tonight
 - Sign up for recitation; they start next Monday and Tuesday
 - Do the first exercise with Java, Eclipse by Friday

MIT OpenCourseWare
<http://ocw.mit.edu>

1.00 / 1.001 / 1.002 Introduction to Computers and Engineering Problem Solving
Spring 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.