CS100J September 2007

- **CS100J:** 12 weeks programming using Java, 2 using Matlab. David Gries.
- **CS100M**: 7 weeks of Matlab and 7 of Java. Daisy Fan.
- **CS100H**: 12 weeks Java, 2 Matlab 11:15 only. Graeme Bailey

M

- No prior programming experience
- 7 wks Matlab, 7 wks Java
- One semester of calculus
- Math & engineering type problems

J

- No prior programming experience
- 12 wks Java, 2 wks Matlab
- No calculus
- More non-numerical problems

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

Website. http://www.cs.cornell.edu/courses/cs100j/2006sp/

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 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.

- **Section** is in **the ACCEL Lab**. Guided exercises on the computer, with a TA & consultants there to help.
- **Quizzes** let you know what material is important at eachpoint. Everyone expected to get A on each quiz.
- **Lectures** are not 45 minutes of Gries talking. You will see demos of programming and execution of programs in class almost every lecture.
- Use **iclickers** to get sense of class and encourage participation.
- **Course text** provides alternative learning medium; the **CD** has 250 2-4 minute lectures, each on one specific point.
- **Interludes** discuss aspects of computing or the internet or CS to help you understand the computing world we live in today.
- **AEW classes**. 1 credit, 2 hrs per week, no homework. Small collaborative classes. Not remedial. Students do better. Go to Olin 167 for info.

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

Reading for this lecture:

Sections 1.1, 1.2, 1.3. Lab 01 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”.

Topics of all lectures (this may change): On course home page, click on “Syllabus”.

Today: Introduce expressions and assignment (using DrJava)

Recitations (Labs) Are in the Engineering ACCEL LAB

Do NOT go to the room given in the Course and Room Roster.

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

Programming Languages

- **Computer program:** set of instructions for a computer to perform, or execute. Written in a programming language.
- We use mainly the programming language Java.
- **Other programming language:** Java, C, C++, Matlab, Algol 60, Lisp, Scheme, Ada, Fortran, Javascript, Python, …
- **Compiler:** translates program written in Java into the machine language.

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

Memorize this definition!

Write it down several times.

Type **integer**:

values: …, –2, –1, 0, 1, 2, 3, 4, …

operations: +, –, *, /, unary –

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**:

values: –2147483648, –2147483647, …, –3, –2, –1,
0, 1, 2, 3, 4, 5, …, 2147483646, 2147483647

-2^{31}. It uses 32 bits

operations: +, –, *, /, unary –

$2^{31}-1$. It uses 32 bits

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

Type **double**:

mantissa **exponent**

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 –

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.

x $\boxed{5}$ **int**

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

area $\boxed{20.1}$ **double**

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

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

In Java, execution of an assignment statement stores a value in a variable.

Memorize definitions!

Write them down several times.

To execute the assignment

<var>= <expr>;

evaluate expression <expr> and store its value in variable <var>.

x= x + 1;

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

area= 25.0;

Evaluate the expression 25.0 and store its value in variable area.