Computer Science 121
Intro to Programming: Java - Lecture 1
An intro programming class in Java / 4 credits
Professor Robert Moll (+ TAs) CS BLDG 276
545-4315 moll@cs.umass.edu
Course home page:

http://twiki-edlab.cs.umass.edu/bin/view/CS121Fall2010/WebHome

enrolling: email me your name, student id
What’s it all about?

• This is an introductory programming class in Java programming.
• No programming background is assumed.
• This is a challenging class, with lots of work.
• It relies on a novel textbook delivery system - iJava.
• iJava makes the course somewhat self-taught.
• Don’t take the class just because you need to take some R2 - it’s too challenging for that.
Is this the right class for you?

Do you know how to program in any language?

Is this is the only programming class you’ll ever take?

Do you know your way around your computer? (RAM, downloading, text files, applications, spreadsheets, secondary storage, byte, Internet, www...) If many of these terms/concepts are a stretch, consider taking CS 105, CS 120, CS 145.. ·

How’s your math? You need to be comfortable with basic math, logic, compound interest, simple logic ·

Do you want to take this class? Here for an R2? This one’s too hard. Do an easier R2.
Who are you? FR / Soph / Jr / Sr / Grad / Staff / HS

Major or probably major:
  CS / Sciences / Engineering / Soc Sci - Hum / SOM

Interested in the IT-Minor?

Never / ever programmed? Java / C++ / VB / Other

CS 191P - Python programming
Course Materials

No paper textbook.

We’re using a (free!) electronic book - *iJava*

How do you get to it? **GO TO WEBSITE**

**OWL** - heavyweight assignment system

**IDE** - Dr Java is the class’s integrated development environment

Of course you’ll need Java..
OWL hwk
~ 15 prob per chapter

iJava - narrative
Embedded Problems
Movies etc

Program Problems
~8

website
course
logistics
layout

IDE
DrJava
Special editor
Course Layout

General Pattern will be Tuesday/Thursday Lectures on a subject... then Monday discussion

There are 3 kinds of work in the course (+ 3 exams):

• Embedded Questions
• OWL Hwk Problems
• Programs

CourseWeek link on website gives work due in the coming week or two
More Administration, Details

The grading formula:

- Embedded Questions: 8%
- Programming assignments: 20%
- OWL assignments: 15%
- Midterm exams: 24%
- Final Exam: 33%

Also: to get C in course, you must get C on final!

Collaboration- conceptual collaboration ok, do your own coding (more on this later)

System: You’ll need Java 1.5 or 1.6. And you’ll need DrJava. Information about this online
Agenda

understand, appreciate object-oriented programming, its aims, methods, and (we hope) joys;

Teach yourself to be a skilled beginning Java pgmr

Learn about some additional aspects of computing
What matters
Software is a hybrid endeavor..
Who fails, and why.. phrasebook Java
You must keep up
In general CS is as much about technique as it is about brains
iJava - how to succeed... the flaw in working backwards
Days of the Week

**Monday:** Discussions - These vary. Different sessions for different audiences. Covers materials from the previous week.

**Tuesday:** A general lecture on the week’s chapter’s material.

**Wednesday:** Embedded problems generally due

**Thursday:** Follow-on lecture to Tuesday’s.. But more interactive

Help - some available every day - office hours, learning resource center, discussion sections; stay tuned...
Computer Programming

High Level Languages- human-oriented: accessible syntax; built-in conceptual decomposition

High level languages require language translation

What Java brings..

Object-oriented
Controlling complexity
Hygiene
recycling
Machinery for dealing with web
Hardware neutral (more or less)
But note: ----> Java is hard: it’s for pros
Hardware / Software

Hardware is easy - it’s the physical computer - the chips, the buses, and so forth.

Software is more subtle - it’s the pattern of instructions that directs the hardware. Think of:

Knitting
Origami
Driving directions
Chili recipe
Early model of a computer program: roughly speaking, a sequence of instructions for shopping:

go to store
buy milk
if bananas cost $< 50 cents, buy 6
pay
come home
....

The Java model is more like a library of how-to books with blueprints: e.g., how to frame a house, how to install windows, etc.

Each volume gives sequences of instructions for doing specific jobs.
A First Java Program
public class Howto{
// a baby intro example
    public static void main(String args[]) {
        System.out.println("Welcome to 121");
        System.out.println("3 + 5");
        System.out.println(3 + 5);
    }
}

Notice: this Java program (application) consists of a single class
That class - a single (main) method (subprogram)
That method consists of three “write something to the console” statements, or instructions
Languages, Translators, and Computing

Our program is actually incomprehensible to a computer.

Machine language instructions are VERY primitive.

One aspect of computer science focuses on the translation process -- most importantly, how can a language for programming that’s fairly natural for humans (e.g. Java) be faithfully converted into machine language, a seemingly patternless sea of incomprehensible gibberish?

Languages such as Java, translator called a compiler
01011010101011 0101101010101011 <<<- machine language!
Syntax and Semantics In English

“Tomorrow I’ll come”, and “I’ll come tomorrow” mean the same thing (are semantically the same), even though they’re syntactically different (the parts have been rearranged).

This is a big deal for linguists. Also: natural language (English) is often semantically ambiguous:

“Jay doesn’t kiss Bev because he’s worried about her”

Computer languages are far less flexible. The rules of form for Java (say) are absolutely precise and may not be messed with.

The semantics, likewise, of a Java statement, are unvarying (a println statement prints!).
Errors

- Compile-time errors - syntax, type errors
- Run-time errors - divide by 0
- Logic errors - Everything works fine - get wrong answer

System.out.println(5 + 3; System.out.println(5/0);

System.out.println("area = " + " " + (3*radius*radius));
Assignment for Thursday

1) Look over the course web site

2) Download Java (JDK), and DrJava (IDE). Be sure to do them in this order:

   **first**: Java JDK;

   **second**: DrJava

3) Get your OWL/textbook account going: instructions on website (look under “TextbookStart”)

4) Read Preface, Title Page, Chapter 1; Do intro survey

See me after class if you have administrative issues