CS 121 – Intro to Programming:Java - Lecture 2

Professor Robert Moll (+ TAs)
CS BLDG 276 - 545-4315
moll@cs.umass.edu

Course home page:

http://twiki-edlab.cs.umass.edu/bin/view/Moll121/WebHome

First OWL assignment was due today
First programming assignment is up, due Friday at 5.
Office hours are off and running today - see website
Also: survey; OWL #2; Read Ch 3 for next Tuesday!

Honors section: meet me after class!!!!
Course Materials:
No textbook. We’re using a (free!) electronic book

Additional features:

the Wiki - the web world for course administration

OWL - heavyweight assignment system

IDE - Dr Java is the class’s integrated development environment
Is this the right class for you?

• Do you know how to program in any language? (if you do, consider 191B..)

• Do you know your way around your computer? (RAM, downloading, text files, applications, spreadsheets, Excel, 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? If you’re here because you need an R2, you’ve got the wrong course. This one’s too hard. Do an easier R2. (Also - if you’re a computer art student, see me after class)
More Administration

The class is organized into four discussion sections

The grading formula:

- Embedded Questions: 8%
- Programming assignments: 20%
- OWL assignments: 15%
- Midterm exam: 21%
- Final Exam: 36%

Also: to get C in the course, you must get at least a C on final!

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

System: You’ll need Java 1.5 (1.6) You’ll need Dr Java. Information about this online
The layout of the course:

Tuesday: a big-picture lecture day

Thursday - Discussion - will be just that: discussion, + a fast-paced interactive problem-solving experience.

You are expected to attend class - but roll isn’t taken.

Miss classes and lectures at your own peril.

Bring your brains.

For Thursday: Do second homework assignment!
Bring your questions!
The pieces of the class

The textbook

The IDE - Dr Java

The website (has: office hours, assignments, lec notes, etc)

OWL weekly hwk

Programming projects

2 exams

Lecture

Discussion

Office hours
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:

Knitting
Origami
Travel instructions
Chili recipe

Fact of Life: in the computer world, hardware can vary!
Why Java?

• Designed for big projects, complexity control
• Machinery for programming the web
• Hardware neutral (more or less)
• Object-oriented means: main currency are objects (rather than simply statements)
• Objects are realizations of blueprints, called classes
• There are libraries of thousands of these, waiting to be recycled
A typical Java program
A very, very simple application

Class G

main
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);
    }
}

Prints:
Welcome to 121
3 + 5
8
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));
How + works

3 + 5 \rightarrow 8

“three” + “five” \rightarrow threefive

“three” + 5 \rightarrow three5

“three” + 5 \rightarrow three 5

“three” + (5 + 4) \rightarrow three9

“three” + 5 + 4 \rightarrow three54

5 + “three” + 4 \rightarrow 5three4
Languages, translators, and computing

Our little program is actually incomprehensible to a computer.

How can a language for programming that’s fairly natural for humans (e.g. Java) be faithfully converted into machine language gibberish?

For languages such as Java, the translator is called a compiler.

```
01011010101011
01011010101011 <--- machine language!
11010101101010
10000000000011
```

...
There’s a problem:

My machine language (a Mac, with a PowerPC G4 processor)

Is different from your machine language

(a Windows machine, with an Intel processor)

Can a program that’s been compiled on my machine run on your machine (which has a very different machine language)??
Two translation styles

Compiler (War & Peace, Russian -> English)

Interpreter (Simultaneous translator at the U.N.)

Compiled translation presents problems:

a compiled program on my computer may not run on your computer - my machine language may be different from your machine language.

Java has a solution that (mostly) handles this portability issue:

Two step translation from Java source code to machine level object code
Step 1: Source code is compiled into a universal machine language called bytecode.

Step 2: Each machine invokes its special interpreter for bytecode, which produces that computer’s running machine code.
Java’s Object Model

This subject will occupy us for a good deal of the next month or so!
The Object Model

We model “things” as objects

Objects have attributes, and behaviors

Trip (attributes: start, end, days, distance)

    (behaviors: getDays, setDays, distPerDay ..)

Horse (attributes: name, breed, age, height)

    (behaviors: getAge, setAge ..)

Car

House

Student

Tree
Where are all of these characteristics written down?

Answer: In a class declaration or definition

A class declaration for an object is different from an object, in the same way that the blueprint for a house is different from a house.

Still -

How do you make an object?

How do you invoke its behaviors?
Infant objects

Attributes

name

age (in months)

Behaviors

getName

geteAge

anotherMonth (make kid one month older)
```java
public class InfantTester{

public static void main (String[] args){
    Infant myKid = new Infant("Kit",4);
    System.out.println("name: " + myKid.getName());
    myKid.anotherMonth();
    System.out.println("my kid is now " + myKid.getAge());
}
}
```
Broadly, two kinds of behaviors:

Tell or get behaviors - how old’s the kid, what’s the kid’s name, what’s the kid’s weight in ounces, what’s the kid’s weight in kilos. These do queries, calculations, reports..

Tell behaviors do NOT change the calling object

Mutating behaviors - These alter the calling object - a name is changed, an age is increased, a car’s gas tank is filled, etc. When a mutator does what it does, the state of the calling object changes.
public class Infant {

    private String name;
    private int age; // in months

    public Infant(String who, int months) {
        name = who;
        age = months;
    }

    public String getName() { return name; }

    public int getAge() { return age; }

    public void anotherMonth() { age = age + 1; }

}
Assignment Statements and Identifiers

An identifier is the name of a variable (or method, or class..)

```java
int number = 7; // number now “holds” 7
    number = 4; // number now “holds” 4
    number = number + 2; // number now “holds” 6
```

Assignment is NOT equality!
Assignment is an action operator: Compute the RHS, then copy the result to the variable named on the LHS.