EECS 22L: Software Engineering Project in C Language

Lecture 12

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 12: Overview

- Course Administration
  - Project 2 software releases
  - Team presentations and demos
  - Final exam and competition
- Towards Object Oriented Programming in C++
  - Introduction to C++ concepts from the C perspective
  - Classes, a deeper look

Course Administration

- Completing Project 2
  1. **Software Releases**:
     - Alpha version, 33% complete, due Monday, March 6, at noon
     - Beta version, 66% complete, due Monday, March 13, at noon
     - Final release, 100% complete, due Monday, March 20, at noon
     - ➢ Refer to posted instructions for details on expectations!
  2. **Team Presentations and Demos**:
     - Week 10 during lecture times
       - – 8 teams on March 14, 8 teams on March 16
       - – Voluntary or random order
     - Software presentation and demo (10 minutes total)
       - – By one or a few selected team members
     - ➢ Main features of your Taxi Management system
     - ➢ Demonstration of an hour of taxi management (1 minute!)
     - ➢ Q + A

Course Administration

- Completing Project 2 (cont'd)
  3. **Project Competition**: Week 10 during lab sessions
     - "5 hour shift" of Taxi Management (5 minutes demo time)
       - – Competition Client App provided by instructors
       - – Taxi Management Server provided by each team
     - **Tentative Rules**:
       - – Use configuration file "`NewIrvine.map`", basic protocol (no special)
       - – Use the following taxi-fare pricing structure
         - » Base price of $3.75 applies per ride
           unless the customer is picked up or dropped off at a taxi stand
         - » Distance charge is $2.00 per mile on the shortest path
         - » Taxi drivers earn $0.20 per each city block driven (expense)
       - – Obey all the rules of the City of New Irvine
         - » 45 MPH speed limit implies max. 3 blocks per minute
       - – Use simulated time: 1 minute means 1 hour in real-time
     - ➢ *Maximize your profit = Maximize your bonus points!*

Course Administration

- Completing Project 2 (cont'd)
  4. **Final Course Evaluation**:
     Monday, Feb. 27, 4:30pm – Friday, Mar. 17, 11:45pm
     - Online via EEE Evaluation application
     - Feedback from students to instructors
       - ➢ Voluntary, anonymous, confidential
     - ➢ Help to improve this class!
       - ➢ Student feedback is very valuable

Course Administration

- Completing Project 2 (cont'd)
  5. **Peer Evaluation**:
     Tuesday, March 21, 8am – Thursday, March 23, 6pm
     - Online EEE survey
     - ➢ *Mandatory*, individual, confidential!
       - ➢ Results will be seen only by the instructor and TAs!
     - ➢ Questions:

     Q1: *For all students in your team (including yourself), please estimate the effort to project 2 by each team member*
     - – Effort includes attendance, participation, communication, coding, and documentation.
     - – Scale of 1 ("poor") through 5 ("excellent")

     Q2: *Any additional comments on your team's effectiveness?*
     - – Optional

Course Administration

- Completing Project 2 (cont'd)
  6. **Final Exam**:
     Thursday, March 23, 8am – 5pm (MDE and other rooms)
     - 3 minute individual *oral exam* by instructor
       - Exams per team with members in alphabetical order
       - 20 minute slots scheduled per team
     - Detailed schedule:
       - ➢ To be determined by EEE availability survey

     ➢ *Login to the server and set up your terminal so that each exam can start on time!*

Course Administration

- Completing Project 2 (cont'd)
  6. **Final Exam**:
     Thursday, March 23, 8am – 5pm (MDE and other rooms)
     - ➢ *Present your contribution to your team's project, and explain your source code (at the computer terminal)*
     - ➢ **Oral Exam Questions**:

     Q1: Show your local CVS checkout!
     - ➢ Demonstrate `cvs update`, `cvs status`, or `cvs diff`

     Q2: How does your code fit into your team's software program?
     - ➢ What do you provide? What do you depend on?

     Q3: Show and explain your unit test!
     - ➢ Demonstrate `make test` for your module or component

     Q4: Few ad-hoc questions on your code…

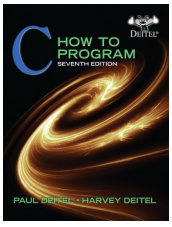
Object Oriented Programming

- Towards Object Oriented Programming in C++
  - C++ can be seen as “improved” C
  - C++ offers a number of new features, including:
    - Inline functions
    - References
    - Default arguments
    - Function and operator overloading
    - Classes and objects
    - Member functions (methods)
    - Constructor and destructor
    - Class and function templates
    - Class inheritance
    - Polymorphism
    - Exception handling

Object Oriented Programming

- “Crash Course” Introduction to C++
  - Selected slides from supplemental text book:

    Paul Deitel, Harvey Deitel,
    “C: How to Program”,
    Seventh Edition,
    Prentice Hall, 2013.



  - Excerpts from Chapters 17 and 18:
    Classes, a Deeper Look