

CS3157: Advanced Programming

Lecture #1

Sept 7

Shlomo Hershkop
shlomo@cs.columbia.edu

1

Welcome

- Today:
 - Basic overview of the course and objectives
- Goal:
 - Things are much easier if everyone knows why they are here, and what we are trying to accomplish.
 - I will not stand here and lecture (although there will be some of that). This is going to be a very interactive course.
 - We will learn about programming ideas while trying to have fun.

2

What?

- CS3157: Third course for CS majors.
- Prerequisites:
 - Intermediate knowledge in Java Programming
 - Object Oriented Programming:
 - What, why, how, and when.
 - Program Designs.
 - Not enough to know how to write the program, need to know how to do it correctly.
- C, C++, perl, unix, general useful tools.

3

Basics

- Instructor: Professor Shlomo Hershkop (shlomo@cs.columbia.edu)
- Class website:
 - cs.columbia.edu/~sh553/teaching/3157f05
 - Check it regularly (at least twice a week).
 - See announcement sections for update info.
- Meet twice a week. There will sometimes be lab sessions, sometimes lectures. See class page for schedule.

4

Resources

- TA's:
 - Mark Tabry
 - Tae Yano
- Class Webboard:
 - Excellent place to post GENERAL questions, and solutions.
 - Good: How do I check what version of java is running?
 - Bad: What is wrong with my code:
`public class foo()`

5

Requirements

- Interest to learn about Computer Science
- Learn to use cool tools
- Learn to make your own tools
- Textbook(s):
 - Textbook can be acquired online or at the Columbia Bookstore.

6

Textbook

- Perl
 - Programming Perl
O'Reilly
- C
 - Brian Kernighan and Dennis Ritchie. (2004)
The C Programming Language, 2nd Edition.
(isbn 0131103628) publisher: Prentice Hall PTR.
- C++
 - Gregory Satir and Doug Brown. (2004)
C++ The Core Language.
(isbn 156592116X) publisher: O'Reilly.

7

Course Structure

- 6 Homeworks – 150 points (50% of grade)
 - Will have about 2 weeks per homework
- Midterm (50 points), Final (100 points)
 - open book
- Homework is very important:
 - Firm believer in hands on learning
 - Start early
 - Come to office hours, and ask questions
 - We are here for YOU!

8

Class participation and Attendance

- Attendance and participation is expected
 - Very interactive lectures & Labs
 - I will learn everyone's name by midterm
 - Useful for your grade
 - Class anonymous feedback system
- If you have to miss class, I expect you to catch up.
 - There will be class notes posted to the website
 - There will be many examples in class only, so make sure to get someone's notes.

9

Homework & Projects

- Written:
 - Will be collected at first class after HW deadline.
- Programming:
 - Online submission
 - Must be able to run on cunix system (this is important).
- Late policy:
 - You have 3 late days that can be used during the semester.
 - Late day is exactly 24 hours.
 - After your late day deadline passes, the homework will not be accepted.
- Extra Credit:
 - To allow for some maneuvering room, there will be extra credit assignments during the semester.

10

Cheating

● Don't

11

Cheating Policy

- Plagiarism and cheating:
 - I'm all against it. It is unacceptable.
- You're expected to do homeworks by yourself
 - This is a learning experience.
 - You will only cheat yourself.
 - My job is to help you learn, not catch you cheating, but...
- Automated tools to catch plagiarizers
 - <http://www.cs.berkeley.edu/~aiken/moss.html>
 - Moving stuff around, renaming, etc. doesn't help
- Results: instant zero on assignment, referral to academic committee
 - Columbia takes dishonesty very seriously
 - I'd much rather you come to me or the TAs for help

12

Feedback System

- Last minute of class will be set aside for feedback:
 - Please bring some sort of scrap paper to class to provide feedback.
 - Feel free to leave it anonymous.
 - Content: Questions, comments, ideas, random thoughts.
- I will address any relevant comments at the beginning of each class.
- Please feel free to show up to office hours or make an appointment at any time.

13

Shopping List

- You need a CS account.
 - <https://www.cs.columbia.edu/~crf/accounts>
 - Try to log into the account asap
- Check out the class page
- Make textbook plans
 - choose a perl reference
- See Homework 0 on class page
 - General overview of cs accounts and some useful information.

14

Example:

- Task:
 - Create a program to run web based chess game, which will be marketed to both desktop and phone users.
 - Any Ideas on how to design the programming backend?
 - Ideas on how to measure requirements.
 - What else?

15

Java Language

- A *programming language* specifies the words and symbols that we can use to write a program
- A programming language employs a set of rules that dictate how the words and symbols can be put together to form valid *program statements*
- *Perl* was originally designed as a logging tool, released by Larry Wall in 1987. Open source and cross platform. Current version 5.8.7.
- Called “duct-tape” of the internet.

16

Difference: Java and PERL?

- Java
 - High Level Language
 - Source code is compiled to byte code
 - Byte code = java execution instructions
 - Byte code executed by java
 - Most functionality built into libraries, very strong graphic capabilities
- Perl
 - Scripting language
 - Very very non rigid structure (i.e. what ever you want)
 - code is interpreted line by line in real time
 - i.e. compiles and executes each time invoked
 - A lot of functionality in base language
 - String handling second to none

17

Difference between CPP and Java

- Java
 - Class definition built from source code
 - No need to worry about memory, GC
- CPP
 - Class definition split into header and def file
 - Responsible for all memory manipulations
 - Multiple inheritances
 - Output of compiler is machine executable code

18

Who cares?

19

Topics to be covered

- Tools of the code trade
- Perl
- C
- C++
- CGI/web based programming
- Outside the box thinking

20

Something to think about:

- You've got someone working for you for seven days and a gold bar to pay them. The gold bar is segmented into seven connected pieces. You must give them a piece of gold at the end of every day. If you are only allowed to make two breaks in the gold bar, how do you pay your worker?

21

Perl

- Perl
 - History
 - Version 5.6+
- What is it?
 - Scripting language
 - Aims to be a USEFUL language
 - Base + tons of libraries
 - Both a compiler and byte code executable
- Where to get it?
 - cpan.org
 - www.activestate.com/Products/ActivePerl/

22

Conventions

- **Something.pl**
 - version: `>perl -v`
 - Location: `>which perl`
- **First line of script**
 - Linux: `#!/usr/bin/perl`
 - Windows: `#!c:\perl\bin`
- **comment lines**
 - Hash (#) to the end of the line
- **Can make the perl script executable (chmod +x command).**

23

Structure

- **Whitespace**
 - only needed to separate terms
 - all whitespace (spaces, tabs, newlines) are treated the same
 - Use them to make the code look nice, easier to look over
- **Semicolons**
 - every simple statement must end with one
 - except compound statements enclosed in braces (i.e., no semicolon needed after the brace)
 - except final statements within braces
- **Declarations**
 - only subroutines and report formats need explicit declarations
 - otherwise, variables in perl are like in shell scripts — they are declared and initialized all at once

24

Variables

- Variables
 - Data dependant
 - No space
 - names consist of letters, digits, underscores; up to 255 chars
 - CASE SENSITIVE
 - Should start with letter or underscore
 - Initialized variables have the value of `undef`

25

Data types

- scalars (\$)
- arrays (@)
- hashes (%)
- subroutine(&)
- typeglob(*)

26

Scalars

- Starts with \$
 - \$first
 - \$course
- int, real, string
 - 234
 - -89
 - 36.34
 - "hello world"
- Context dependant
 - \$name = "shlomo";
 - \$name = 123;

27

Arrays

- Starts with @
- Order list of scalars
 - `@class3157 = ("shlomo","weijen","edward");`
- To reference elements, use the variable name with a dollar in front and subscript
- `$class3157[0]; #is shlomo`
- What is
 - 1) `$class3157[-1];`
 - 2) `$a = @class3157;`

28

Hashes

- name/values pairs
- `%phonenumber = {shlomo=>718, barry=>345};`
or
`%phonenumber = {"shlomo",718,"barry",345};`
- Use the name to find the value
`$phonenumber{"shlomo"} #is 718`
- Any other ideas for this?

29

Next class:

- Overview of Projects in the course
- Introduction to Perl. continued

30

Poll

- To better tailor the class content:
 1. Class: CC, GEAS...
 2. Year:
 3. Computer background
 4. Familiar with unix/linux/windows command prompt?
 5. Why are you taking this course, and what are you planning on doing long term.
 6. What languages are you already familiar, how much, and how did you learn it.
 7. Will you be using your own computer or mostly in the lab?

31