Introduction

CSCI 201
Principles of Software Development

Jeffrey Miller, Ph.D.
jeffrey.miller@usc.edu
Outline

• Introduction
• Syllabus
My Background

▪ BS in Computer Engineering and Computer Science in 2002
▪ MS, Ph.D. in Computer Science in 2002, 2007
▪ Adjunct Professor from 2002-2007
▪ Assistant/Associate Professor from 2007-2013
▪ Associate Professor of Engineering Practice at USC from 2013-present
▪ Worked part-time and full-time as a system administrator, junior programmer, intermediate programmer, senior programmer, technical lead, chief architect, director of engineering, and founder of a company
▪ Still do consulting work for all types of applications and companies, including expert witness work on legal cases
Research Interests

▪ Computer science education
  › Undergraduate
  › Graduate
  › K12 Science, Technology, Engineering, Mathematics (STEM) education

▪ Ethics with Driverless Vehicles

▪ Intelligent Transportation Systems (ITS)
  › Routing algorithms
  › Dynamic graph algorithms
  › Data gathering and mining

▪ Vehicular Networking
  › Vehicle-to-Vehicle (V2V)
  › Vehicle-to-Infrastructure (V2I)
  › Vehicle-to-Vehicle-to-Infrastructure (V2V2I)
You are in the middle of a three lane road with cars next to you on each side and a large obstacle in your lane. Assume you can’t stop before hitting the large obstacle.

What do you do?
What if you are by yourself and the neighboring vehicles have families of four in them?
Obstacle in Road with Neighboring Vehicles

- What if the neighboring vehicles were school buses full of children?
Obstacle in Road with Neighboring Vehicles

- What if the neighboring vehicles were motorcycles?
  - What if one motorcyclist had a helmet and the other didn’t?
Students lose interest in STEM fields based on

- Stereotypes
  - “Boys are good at math and science, girls are good at art and history”
  - “Only nerdy white guys are programmers”

- Lack of encouragement
  - “My parents don’t even know what programming is”
  - “My friends don’t want to program”
Outline

• Introduction
• Syllabus
Course Description

- Object-oriented paradigm for programming-in-the-large in Java; writing sophisticated concurrent applications with animation and graphic user interfaces; using professional tools on team project.
- We will port over all of your C++ knowledge to Java
- By the end of the semester, you should be more proficient in Java than you are in C++
- You will understand how to program large-scale applications
- You will understand general software engineering principles and methodologies
- Prerequisite – CSCI 104L – Data Structures and Object-Oriented Design
Textbooks

Grading

- Labs 10%  Programming Exam #1 10%
- Assignments 20%  Written Exam #1 10% 20%
- Group Project 30%  Programming Exam #2 10%
  Written Exam #2 10% 20%

Grades will be based on a curve that operates in favor of the students. The following percentages are guaranteed though. If the average is higher than 80%, the average will be the cut-off between a B- and a C+.

<table>
<thead>
<tr>
<th>Score Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>x &gt;= 93</td>
<td>A</td>
</tr>
<tr>
<td>90 &lt;= x &lt; 93</td>
<td>A-</td>
</tr>
<tr>
<td>87 &lt;= x &lt; 90</td>
<td>B+</td>
</tr>
<tr>
<td>83 &lt;= x &lt; 87</td>
<td>B</td>
</tr>
<tr>
<td>80 &lt;= x &lt; 83</td>
<td>B-</td>
</tr>
<tr>
<td>77 &lt;= x &lt; 80</td>
<td>C+</td>
</tr>
<tr>
<td>73 &lt;= x &lt; 77</td>
<td>C</td>
</tr>
<tr>
<td>70 &lt;= x &lt; 73</td>
<td>C-</td>
</tr>
<tr>
<td>67 &lt;= x &lt; 70</td>
<td>D+</td>
</tr>
<tr>
<td>63 &lt;= x &lt; 67</td>
<td>D</td>
</tr>
<tr>
<td>60 &lt;= x &lt; 63</td>
<td>D-</td>
</tr>
<tr>
<td>x &lt; 60</td>
<td>F</td>
</tr>
</tbody>
</table>

In fall 2016, 79.38% of the class earned a B- or better, and the average was 80.17%.
Exams

- The written exams are closed book with one 8.5”x11” double-sided paper of hand-written notes
  - They will consist of theoretical questions and may have code to be analyzed, though very little code will be required to be written.
- The programming exams are open book and open Internet, though no other people can be used (i.e. no posting on discussion boards, email, chatting, texting, etc.).
  - You will need to write a program that compiles based on certain specifications, similar to assignments (though adjusted based on time constraints).

<table>
<thead>
<tr>
<th>Written Exam #1</th>
<th>Monday</th>
<th>July 17, 2017</th>
<th>5:00-6:30p.m.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Exam #1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programming Exam #2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Written Exam #2</td>
<td>Monday</td>
<td>August 7, 2017</td>
<td>3:30-5:30p.m.</td>
</tr>
</tbody>
</table>
Exams (cont.)

- Exams can only be taken on the date and time scheduled (Note: DSP students coordinate with me ahead of time)
- There are no makeup exams
- If you must miss an exam because of an emergency, you must provide me with documentation as soon as possible
  - Approval will be based solely on my discretion based on a documented illness or emergency
- If an excuse is not approved, you will receive a 0 on the exam
- If the excuse is approved...
  - For written exam #1, the percentage will be added to written exam #2
  - For programming exam #1, the percentage will be added to programming exam #2
  - For programming exam #2, the percentage will be added to written exam #2
  - For written exam #2, you will receive an Incomplete in the course and have to make up the exam based on the conditions of an Incomplete
Labs (10%)

- The TA/CPs will lead the lab section each week.
- The lab program will reinforce the topics covered in the lectures.
- Each lab will be graded on effort and attendance.
- You must attend your own lab section.
- Labs must be completed in lab. Lab assistants won’t grade labs until after at least one hour has elapsed.
  - You must show up within the first 10 minutes of the lab session or you cannot attend the lab that week
- Labs are worth 0.7% each, and with 16 labs, that makes 11.2%.
  - That means you can miss almost 2 labs without penalty to your final grade
  - Or you can earn up to 1.2% extra credit
Assignments

- Assignments
  - The program needs to compile, and grading will only occur if the program is able to be run.
  - Grading criteria will be provided at the time the assignment is given.
  - The graders will grade the assignments.
    - If any questions arise based on the grade on the assignment, students should first contact the grader.
    - If a satisfactory resolution is not reached, the TA should be contacted.
    - If a satisfactory resolution has still not been reached, then come to the professor.

- Assignments will be submitted via Github/Blackboard and are due by 11:59p.m. on the due date (see Late Policy).
Project

- The project in the class will be assigned and discussed about half-way through the semester.
- The project will consist of between 4-6 students.
- The software engineering process including high-level requirements, technical specifications, design, architecture, implementation, testing, and formal documentation will be required.

- The project deliverables will be submitted via Blackboard and are due by 11:59p.m. on the due date (see Late Policy).
Late Policy

- There is no late policy.
  - In extenuating circumstances, students may be allowed to submit an assignment late, but only if approved by the professor. This typically should be done before the due date, though I understand some situations may not allow this.
  - For any assignment or project that is submitted after 11:59 p.m. on the due date, the student will receive a 0.
Academic Integrity

- The Viterbi School of Engineering’s policy on Academic Integrity can be found at http://viterbi.usc.edu/academics/integrity/.
- All students are expected to understand and abide by these principles.
- SCampus (http://scampus.usc.edu), the Student Conduct Code, contains information about violating University standards in Section 11.00.
- Any potential violations will be taken seriously and the proper academic process will be followed, including reporting to the USC Student Judicial Affairs and Community Standards (SJACS)