

Computer Science II

CSCE 156/156H

Course Introduction

Spring 2021

Dr. Chris Bourke

Outline

- Course Overview
- Course Schedule
 - Labs
 - Assignments
- Project
 - Phases
 - Design Document
 - Collaboration
 - Resources
- Course Resources

Overall Goal of Computer Science I

- At the end of CS1 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
- “By the time you’ve sorted out a complicated idea into little steps that even a stupid machine can deal with, you’ve certainly learned something about it yourself.”

Overall Goal of Computer Science II

- By the end of CS2 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
- “Smart data structures and dumb code works a lot better than the other way around.”

156/156H: Approximate Schedule

- Week 1 – 3: Intro to Java (H: PHP)
- Week 3 – 6: Object Oriented Programming
- Week 7 – 8: Database Design & SQL
- Week 9: Database Connectivity
- Week 10: Lists Data Structures
- Week 11: Algorithms & Algorithm Analysis
- Week 12: Searching & Sorting
- Week 13: Stacks & Queues
- Week 14: Binary Search Trees, Heaps

Labs

- Weekly (Wednesday)
- Starter code provide via github
- Lab sessions: in-person or zoom are if you want to work with a partner or need help from an LA
- Otherwise: completion can be done asynchronously
- Grade based on completion (passing all junit tests)

Assignments

- Assignment 1
 - 3 basic programs in PHP or Java
 - Individual, no partners
- (Project) Assignment 2 – 7: Database Application
 - Basic Object Design/EDI
 - Application design & implementation
 - Database design
 - Database loading
 - Database persistence
 - ADT design & integration
- Assignment 8: written: theory, data structures, algorithms, analysis (“required but not graded”)

Project Phases

- Project-based: multiphase themed project
- Modern Geek Gaming (used game store inventory & sales system)
- Each phase is graded based on correctness and design (see rubrics)
- A “living” Design Document will be maintained and updated in each phase
 - Document due 1 week prior to the assignment
 - Each *draft* is graded, feedback given
 - Drafts are part of the grader, but only the final draft’s *content* counts
 - Expected to follow IEEE template

Collaboration

- Everyone is *highly encouraged* to work in **pairs**
 - May discuss, at a high-level, with other teams, but all work must be original
 - Shared, *equal* 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!
 - Do not undermine the learning process of your partner
 - Use Git!!!
 - Pair up in Canvas

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 “on paper” 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!
- Ask questions!
- Attend help hours, **early!**

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
- Technical Debt: “If you don't have the time to do it right, then you'll have to find the time to do it over.”

Project Resources

- Collaboration:
 - GIT!
 - Saros: <https://www.saros-project.org>
 - More tools: <https://gist.github.com/rouzbeh84/4bafc9fe02edf506d11997c4674b0>
- Gantt Chart

Administrivia

- Course Resources:
 - Canvas (<http://canvas.unl.edu>)
 - Piazza (<http://piazza.com>)
 - GitHub (<https://github.com/cbourne/ComputerSciencell>)
 - Textbook
 - YouTube
 - Course Webpage:
 - Schedule
 - Help Hours, Learning Assistants
 - Labs
- Syllabus & Policies

Resources

- Instructor, LAs
- Prepared Videos
- Lectures & Video Captured lectures
- Course Textbook
- Online readings, lecture notes
- Piazza
- Rubrics & Common Mistakes
- Weekly Lab/Homework Hack Sessions
- Webhandin
- Webgrader
- CodePost
- Office hours, peers, etc.