Welcome to CS1007!

Introduction to Computer Science in Java

Fall 2002

Section 001: TR 2.40pm - 3.55pm 301 Pupin

Professor Elizabeth Sklar

email: *sklar@cs.columbia.edu*
web: *http://www.cs.columbia.edu/~sklar*
office: 460 Computer Science building (through Mudd)
office hours: Tue 12noon-1pm and Wed 11am-12noon

Class web page:
http://www.columbia.edu/~cs1007

course objectives.

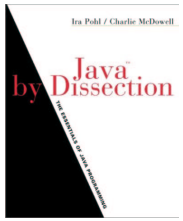
- learn your first (?) programming language
- become fluent in Java
- understand abstract computer architecture
- develop good programming habits
  - structuring code
  - commenting code
  - debugging
- gain skills that can transfer to other programming languages, like C or C++

resources.

- lectures
- textbook
  - Pohl & McDowell: Java by Dissection
- lecture notes
- recitations
- web page
  - *http://www.columbia.edu/~cs1007*
- TAs
- me
  - BUT DON'T E-MAIL ME CODE!!

textbook.

Java by Dissection.
by Ira Pohl and Charlie McDowell
Addison Wesley, ISBN 0-201-75158-5



available at Labyrinth Books
(112th, between Broadway & Amsterdam)

assessment.

- out of 105 points (possible 5 points of extra credit)
- 6 homework assignments (40 points total)
  - homework #1 (5 points) – due Tue Sep 17
  - homework #2 (7 points) – due Tue Oct 1
  - homework #3 (7 points) – due Tue Oct 15
  - homework #4 (7 points) – due Thu Oct 31
  - homework #5 (7 points) – due Thu Nov 14
  - homework #6 (7 points) – due Tue Dec 3
- 3 exams (65 points total)
  - short quizzes (25 points)
  - midterm exam (20 points) – Tue Oct 22
  - final exam (20 points) – TBA (during exam period)
- *note that all dates are tentative and subject to change!*

a word about homeworks.

- should be done on your own, as much as possible
- get help from TAs, me, friends
  but you must acknowledge all help received by citing the names of those who helped you in the comments of your program (I'll explain what comments are before you need to do this)
- this not only protects you from being accused of cheating, but also protects you in case your helper gives you misinformation
- this also lets me know who is really helpful, which is useful in selecting TAs for next semester

homeworks: submission policy.

- homeworks are due on the day that they are due
- here are the rules — please know them well:
  1. all homeworks MUST be submitted electronically by 6AM on the due date
  2. you MUST submit BOTH an ELECTRONIC and a HARDCOPY of all assignments
  3. electronic submissions:
     - only electronic assignments are graded
     - hardcopies are a convenience for the TA and are used to return comments to you — WITHOUT AN ELECTRONIC COPY, YOU WILL NOT RECEIVE A GRADE FOR THE ASSIGNMENT
     - submission time is clocked according to the time of your electronic submission
     - be aware that the system tends to get clogged when too many people try to submit at the same time — so AVOID A LATE PENALTY and don't submit too close to the 6AM deadline!
     - you may have a total of 50 hours grace time for lateness of electronic copies, which may be used up all at once or split between several assignments

  4. hardcopies:
     - hardcopies must be brought to class on the day the assignment is due and deposited in the homework box in the front of the classroom within the first 5 minutes of class
     - a TA will come to class and collect the box at 2.45pm
     - if your hardcopy does not make it into the box, you will lose 1 point and the TAs will print the hardcopy
     - if you attempt to bribe the TA, you will receive 0 for that assignment
     - if you must miss class, have a friend deposit your hardcopy
     - late hardcopies will not be accepted — PERIOD.
- exceptions and extensions are possible, primarily based on MEDICAL EMERGENCIES — circumstances must be documented and suitable arrangements will be made — you must consult me via email on an individual basis

homeworks: regrade policy.

- if you feel that there was an error in grading your homework or exam, then you need to write on a piece of paper a description of the error
- STAPLE the paper to your homework or exam and leave it with me to be regraded
- know that the TAs are given a list of expectations for each homework assignment, quiz and exam problem and told where to take off points — so if your complaint is that too many points were taken off for one kind of mistake or another in your program, then generally those types of things will not change in a regrade
- if there is a genuine error in the marking, like we thought something was missing, but it is really there, then you will likely get points restored
- HOWEVER, a regrade means that the entire assignment or exam will be remarked, so be aware that your mark can go DOWN as well as UP
- regrades take a while to process, so be patient — if you need the work to study from, then make a copy of it before you turn it in for a regrade

homeworks: a word to the wise.

- save early and save often!
- disk drives crash
- floppies have bad sectors
- power supplies fail
- monitors die
- mice get trapped
- paper print-outs are the best security known to mankind

a word about lectures.

- brief notes for every lecture will be placed on the syllabus section of the class web page
- but they are NOT A SUBSTITUTE FOR COMING TO CLASS
- I know, I used to skip classes too
- If you must miss a class, YOU are responsible for getting notes from someone who did come to class
- I will try to post lecture notes on the web before class, BUT:
  - I strongly encourage you to take notes yourself because you learn better when you actually write things down
  - everything I say is NOT in the lecture notes, although anything I say MIGHT be on a quiz, an exam or in a homework, so you need to take notes on what I SAY
  - sometimes there are mistakes in the lecture notes which get caught and corrected during class

a word about quizzes and exams.

- are the only way I know you are doing your own work
- are the only way YOU know you are doing your own work
- are not hard if you really know the material
- here is my weighting scheme:
  1. quizzes: 20%
  2. midterm exam: 20%
  3. final exam: 20%
- quizzes give you practice, so you can learn how to take an exam in computer science

a word about feedback.

- homeworks, quizzes and exams let me know how you are doing
- and in a way, they let me know how I am doing, as a reflection of how you are doing
- but, I welcome feedback from you
- email, anonymous written notes, etc.

a word about academic integrity.

- the work you submit for assessment should be completed ON YOUR OWN
- you may get help from TAs, me, friends
- you must acknowledge all help given
- you should not mail code or copy files
- if someone asks you to do this, *JUST SAY NO!*

topics covered.

- compiling, linking and running programs
- basic Java language elements
- debugging techniques
- program organization
- writing your own Java classes
- graphics
- animation, simulation
- applets, web pages and interfaces

how to learn a programming language.

- YOU are responsible for your own learning!!!
- I will point you in the right direction...
- but YOU must PRACTICE, PRACTICE, PRACTICE...
- and PRACTICE some more!!!
- if you don't understand, then ASK for help!

which environment?

- there are lots of Java compilers and programming environments
- in class, we'll use the CUNIX system and EMACS
- this is a text-based environment, which may be boring, but it's free and it's easier to understand because it doesn't try to do everything
- but the main reason to use the CUNIX environment is that this course is geared toward CS majors, and you will need to be comfortable in a UNIX environment for all other classes in the CS department — so learn it now and don't wait until Data Structures (W3137) when things get really hard

getting started.

- programming is like solving puzzles
- think differently
- the world is now made up of
  objects
  and
  actions
- today's introductory topics:
  - computer basics
  - our first program

computer commands.

- computer follows commands
  commands = series of instructions
- you will learn how to *command* a computer
  command = program = write instructions
- you understand the commands,
  but does the computer?
  that's a question of cognition...
  $\rightarrow$ Artificial Intelligence, Cognitive Science

computer components.

- computer = hardware + software
- a computer is organized into *logical units*:
  - input
  - output
  - memory
  - arithmetic and logic (ALU)
  - central processing (CPU)
  - secondary storage

computer instructions.

- set of instructions = *program*
- types of instructions:
  - machine language
  - assembly language
  - high-level language (e.g., C, C++, Java)
- program is *compiled* into machine language and then *executed (ran)*
- *executing (running) program = job = process = task*

machine language.

- lowest level
  - numeric
- computer is comprised of zillions of *switches* or *relays*
  - switches = ON or OFF
  - relays = OPEN or CLOSED
- hardware position is abstracted into software as 1's and 0's
- 1's and 0's $\Rightarrow$ *base 2*, or *binary*

assembly language.

- medium level, but still pretty low; i.e., hard to read and understand
- "English" words and abbreviations
- examples:
  LOAD
  ADD
  SHIFT
  STORE

high-level languages.

- examples: C, BASIC, FORTRAN, Pascal, C++, Java, LISP, Scheme
- even more like "English"
- high-level languages are
  1. *compiled* into machine language or *object code*
  2. *linked* into *executable code*
  3. *executed* or *ran* as programs

language examples.

- machine language:
  +1300042774
  +1400593419
  +1200274027
- assembly language:
  LOAD BASEPAY
  ADD OVERPAY
  STORE GROSSPAY
- high-level language:
  grossPay = basePay + overTimePay;

Java.

- Java is an *object-oriented* language: it is structured around *objects* and *methods*, where a method is an action or something you do with the object
- Java programs are divided into entities called *classes*
- some Java classes are *native*
  but you can also write classes yourself
- Java programs can run as *applications* or *applets*

our first application.

"hello world"

- typical first program in any language
- output only (no input)

the application source code.

file name = hello.java

```
/*-------------------------------------------------------
   EISklar, 11-Sep-01, hello.java

   This class demonstrates output from a Java application.
-------------------------------------------------------*/
public class hello {
    public static void main ( String[] args ) {
        System.out.println( "hello world!\n" );
    } // end of main()
} // end of class hello()
```

to do.

- get a CUNIX account (if you don't already have one)
- ... and try logging in
- get a copy of the textbook!
- ... and read chapter 1
- check out the class web page:
  http://www.columbia.edu/~cs1007

about me.

- undergrad: Barnard, CS major, class of 1985
- 10 years of industry experience working as a scientific and business programmer
- grad school: Brandeis University, PhD 2000
- previous teaching:
  - Monash University, Melbourne, Australia
  - University of Melbourne, Melbourne, Australia
  - Boston College, Massachusetts
  - came to Columbia in Fall 2001
- research interests center around educational technologies:
  - robotics — RoboCup and RoboCupJunior
  - Internet communities
  - software agents
  - artificial intelligence — evolutionary computation

about you.

- please take out a piece of paper and write down...
  1. your name
  2. your class and major OR if you are a non-matriculating student, categorize yourself
  3. your background in computers, if any
  4. why you are taking this course
  5. what you hope to get out of this course
  6. one sentence about one wonderful thing you did over the break
- ...and give it to me before you leave