CIS 211: Computer Science II

Instructor: Daniel Lowd
Teaching Assistant: Benjamin Mood
Spring 2011

What’s CIS 211 all about?

• Like CIS 210, but more.
• Old topics in greater depth
  — Objects, inheritance, polymorphism
  — Java collections (arrays, ArrayList)
  — Linked lists
  — Recursion
• New topics
  — Exceptions and error handling
  — Computational complexity (CIS 313, 315)
  — Searching and sorting (CIS 315)
  — Binary trees (CIS 313)
  — Design patterns, OOP

Tentative Course Outline

• Week 1: Inheritance (Ch. 9)
• Week 2: ArrayIntLists (Ch.15)
• Week 3: Linked lists (Ch.16)
• Week 4: Java collections (Ch. 11)
• Week 5-ish: Midterm
• Week 6: Recursion (Ch. 12)
• Week 7: Searching and sorting (Ch. 13)
• Week 8-9: Binary (search) trees (Ch.17)
• Week 10: Design patterns (Not in book)

Primary Resources

Lecture:
  MWF 11am in 216 Allen
Labs:
  026 Klamath
Textbook:
  Building Java Programs, 2nd Edition
  by Reges and Stepp
Web page:
  http://www.cs.uoregon.edu/classes/11S/cis211/index.php
Office Hours

• Daniel: MWF 1-2pm, Deschutes 239
• Benjamin:
  – T 2-5pm, Klamath
  – Th 2-5pm, Deschutes 233

Supplementary Resources

• Textbook-related-stuff:
  – Practice-It! tool
  – Online video lectures (with book code)
  – PowerPoint slides
    (sometimes different from mine)
• Java API specification
• Other books

Grading

• Similar to CIS 210
• Assignments: 35%
  – 8 assignments total (last one due during Dead Week)
  – Due Thursday at 9:00pm submitted through Blackboard
  – No pair programming
• Lab attendance: 10%
  – New option: Skip lab for the week if you complete and turn-in the lab problems by 9:00am on Tuesday.
• Midterm: 25%
• Final: 30%

Collaboration

• All assignments are to be done independently.
• Looking at anyone else’s code: Not OK
• Sharing pseudo-code: Not OK
• Discussing general homework strategies: OK
• Discussing homework instructions: OK
• Studying CIS 211 concepts together: Awesome
### Tips for Success

- Read the book
- Come to lecture and lab
- Start on the assignments early
  
  - Read the instructions very carefully
- Use Practice-It!
- Ask for help

### Tips for Getting Help

- Be specific. Don’t just say, “my code doesn’t work.”
- Describe the debugging steps you’ve already taken.
  
  - Did you read the book or look up the API reference?
  
  - Did you try to come up with a simpler test case?
- If asking by email, allow 24 hours.
- You’ll get a faster response if your email can be answered in 15 seconds.

### Teach Yourself Programming in Ten Years (ideas by Peter Norvig)

- Many books promise quick success:
  
  ![Java 2 in 21 Days](image)
  
  - Mastering a skill takes about 10 years or 10,000 hours; programming is no different.
- More: [http://norvig.com/21-days.html](http://norvig.com/21-days.html)

### Peter Norvig’s recipe for success:

- Get interested in programming, and do some because it is fun.
- Talk to other programmers; read other programs.
- Program.
- Get a degree in computer science. (Optional)
- Work on projects with other programmers.
- Work on projects after other programmers.
- Learn 6+ languages.
- Understand the computer running the programs.
- Get involved in language standardization.
- Get out of language standardization.
What if you don’t want to be an expert?

- Computer science is prevalent in more and more fields:
  - Biology: Computational biology, bioinformatics
  - Physics: Simulations
  - Math: Theoretical computer science
  - Linguistics: Computational linguistics
  - Sociology: Social network analysis
- Any knowledge of programming is handy.
- CIS 200-series can help with this, too.