CS 61B Data Structures and Programming Methodology

June 23 2008
David Sun

http://inst.eecs.berkeley.edu/~cs61b/su08/
Course Overview
Course Goals

• Roughly a third of each of these things:
  – Java
  – Data structures and algorithms
  – Programming methodology
Relevance
Prerequisites and Expectations

• Haven’t taken 61A?
  – Talk to the instructor after class.

• Your first time taking summer session course?
  – The summer session runs at 2x the speed.
  – Expect an average of 12 hours of self-scheduled study/programming lab per week, outside of scheduled meeting times.
Course Mechanics
Class Meeting times

• Lectures:
  – Mon/Tue/Wed/Thurs, 11:00am – 12:30pm, 306 Soda.

• Labs sections:
  – 2 times a week, on Mon/Wed, 275 Soda Hall

• Discussion sections:
  – 2 times a week, on Tue/Thurs, 310/320 Soda Hall
Labs and Discussions

• Labs
  – Starting this week.
  – Get your Instructional Account.

• Discussions
  – Will cover new material. You should attend!
  – Know the name of your TA.
  – Also starting this week.
TAs, Office Hours

• Teaching Assistants:
  – Ben Blum
  – Adam Kirk
  – George Wang

• Office Hours
  – David: Mon/Wed 2-4p.m in 360 HMM (building behind Cory Hall)
  – Ben: Tue/Thur 2-4p.m in 611 Soda Hall
  – Adam: Tues 4-5pm and Wed 5-6 pm in 611 Soda Hall
  – George: Thurs 4-5pm and Fri 2-3 pm in Soda Hall
  – Also by appointment
Reaching Us

• Email: cs61b@imail.eecs.berkeley.edu
  – For fastest response
  – Please avoid mailing us directly
  – Please add text “[CS61B]” to the subject header

• Newsgroup: ucb.class.cs61b
  – Required reading.
Textbooks

- **Head First Java**
  - by Kathy Sierra and Bert Bates (second edition, OReilly, 2005);
- **Pragmatic Unit Testing in Java with JUnit**
  - by Andrew Hunt and David Thomas (The Pragmatic Bookshelf, 2004)
- **Objects, Abstraction, Data Structures and Design using Java 5.0**
  - by Elliot B. Koman and Paul A.T. Wolfgang (Wiley, 2005)
  - Downloadable version for much less:
- **Readings are important to the class. Please keep up speed.**
Grading

• 200 total points.
  – Homework (30 pts)
  – Projects (70 pts)
  – Midterm 1 (25 pts)
  – Midterm 2 (25 pts)
  – Final (50 pts)

• Grading Scale
  – No curving. Your grade depends solely on how well you do in this class.
  – 185+ points is an A+, 175-185 is a A, down to D- (75-85).
Assessments

• Homework
  – 12 homework assignments, equally weighted
  – Assigned at the end of each lab section

• Three Projects:
  – Project 1 (individual)
  – Projects 2 and 3 (work in a group of two or three students).

• Midterm 1
  – July 8th, in class 120 minutes

• Midterm 2
  – July 29th, in class 120 minutes

• Final
  – Aug 14th (last day of instruction), in class 180 minutes.
Policy on Lateness

• Late homeworks
  – Not be accepted.

• Late projects
  – Loose 1 percent for every two hours by which you miss the deadline.

• Free three late days (72hours)
  – For project work only.
  – For the second and third project, the number of slip hours is the average between what’s left for you and your partner.
Collaboration

• **Appropriate:**
  – Discussing how to approach homework problems or projects is OK, as long as you write the code.

• **No code rule:**
  – you should never examine or be in possession of another student's solution or partial solution, either electronically or in hardcopy form
  – you should **not** give your solution to someone else.
UCWise
Eclipse
A First Java Program

```java
public class FirstApp {
    public static void main(String[] args) {
        String[] greeting = new String[3];
        greeting[0] = "Welcome to CS 61b";
        greeting[1] = "from David, Ben, Adam";
        greeting[2] = "and George";

        for (String g: greeting) {
            System.out.println(g);
        }
    }
}
```

• Every Java program needs a **class definition**.
• A Class definition consists of a collection of **methods**.
• All the statements under the **main** method is executed when java executes the program.
Difference to Scheme

• Java programs must be *compiled* before you can run them.

Scheme program (.scm) ➔ eval ➔ Result

Java program (.java) ➔ javac ➔ .class file ➔ java ➔ Result
Classes

• Two ways to get classes:
  – Define on yourself (in next lecture)
  – Use one defined by others. Many come in the “Java Standard Library” provided with every Java compiler.

• String class
  – Built-in class
  – A sequence of characters, e.g. the word “Hello”
• Every variable in Java must be declared.
  – `String myString;`  
    (class)D (variable name)
  – No String object is created just yet!

• To create an object
  – `myString = new String("Hello");`
  – `new String("Hello");` constructs a new String object.
  – The assignment statement causes `myString` to reference the object.
Difference Between Scheme and Java

• Every variable in Java has a type (or class), and you must explicitly “declare” it.

• To assign value 1 to x
  – Scheme: `(let ((x 1)))`
  – Java: `int x; x = 1;`
Objects and References

// declare two new string variable and assign two //
   different empty string objects
String s1 = new String();
String s2 = new String();

// constructs a new string; make s1 reference it
s1 = "Hello";

s2 = s1;  // Assign s2 the value of s1
s2 = new String (s1);  // constructs a copy of s1;
      // make s2 reference to it.
Constructors

- **We’ve seen 3 String constructors:**
  - `new String()` constructs an empty string with no characters.
  - “hello” constructs a string containing characters “hello”
  - `new String(s1)` takes parameter `s1`. Makes a copy of the object `s1` references.
    - Other constructors exist. Check the online documentation.

- **Generally constructor always have the same name as the class**
  - Only exception: the special “quotes” syntax.
  - `new String()` and `new String(s1)` are different constructors.
Methods

• Constructors are methods that create new objects
• Once an object is created, other methods can act on the object

s2 = s1.toUpperCase(); //”HELLO”
          (object)   (method)

String s3 = s2.concat(“!!”); //”HELLO!!”

String s4 = “*”.concat(s2).concat(“*”); //”HELLO*”
Reading Assignments

• **Head First Java**
  – Chapters 1-2
  – CS 61B readings (read Scheme to Java article)

• **Next time:**
  – Types
  – Defining Classes