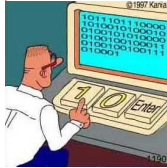


**CS1100 Fall 2008. Instructor: David Gries**

**Website.** [www.cs.cornell.edu/courses/cs1110/2008fa/](http://www.cs.cornell.edu/courses/cs1110/2008fa/)

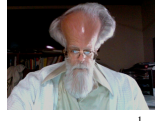
**CS100M: Matlab**

- No prior programming experience
- One semester of calculus
- Math & engineering type problems



**CS100J: Java**

- No prior programming experience
- No calculus
- Non-numerical problems
- Later assignments: processing images, games, playing music



1

**Methods to increase chances of success in the course.**

1. **Section.** In the **ACCEL Lab**. Guided exercises on computer, TA and consultants walking around, helping. Mandatory.
2. **Quizzes.** Let you know what material is important for you to know at that point. You will know quite clearly what the quiz will cover, and everyone is expected to get A on each quiz.
3. **Lectures are not 45 minutes of Gries talking.** See demos of programming and execution of programs in class almost every lecture. There will be some interactive work with you. We will try to make it interesting.
4. **Course text:** CD at the back of the book has 250 2-4 minute lectures, each on one specific point.
5. **One-on-one sessions beginning 3rd week.** You will work for 30 minutes with Gries, TA, or consultant on the computer.

2

**Methods to increase chances of success in the course**

6. **First prog assignment, everyone gets 10. Requires mastery.** You submit and get feedback and resubmit until it is right.
7. **"Interludes"**, discuss some aspect of computing, internet, or CS to help you understand the computing world we live in today.
8. **AEW Workshops.** 1 credit, 2 hours. No homework. Small, collaborative classes parallel to course. No class this week. Talk to advisors in Olin 167.

**Academic Integrity.** We ask you not to cheat, in any way, shape, or form. On our side, we will try our best to be fair about the amount of work we are giving you, in the grading of that work, and in giving you a course grade. For more info, see course website.

Course Management System. Visit [cms.csuglab.cornell.edu/](http://cms.csuglab.cornell.edu/) If you are not listed there, email Maria Witlox, [mwitlox@cs.cornell.edu](mailto:mwitlox@cs.cornell.edu), Ask to add you to CS1110 CMS. Need your Cornell netid.

3

**Read through for Thursday: Sec. 1.3 on classes & objects**

You will not understand it all. It may seem hard. It isn't; it is just new.

**Big problem:** Lots of new terminology. Reading through the section will help you become familiar with it and will make Thursday's lecture seem easier.

Thursday, we will go through the material carefully and use DrJava to show you everything simply and clearly.

Glancing ahead can make introduction of the material in class easier for you.

Learning steadily, in small doses, is superior to cramming every two weeks.

4

**Reading for this and the next lecture:**

Sections 1.1, 1.2, 1.3. Lab 1 will give you practice with concepts and details of 1.2, 1.3

**PLive:** Lesson 0, Lesson page 1.3, Activity 1-4.1.

**Summary of lectures:** On course home page, click on "Handouts" and then "Outline of lectures held so far".

**Today:**

- Introduce expressions in Java (using DrJava)
- Show you around the CD **ProgramLive**

**Dr.Java.** We write programs using the free IDE (Integrated Development Environment) called DrJava. Download it from the course website.

5

**Recitations (Labs) in the Engineering ACCEL LAB**

To get to the ACCEL Lab, go into the Engineering Library in Carpenter Hall, walk straight until you come to a staircase on your left, and go up the stairs.

Do not be concerned if you haven't been able to register for a recitation section. Just go to the one you want this week. We will straighten it out soon, so that you can register.

Here are the times of the recitation-labs: Attend ONE of them.

Tuesday: 12:2, 1:25, 2:30, 3:35

Wednesday: 12:2, 1:25, 2:30, 3:35

6

### Terminology

**Programming language** (Java, C, Fortran, Matlab, Python): a language in which you write programs, often to be executed on a computer.

**Program:** A set of instructions, written in a programming language, to be executed (carried out, performed) to get some task done. Like a recipe in a cookbook.

**Machine language.** The language of instructions that a computer is able to execute (carry out, perform).

**Java Compiler.** A program that translates a Java program into a machine language form so that it can be executed on a computer.

7

**Type:** A set of values together with operations on them.

**Memorize this definition!**  
Write it down several times.

Type **integer:**

values: ..., -3, -2, -1, 0, 1, 2, 3, 4, 5, ...

operations: +, -, \*, /, unary -

Type **int:**  $-2^{31} .. 2^{31}-1$

values: -2147483648, -2147483647, ..., -3, -2, -1, 0, 1, 2, 3, 4, 5, ..., 2147483646, 2147483647

operations: +, -, \*, /, unary -

8

**Type:** A set of values together with operations on them.

Type **double:**  
values: Examples:  $-22.51E6$  equivalent to  $-22510000$  or  $-22.51 * 10^6$   
 $22.51E-6$  equivalent to  $.00002251$  or  $22.51 * 10^{-6}$

An approximation to the real numbers.

operations: +, -, \*, /, unary -

**Type boolean**

Values: true false

Operators: and && or || not !

9

### Precedence of operators (page 23)

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

The lecture also touches on:

- **Types boolean (p. 20) and String (p. 22)**

You will use these things in Lab 01.

10

### Variables. p. 26

- A variable is a name together with a value.
- A variable is a named box with a value in the box.

**Memorize definition!**

**Write it down several times.**

x  **int** Here's variable x, with value 5. It can contain an **int** value.

area  **double** Here's variable area, with value 20.1. It can contain a **double** value.

11

### Declaration of a variable. p. 26

In Java, a *declaration of a variable* gives the name of the variable and the type of value it can contain.

**Memorize definition!**  
Write it down several times.

**int** x; Here's a declaration of x, indicating that it contain an **int** value.

**double** area; Here's a declaration of area, indicating that it can contain a **double** value.

**Assignment statement. p. 27** Execution of an assignment statement stores a value in a variable.

**To execute the assignment**  
<var>= <expr>;  
evaluate expression <expr> and store its value in variable <var>.

x= x + 1; Evaluate expression x+1 and store its value in variable x.

12