

# Course Introduction

Bryce Boe

2012/08/06

CS32, Summer 2012 B

# About Me (Bryce Boe)

- Ph.D. Candidate in Computer Science Education
  - Focus on automated assessment
- B.S. in Computer Science from UCSB 2008
- Background in networking and security
- First time teaching

# How to make class smoother

- Feedback, feedback, feedback
  - “Bryce,  $X$  doesn’t make sense”
  - “It might be better if  $Y$ ”
  - “I can’t read your handwriting”
  - “Your going way too fast”

# Outline for today

- Review the syllabus
- Demo Piazza
- Overview HW1
- Demonstrate the submission and feedback process

# Object Oriented Design and Implementation

- Advanced topics in object-oriented computing. Topics include encapsulation, data hiding, inheritance, polymorphism, compilation, linking and loading, memory management, and ~~debugging~~; recent advances in design and development tools, practices, libraries, and operating system support.

# Course Info

- Instructor: Bryce Boe
  - Office Hours
    - Monday 2:15 – 3:15 PM, GSL
    - Wednesday 11:15 – 12:15 PM, GSL
- TA: Saeed Mahani
  - Office Hours
    - Tuesday 2:00 – 3:00 PM, CSIL
    - Thursday 1:00 – 2:00 PM, CSIL
- Website: <http://cs.ucsb.edu/~cs32>

# Required Texts

- Problem Solving with C++, 8th edition
  - Walter Savitch
- Computer Science 32 Reader
  - Available at the Alternative Digital Printing

What do you already know?



# What you should already know

- C and basic C++
  - Loops and conditionals
  - Pointers
  - Functions
  - Classes
  - Recursion
  - Arrays and linked lists
  - Memory allocation and de-allocation
- Abstract Data Types
  - Stacks and queues
  - Trees, binary search trees, and heaps
  - Hash tables

# Student Learning Outcomes

- By the end of this course, you will be able to use the shell to start processes in all possible input/output redirection states.
- Examples:
  - `someprogram < input_file | diff - another_file | grep "some string" > output_file`

# Student Learning Outcomes

- By the end of this course, you will have designed an object-oriented project that demonstrates the use of templates, inheritance, polymorphism, friend classes and operator overloading.
- Project 2 will cover each of these topics

# Student Learning Outcomes

- By the end of this course, you will be able to justify the selection of a particular sort algorithm for a given task.
- Sorts:
  - $O(n^2)$ 
    - Bubble sort, insertion sort, selection sort
    - Quicksort (average:  $n \cdot \log(n)$ )
  - $O(n \cdot \log(n))$ : Merge sort, heapsort

# Student Learning Outcomes

- By the end of this course, you will be able to step-through the process the operating system performs to load a program into memory.
- System calls:
  - fork, exec, dup, wait

# Student Learning Outcomes

- By the end of this course, you will be able to identify where in a process's memory structure a defined variable is located.
- Segments:
  - text
  - data
  - bss
  - heap
  - stack

# Grading Distribution

- 24% Projects (2)
- 24% Labs (5)
- 24% Final (Wednesday Sept. 12)
- 16% Midterm (Wednesday Aug. 22)
- 08% Homework (2)
- 04% Participation

# Participation

- Earned by:
  - Participating in class
  - Answering questions on Piazza
  - Responding to questions on Piazza
  - (Maybe) editing questions and answers for clarity on Piazza
- Participation points are relative to the overall class effort



# Late Submission Policy

- Grading based off your latest (most recent) submission
- 1% off every 5 minute interval late
- Examples:
  - Submission at 00:00:00-00:04:59, 1% off
  - Submission at 00:45:00-00:49:59, 10% off
  - Submission at 04:05:00-04:09:59, 50% off
  - Submission on or after 08:15:00, 0%

# Grading Petitions

- Applies only to tests
- Not required for grading “mistakes”
- Must meet the following conditions:
  - Wait 24 hours after the test was returned to you
  - Provide a written argument that:
    - Clearly states why your answer is suitable for the question
    - Acknowledges your understanding of the *expected* answer
    - Compares the two

# Attendance

- Lectures:
  - Strongly encouraged, not required
- Labs:
  - Required for the first lab (unless already notified)
  - Encouraged but not required for subsequent labs

# Academic Integrity Discussion

- Break into groups of 4 or 5
- Discuss the following questions:
  - What constitutes a violation of academic integrity?
  - What sort of collaboration between students are acceptable?
  - Why are we having this discussion?

# Excluded Topics

- Topics normally taught in CS32
  - Unix programming tools
  - Program building
  - Libraries
- Topics in textbook not covered
  - Separate compilation and namespaces
  - Exception handling

# Course Syllabus

- The official course syllabus is viewable on the course website:
  - <http://cs.ucsb.edu/~cs32>
- It will be updated as necessary

# Online Interaction

- Avoid class-related emails
- Class discussion and online interaction to take place on Piazza
  - <https://piazza.com/class#summer2012/cs32>
- Piazza allows:
  - You to ask questions anonymously
  - Ask questions privately to the instructor and TA
  - You to respond to questions
  - Edit questions and answers

# Piazza Demo

- <https://piazza.com/class#summer2012/cs32>



# Homework 1, FizzBuzz

- <http://cs.ucsb.edu/~cs32/p/hw1>

# Submitting your work

- Refer to [http://cs.ucsb.edu/~cs32/p/automated feedback](http://cs.ucsb.edu/~cs32/p/automated_feedback)
- Feedback is sent to your @cs.ucsb.edu email
  - Most likely forwarded to your umail
  - Also try both servers on:
    - <https://webmail.engr.ucsb.edu/>

# Feedback Caveats

- Output produced prior to a segfault will not be shown
  - It will appear as if your program produced no output
  - There is no other indication your program segfaulted
- Trailing whitespace on lines may not be possible to detect
  - You should never have trailing whitespace unless otherwise specified

# For tomorrow

- Complete HW1
- Read as much of “Operating systems, Unix and shells” in the Reader as you can

Questions?