

CSC/CPE 203-03: Project-based Object-Oriented Programming and Design

Professor: Kirsten Mork

Email: klmork@calpoly.edu

Office: online – check canvas for links

Office Hours: MW 2-3pm, TR 12-2pm

Lecture: MWF 10:10am-11am *online* – *check email / canvas for zoom link*

Lab: MWF 11am-12pm labs *online*

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

Policies and Advisories

1. **The Course Schedule** is on Canvas - check this frequently to see due dates for reading/labs/assignments/exams as well as the video lectures and our meeting times
2. There are generally no make-up exams. You must take all exams at their scheduled time and dates. Check the schedule on Canvas - if you know in advance you will miss an exam, reach out to me as soon as possible. If you miss due to an unforeseen event, contact *me as soon as possible* and we can discuss remakes.
3. Come to office hours! Try problems before coming – you learn from failing, struggling, and trying again – but don't be stuck on a problem for too long when you could get help. I won't often be able to sit and debug with you, but I can give hints and help on where to look if it's a bug I've seen before. *If you are unable to come to office hours*, talk to me and I can schedule an appointment with you for another time. I want to be very available to help everyone.
4. Source code for labs will be tested on the CSL servers (unix 1, 2, 3, or 4). **Be sure to test before submitting.**
5. Source code for projects will use Java 15

Assignments/Grade breakdown

- 4-6 weekly quizzes (24% of final grade)
- 2 larger 'checkpoint' quizzes (12% of final grade)
- ~9 Lab exercises (17% of final grade)
- 5 programs (40% of final grade)
- Video Lecture Participation (2% of final grade)
- Live Lecture Participation (5% of final grade)
 - Online surveys during zoom meeting
 - In class group work

Please see each program description for final grading/rubric details. Also note, **I do not curve. Do not ask if I will curve your grade at the end of the quarter. Check your grade throughout the quarter instead.**

Before Lecture Activities:

- **Videos and Video Surveys-** The schedule (on canvas) outlines the order and due dates for watching these videos. The lecture videos are for your benefit, and you will get credit towards your grade in the class by doing the associated quizzes/surveys by the deadlines. You should ask questions when you have them on Piazza (our course's online forum).
- **Reading (Optional)** - The schedule (on canvas) outlines the order in which topics will be covered in lecture and the associated chapters/sections in the textbook that you should read. I will not test you on material that isn't presented in lecture or the lecture videos, but the reading is an excellent additional source of instruction. Everyone learns differently. If you "learn by reading", then make sure you follow along in the book.
 - **Text:** "Core Java" (Horstmann) - available as an ebook linked on Canvas

Lecture/Lab Attendance and Environment

- **Lecture Hour** – we will meet live on zoom for lecture, class discussion, practice problems, answering questions from the lecture videos, or doing group work. You will get participation credit for participating in lectures and completing associated surveys.
- **Lab Hour** – during the scheduled lab hour, you will be able to get help from the professor or TA as you work on your labs and projects. Some lab periods we may also have in-lab assignments or discussions - the schedule will specify when you are required to attend the lab for these assignments. You will be put in breakout rooms to get help from peers as needed, and can also ask individual questions from your professor / TA. When you have a question, you can **write your name on a google doc of waiting students**.

Assignments

Lab and Lab Exercises

Regular and frequent labs will be assigned and collected each week and together will comprise 17% 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. You may work on your projects in the lab ***after*** completing all currently assigned labs.

Late Policy:

Labs may be turned in 1 day late for up to 95% credit and any other time before Friday of Finals week for up to 90% credit.

- The 95 / 90% late days is only for labs – this does **not apply** to projects

Grading:

Labs will be graded against test cases, in lab demos, by hand, or sometimes all three. For the labs that are graded against test cases, you will have a 24 hour period before the due date to get automated feedback on which test cases you are passing. It is recommended you try and finish your assignment a day early so you have a day to test your code.

Programming Assignments (Course Projects)

There will be 5 programming assignments throughout the quarter, all building upon each other. These assignments will require analysis, design, implementation, testing, and documentation.

Late Policy:

- If you miss the deadline, the maximum you may get on a project is 75%.
- **The exception is**, for **ONE** of the first four programming assignments (***not labs*** and ***not*** project 5) you may have up to a three day extension with no penalty

○ **WARNING**

- Projects build upon each other - you do not want to use your extension unless it is necessary. The class is intentionally paced - if you use this policy as a procrastination tool, your work will pile up. **You will also have to wait longer** to get your grade when you use it.
- Do not lose track of when you use this - you only get one extension, non-negotiable.
- The reason this is given is because I recognize life as a college student is often stressful and I don't know all of the situations you each are going through - your mental and physical health is a priority. In addition, if the policy was "no late work", some students may feel more entitled to ask for extensions in an emergency than others - the extension is meant to be an equalizer.

Additional Late Policies

- To encourage learning after deadlines are missed, you can turn in projects through the Friday of finals week for up to 75% max and lab assignments for up to 90%. While it is highly recommended assignments be turned in on time, I realize school can get busy and this may not always happen. Since doing the labs and projects will directly help you on the quizzes and final, the 75% (projects) and 90% (labs) is to give some extra incentive.

Quizzes and Exams

- **Quizzes** - Instead of midterms this quarter, we will have weekly quizzes (most likely starting Friday of week 3). Six of the quizzes will mostly focus on material from the previous week or two weeks. Two of the quizzes will be slightly longer and cumulative (but only weighted a little more than the others). Hopefully, having smaller tests throughout the quarter will reduce stress by having lower risk evaluations.
- **Final Evaluation** – project 5 will be due finals week this quarter in place of a final exam

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**. Feel free to collaborate on labs, but *projects will be compared using software that can reliably detect similarities in source code*. Though you are encouraged to seek help in tutoring and from the TA and instructor, do not look at **classmates' code, code found online, or past students' code**. **Do not discuss specific implementation of your code**. If your projects or parts of your projects 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. This includes students who copied and **students who were copied from** - you are responsible for the privacy of your source code.

Diversity and Inclusion

Our classroom and lab are to be places of learning and inclusion. Students of all ages, abilities, background, race, sexual orientations, beliefs, religious affiliations, gender identities, and origins are to be treated with dignity and respect as contributors to our scholarly environment.

We will:

- Not be code snobs. We will not assume knowledge or imply things as something that should already be known.
- Accept and learn from our mistakes and the mistakes of others.
- Recognize the contribution of others