CSCE 156: Roadmap

Dr. Chris Bourke

Administrivia

• Role
• Blackboard:
  – Schedule
  – Office Hours, TAs, Help Session
  – Materials: lecture notes, assignments, resources
  – Anonymous Suggestion Box
  – Labs
• Syllabus & Policies
• Prerequisite: Homework 1

Overall Goal of 155

• At the end of CSCE 155 you should be able to approach a reasonably complex problem, design a top-down solution, code a program in a high-level programming language
• Key realizations:
  – Computers and programs are not problem solvers, YOU are!
  – Computers are dumb, they do exactly as they are told, there are no ghosts in the machine
  – Programs only automate solutions

Overall Goal of 156

• By the end of 156 you should be able to approach a reasonably complex problem and ask design questions first:
  – What entities/objects would be appropriate to model this problem?
  – How would these objects interact to solve the problem?
  – What data structures would be the most appropriate or efficient to use?
• Fundamentally different problem solving approach: bottom-up

Approximate Schedule

• Week 1 – 3: Intro to Java/PHP
• Week 3 – 6: Object Oriented Programming
• Week 7 – 8: Database Design & SQL
• Week 9: Database Connectivity
• Week 10: List ADTs
• Week 11: Algorithms & Algorithm Analysis
• Week 12: Searching & Sorting
• Week 13: Stacks & Queues
• Week 14 – 15: Binary Search Trees, Heaps

Assignments

• Assignment 1
  – 3 basic programs in PHP or Java
  – Individual, no partners
  – No late passes allowed
• Assignment 2 – 6: Database Application
  – Basic Object Design/EDI
  – Application design & implementation
  – Database design
  – Database integration
  – ADT design & integration
• Assignment 7: mostly written
Application Assignments

• Themed (changes semester to semester)
• Each phase is graded based on correctness and design
• A “living” Design Document will be maintained and updated in each phase
  – Document due 1 week prior to the assignment
  – Each iteration is graded, feedback given
  – Only the final iteration’s grade counts
  – Expected to follow IEEE template

Working in Pairs

• Everyone is *highly encouraged* to work in pairs
  – May discuss, at a high-level, with other teams, but all work must be original
  – Shared work
  – Easy to bounce ideas off of each other
  – Development of “soft-skills”
    • Communication
    • Team work
    • Conflict resolution
  – Careful: partners can mooch, flake out, disappear.
  Choosing to work in pairs means you’re responsible for yourself and your partner!

Keys to Success

• Take the long-view: make sure you understand where we are going; read all the assignments *now*
• Start early
• Have a good, well thought-out design before you even open Eclipse!
• Design your test cases before coding!
  – A test case is a known input/output pair (do it by hand)
• Use proper debugging techniques (a lab is available if you failed to pick up these skills!)
• Ask questions!
• Attend office hours and Help Sessions!

Keys to good Design

• Design comes before code!
• Design objects *first*, then let them interact
• “Smart data structures and dumb code are a lot better than the other way around”
• “It is okay to throw one away” –Eric Raymond

Resources

• Instructor, GTAs, UTAs
• Weekly Lab/Homework Help Sessions
• Webgrader
  – [http://cse.unl.edu/handin](http://cse.unl.edu/handin)
  – Test homework 0 *now*
  – [http://cse.unl.edu/~cse156/grade](http://cse.unl.edu/~cse156/grade)
  – Demo
• Office hours, peers, etc.

Honors/Extra Credit

• Higher expectations
• Faster paced (20-30% faster) to allow for additional/advanced topics
• Each assignment: bonus items are *required*
• At least one additional assignment (JPA or JSP)
• Bottom line: more work, more challenging