Introduction

CSCI 201L

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

HTTP://WWW-SCF.USC.EDU/~CSCI201
Outline

- Introduction
- Syllabus
Background

- Graduated with a BS in CECS, MS in CS, and Ph.D. in CS from USC
- Taught at Cal State LA for 5 years during grad school
- Assistant/Associate Professor at University of Alaska for 6½ years
- Back at USC as an Associate Professor of Engineering Practices in 2014
- 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?
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 Midterm 10%
- Group Project 30%  Programming Exam #2 10%
  Written Final 10%

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+.

\[
x \geq 93 & \quad A \\
90 \leq x < 93 & \quad A- \\
87 \leq x < 90 & \quad B+ \\
83 \leq x < 87 & \quad B \\
80 \leq x < 83 & \quad B- \\
77 \leq x < 80 & \quad C+ \\
73 \leq x < 77 & \quad C \\
70 \leq x < 73 & \quad C- \\
67 \leq x < 70 & \quad D+ \\
63 \leq x < 67 & \quad D \\
60 \leq x < 63 & \quad D- \\
x < 60 & \quad F
\]
<table>
<thead>
<tr>
<th>Week</th>
<th>Lecture</th>
<th>Date</th>
<th>Lecture Topic</th>
<th>Lab Topic</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>January 11-12, 2016</td>
<td>Introduction, Environment, Methods</td>
<td>Environment Setup</td>
<td>1-8</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>January 13-14, 2016</td>
<td>Classes, Polymorphism, Interfaces</td>
<td></td>
<td>9-11, 13</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>January 18-19, 2016</td>
<td>No Lecture – Martin Luther King Jr.</td>
<td>Inheritance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>January 20-21, 2016</td>
<td>Factory Code Walkthrough</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>January 25-26, 2016</td>
<td>Exception Handling, Garbage Collection, File I/O, Serialization, Generics</td>
<td>Exceptions, File I/O</td>
<td>12, 17, 19</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>February 1-2, 2016</td>
<td>Inner Classes, Event-Driven Programming</td>
<td>Layout Managers</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>February 3-4, 2016</td>
<td>Menus, Toolbars, GUI Components, Graphics, Tables, Trees</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>February 8-9, 2016</td>
<td>Software Engineering, Testing</td>
<td>Event-Driven Programming</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>February 10-11, 2016</td>
<td>Software Engineering Methodologies</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>February 15-16, 2016</td>
<td>No Lecture – President’s Day</td>
<td>No Lab</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>February 17-18, 2016</td>
<td>Project Discussion, Review</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>February 19, 2016</td>
<td>Written Midterm Exam</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>February 22-23, 2016</td>
<td>Concurrent Computing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>February 24-25, 2016</td>
<td>Multi-threaded Programming</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>March 2-3, 2016</td>
<td>Networking Programming</td>
<td>Tables, Trees</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>March 4, 2016</td>
<td>Programming Exam #1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>March 7-8, 2016</td>
<td>More Networking, Serialization Revisited</td>
<td>MySQL Installation</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>March 9-10, 2016</td>
<td>Databases, Database Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>19</td>
<td>March 14-15, 2016</td>
<td>No Lecture – Spring Break</td>
<td>No Lab</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>March 16-17, 2016</td>
<td>No Lecture – Spring Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>March 21-22, 2016</td>
<td>SQL</td>
<td>Networking</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>March 23-24, 2016</td>
<td>JDBC</td>
<td></td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>March 28-29, 2016</td>
<td>Concurrency, Critical Sections, Monitors</td>
<td>JDBC</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>March 30-31, 2016</td>
<td>Locks, Conditions</td>
<td>Threads</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>April 4-5, 2016</td>
<td>Monitors and Locks Programming</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>April 6-7, 2016</td>
<td>Semaphores</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>27</td>
<td>April 11-12, 2016</td>
<td>Parallel Computing</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>April 13-14, 2016</td>
<td>Distributed Computing, RPC</td>
<td>Locks and Monitors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>April 18-19, 2016</td>
<td>RMI</td>
<td>Semaphores</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>April 20-21, 2016</td>
<td>Project Demonstritions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>April 25-26, 2016</td>
<td>Project Demonstritions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>April 27-28, 2016</td>
<td>Project Demonstritions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>33</td>
<td>April 29, 2016</td>
<td>Programming Exam #2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>May 10, 2016</td>
<td>Written Final Exam</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Exams

- The written exams are closed book and 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>Exam Type</th>
<th>Day</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Written Midterm</td>
<td>Friday</td>
<td>February 19, 2016</td>
<td>7:00-8:50p.m.</td>
</tr>
<tr>
<td>Programming Exam #1</td>
<td>Friday</td>
<td>March 4, 2016</td>
<td>7:00-8:50p.m.</td>
</tr>
<tr>
<td>Programming Exam #2</td>
<td>Friday</td>
<td>April 29, 2016</td>
<td>7:00-8:50p.m.</td>
</tr>
<tr>
<td>Written Final</td>
<td>Tuesday</td>
<td>May 10, 2016</td>
<td>4:30p.m.-6:30p.m.</td>
</tr>
</tbody>
</table>
Exams (cont.)

- Exams can only be taken on the date and time scheduled.
- 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 the written midterm exam, the percentage will be added to the written final exam.
  - For the programming exam #1, the percentage will be added to the programming exam #2.
  - For the programming exam #2, the percentage will be added to the written final exam.
  - For the written final exam, you will receive an Incomplete in the course and have to make up the exam based on the conditions of an Incomplete.
Labs

The TA/CPs will lead the lab section each week.
The lab program will reinforce the topics covered in the lectures.
The lab assignment 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.
Labs are worth 0.8% each, and with 14 labs, that makes 11.2%. Labs are worth 10% of the final grade, so you have the potential of 1.2% extra credit or can miss up to 1.5 labs without penalty.
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 lead CP should be contacted. If a satisfactory resolution has still not been reached, then come to the professor.

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

- Project
  - The project in the class will be assigned and discussed approximately 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.

- Assignments and the project will be submitted via Blackboard and are due by 11:59 p.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 11:59p.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/](http://viterbi.usc.edu/academics/integrity/).
- All students are expected to understand and abide by these principles.
- SCampus ([http://scampus.usc.edu](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).