CS100J Spring 2008. Instructor: David Gries

Website. www.cs.cornell.edu/courses/cs100j/2007fa

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

Quote for the day:
"I think there is a world market for maybe five computers."
--Thomas Watson, chair, IBM, 1943

CS101J: Transition to OO (using Java)
CS101M: Intro to Matlab

• Both require previous programming experience.

• Self-paced, 1-credit (4 weeks), S/U Engineers take CS100M-CS101J or CS100J-CS100M.

CS211 Computers & Programming

- Uses Java
- Prerequisite: CS100J or CS101J.

CS100J course outcomes:

(1) Basic understanding of object-oriented and procedural aspects of programming, as expressed in Java.

(2) Fluency in Java —able to write programs using classes and subclasses, as well as assignments, conditionals, recursion, and loops.

(3) Knowledge of the basic API classes and their specifications.

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.

Methods to increase chances of success in the course

6. "Interludes", discuss some aspect of computing, internet, or CS to help you understand the computing world we live in today.

7. Use **iclickers?** Wait don't buy yet.

8. AEW Workshops. 1 credit, 2 hours. No homework.
Small, collaborative classes parallel to course. No class this week.
M 0230-0425P or F 0230-0425P. 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 http://cms3.csuglab.cornell.edu/
If you are not listed there, email Amy Fish, amyfish@cs.cornell.edu.
Ask her to add you to the CS100J CMS. Give her your Cornell netid.

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**

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

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

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.

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 –

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

mantissa **exponent**

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 !

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)**

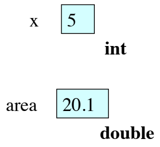
You will use these things in Lab 01.

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.



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

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

Declaration of a variable. p. 26

Memorize definition!

Write it down several times.

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

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.