CSC/CPE 203: Project-based Object-Oriented Programming and Design

Professor: William Foote
Section 3, 4 Lecture: MWF 3-4 2-203, lab MWF 4-5 20-0127
Section 5, 6 Lecture: MWF 5-6 52-E26, lab MWF 6-7 20-127
Office: 14-203
Department Phone: 805-756-2824
My phone: 805 756 2921
Office Hours: MWF 2-3, Tuesday 10-12
Email: wffoote@calpoly.edu

Course Description: Object-oriented programming and design with applications to project construction. Introduction to class design, interfaces, inheritance, generics, exceptions, streams, and testing. 3 lectures, 1 laboratory. Prerequisite: CPE/CSC 202 with a grade of C- or better or consent of instructor.

Course Objectives: By the end of the quarter students will be able to:

- Explain key object-oriented concepts including: classes, objects, methods, instantiation invocation, interaction between objects, composition, encapsulation, and use of class libraries.
- Use object-oriented concepts and design to implement moderately sophisticated "large" programs
- Describe the philosophy and mechanics of interfaces including abstraction and specification independence of specification and implementation, contractual requirements in interface implementation, subtypes and type casting, polymorphism, interface hierarchies, and implementation of multiple interfaces.
- Define interfaces in programs to support abstractions according to the principles of interface segregation and dependency inversion where design weaknesses are identified.
- Create a unit test plan for a set of methods in a class.
- Implement program pieces (classes and methods) that use generic types.
- Discuss the differences among generics, subtyping, and overloading.
- Describe the philosophy and mechanics of inheritance including generalization, specialization via extension, difference between extension and composition, subtypes and type casting, polymorphism, and inheritance hierarchies.
- Use inheritance in the implementation of program components such as in the application of a refactoring process to an existing software implementation to improve some aspect of its design.
- Explain the relationship between object-oriented inheritance (code-sharing and overriding) and subtyping
- Articulate design principles including separation of concerns, information hiding, coupling and cohesion, and encapsulation.

- Apply class design principles to the development of a design document for a moderately sophisticated "large" program.

(Some of the) Computer Science and University learning objectives

- Ability to apply knowledge of computing and mathematics
- Ability to analyze a problem and identify and define the computing requirements appropriate for its solution
- An ability to communicate effectively with a range of audiences
- An ability to use current techniques, skills, and tools necessary for computing practice
- Recognition of the need for, and an ability to engage in continuing professional development

Assignments/Grade breakdown:

- 1 final exam (32% of final grade)
- 1 mid-term exam (16% of final grade)
- 10 Lab exercises (10% of final grade) (1% each)
- 6 stages of one larger project (7% each of final grade, 42% total)
- Please see each program description for final grading/rubric details. **There is a strict late policy for all assignments** – 5% off if one day late, 15% off if two days late. No assignments accepted after two days unless there is something on the order of a family emergency.

Required Text: Core Java – Volume 1 - Fundamentals" by Cay Horstmann, ISBN 978-0134177304. The 10th edition is fine. Purchase or rental is a requirement of this course. I recommend the physical book – studies have shown that retention and concentration are improved by using a physical book instead of an electronic book, including with "digital natives." A limited number of copies might be available as an ebook for evaluation via https://bit.ly/2q1DlXP. Be sure to check the terms of use.

Class style and logistics

I expect you to participate in class and engage with the class material. Studies suggest that taking longhand notes is one of the better ways to guarantee your engagement with the material in class)[1] I also expect you to form a community of scholars for the duration of the quarter (and hopefully longer). My teaching style is very interactive – if you want to know more about why see ([2]).

Laptops have been shown to be distracting in lecture[3] and are not allowed unless specified (or a specific exception is negotiated) -- same for cell phones.

---

1 http://www.theatlantic.com/technology/archive/2014/05/to-remember-a-lecture-better-take-notes-by-hand/361478/

2 Applying the Seven Principles for Good Practice in Undergraduate Education" (1991) Chickering and Gamson

3 http://www.yorku.ca/ncepeda/laptopFAQ.html

Lecture and Lab Attendance: Attendance and participation in the course is mandatory. Participation includes responding to questions in class, lab or office hours and making observations or discussing course material in class, lab or office hours. In order to make up a missed exam, you must have a note from a doctor, or other documentation to prove that you absence was unavoidable.

Lab and Lab Exercises: Regular and frequent labs will be assigned and collected each week and, together; will comprise 10% of your course grade. The three hours of scheduled lab time each week is the primary time your instructor will be available for questions and assistance – ***make wise use of this resource!*** You are expected to work on the lab exercises during your scheduled lab time plus as much additional time as necessary to complete them. The lab exercises are designed to familiarize you with some of the concepts necessary to complete your projects and to help you do well on quizzes and exams. You may work on your projects in lab ***after*** completing all currently assigned labs.

Honesty: Although I encourage you to have lively discussions with one another, **all work you hand in must be your own work, unless otherwise specified**. If your program or parts of your program are plagiarized from another student or unapproved source, you will fail the course and a letter will be put in your file with Cal Poly Judicial Affairs.

Class Logistics: Assignments are turned in via github, and grades are given via github. An introductory assignment sets up the needed github repositories. You will have three:

- **materials** is a read-only repository where class materials are distributed.
- **turnin-XXX** (where XXX is your github user ID) is where you turn in assignments. A tool called “checkgit” allows you to make sure that you’ve pushed the right files to the repository.
- **feedback-XXX** is where grade reports are sent back to you. This includes an ongoing cumulative grade report.

It is your responsibility to check the cumulative grade report to be sure that your grade has been correctly recorded.