

Announcements

Handouts (3) to pick up

1. **Forms to return today after class:**
 - Pretest (take during class later)
 - Laptop information form (fill out during class later)
 - Academic honesty form (must sign)
2. **Materials to read and complete by Friday/Monday**
 - Homework 0 (to be reviewed in tutorial)
 - How to submit homework
 - How to install Java 2 Standard Edition, Eclipse
3. **Lecture 1 notes**
 - Lectures posted in advance; please look at them briefly

1.00 Lecture 1

Course Overview Introduction to Java

Reading for next time: Big Java: chapter 1

Academic honesty

- You may collaborate on understanding lectures, labs, text, tutorials, homework problem statements.
- You may discuss the design of your homework program: options for algorithms, classes, methods...
- You must then write your homework Java code yourself.
- You may get help from students while writing your homework programs only by:
 - Asking them to point out an error, but not to fix it for you.
 - Explaining Java syntax to you. Use a different example than the program you're writing if an example is needed.
- We strongly prefer that you get help from TAs, instructors when writing your homework programs
- You may collaborate on in-class exercises except :
 - You must do them yourself if you don't do them in class
- Quizzes and final exams are individual work

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:**
 - Classes, objects, inheritance, composition, events...
 - Java; C# is very similar
 - Eclipse interactive development environment
- **Use of computation for scientific, engineering, management problems**
 - Homework problems in engineering problem solving
 - Introduction to numerical methods

Course goals, p.2

- **Graphical user interfaces**
 - Java Swing, event models
- **Data structures**
 - Stacks, queues, trees, lists, graphs, ...
- **Algorithms**
 - Sorting, searching, hashing,...
- **Communication, distributed processing**
 - Streams, input and output
 - Threads, brief introduction to World Wide Web

Computer options

- **Your own laptop or desktop computer**
 - WindowsXP, Windows2000, Linux or MacOS
 - 256MB RAM or more strongly recommended
 - Install Java, Javadoc, Eclipse (instructions in tools section)
- **Loaner laptop computers**
 - WindowsXP, Java, Javadoc, Eclipse already installed
 - If you drop 1.00 or fail to turn in 2 problem sets, you must return the loaner laptop immediately
 - If it's lost or stolen, notify us, Campus Police immediately
- **Athena workstations**
 - Eclipse, Java, Javadoc available; same as on laptops
- **Lecture Friday, tutorials next week cover Eclipse, Java, Javadoc**

Laptop computers

- **Limited number of laptop loaners**
 - Request a loaner on the form if you really need it
 - You may use it for other classes, return at final exam
 - If eligible, get laptop computer:
 - Thu 9am-5pm, Fri 9am-1pm
 - Laptops will be used in every class starting Friday
 - Active learning
 - Programming, simulations, short exercises
 - Bring them to tutorials and office hours
 - If you don't get a loaner you can still take 1.00
 - Share a laptop with someone at lecture and tutorial
 - Install Java, Javadoc, Eclipse on your desktop or use Athena

Course information

- **Course staff (instructor name on syllabus):**
 - Instructor, 8 TAs, 2 lab TAs, 8 graders
- **Course site contains all course materials**
- **Grad students: register for 1.001, not 1.00**
- **Prerequisite: 18.01**
 - Calculus is used in homework and some lectures
- **Tutorials.**
 - Monday and Tuesday all day
- **TA office hours.**
 - Wednesday and Thursday afternoon and evening
- **Instructor office hours Mon, Wed 4:30-6pm and by appt**
- **Text:**
 - Horstmann, Big Java
 - Java online tutorials (java.sun.com/docs/books/tutorial)
 - Press et al, Numerical Recipes in C, optional for numerical methods

Course materials

- **Lecture notes:**
 - Posted without solutions one class ahead
 - Posted with solutions after each class
 - Hardcopy at each class
- **Tutorial notes posted every Friday for next week**
 - No hardcopy
- **Homework:**
 - Hardcopy on Fridays
 - Next homework posted one week ahead of hardcopy

Tutorials and homework

- **Tutorials (6% of grade)**
 - Sign up for tutorial sections
 - Tutorials start next Monday and Tuesday
 - Attendance and participation are 6% of grade
 - Come with laptop to all tutorials
 - Review class materials, exercises, homework help
 - 8 students per tutorial
- **Homework (40% of grade)**
 - 10 homework sets (plus homework 0), 40% of grade
 - Homework contains characteristic engineering problems
 - Homework due every Friday except quiz weeks
 - Turn in electronically via MIT server (see homework 0)
 - Print your solutions to review them
 - One no-penalty late homework automatic
 - If three or more late and you have a good reason, see me

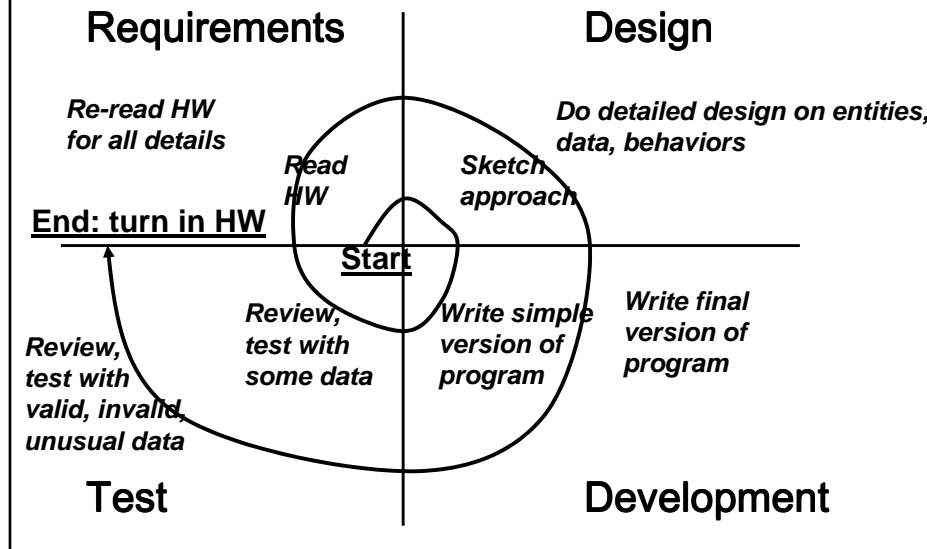
Lectures, quizzes, exam

- Lectures are active learning, exercises (10%)
 - Turn in exercise solutions to lectures 3-36
 - Turn in zip file by 6pm that day (1/3 point each)
 - Can skip 4 turn-ins (turn in 30 out of the 34 lectures)
 - No late turn-ins.
 - Turn-ins (zip files) sampled by graders for reality.
- Two quizzes, each 12% of the grade (24%)
 - On Fridays at regular class time
 - Open book, open notes
- Final exam during finals period (20%)
 - Open book, open notes
- Questions on course requirements, procedures?

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 tutorial
- 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/project
 - 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 | +/- 3.4E+38F (6-7 significant digits) |
| double | 64 | +/- 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

What data type would you use?

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

Very, very rarely use byte, short, float types

A Java program

```
public class welcome {  
    // main method called when program starts, by definition  
    public static void main(String[] args) {  
        System.out.println("Welcome to 1.00");  
        int students= 225;  
        int grads= 15;  
        double pctGrads= grads/students;  
        System.out.println("Percent grads: " + pctGrads);  
        System.exit(0);  
    }  
}  
  
// what will this program output?
```

A Java program

```
public class welcome {
    // main method called when program starts, by definition
    public static void main(String[] args) {
        System.out.println("welcome to 1.00");
        int students= 225;
        int grads= 15;
        double pctGrads= grads/students;
        System.out.println("Percent grads: " + pctGrads);
        System.exit(0);
    }
}

// what will this program output?
Welcome to 1.00
Percent grads: 0.0
// Result of dividing int by int is an int (closed under division)
// How do we get the program to do what we want?
// (double is the data type for real numbers
```

A better Java program

```
public class welcome {
    // main method called when program starts, by definition
    public static void main(String[] args) {
        System.out.println("welcome to 1.00");
        int students= 225;
        int grads= 15;
        double studentDb1= students;
        // One conversion to double is sufficient
        double pctGrads= grads/studentDb1;
        System.out.println("Percent grads: " + pctGrads);
        System.exit(0);
    }
}

// Output:
Welcome to 1.00
Percent grads: 0.06666666666666667
```

Things to do

- **Complete and hand in three items today at class:**
 - Academic honesty form (must sign)
 - Pretest
 - Laptop info form (please fill out even if you have your own laptop)
- **Check Web site to see if you're eligible for laptop**
- **Bring your laptop to lecture Friday**
 - We will learn to use Eclipse, Java, Javadoc
- **Do problem set 0:**
 - Log in to course server after 11pm tonight
 - Sign up for tutorial; they start next Monday and Tuesday
 - Install Java, Eclipse if using your own computer
 - Do the first exercise with Java, Eclipse