COSC220 - Software Design

Fall, 2022

REQUIRED TEXTS:
Core Java, Volume 1 – Fundamentals. 12th edition. By Horstmann. Pearson publishing.

Online Resources

- AdoptOpenJava JDK: https://adoptium.net/temurin/releases/?version=17 (version 17)
- Apache Netbeans: https://netbeans.apache.org/download/nb14/nb14.html (version 14)
- Java 11 API: https://docs.oracle.com/en/java/javase/17/docs/api/index.html

Additional articles as assigned.

INSTRUCTOR:	Stephen Fyfe	**OFFICE:**	140D Vermeer Science Center
		PHONE:	628-5305
		CELL:	641-861-2224
		EMAIL:	fyfes@central.edu
DROP IN HOURS:	MWF: 1:00 – 2:00		T: 11:00 – 12:00
	Other times by appointment		

Do not feel restricted to the hours listed above. Those are times when I will be in my office barring some unforeseen circumstance. In general, if my office door is open then I am available, and you should not feel like you are interrupting me.

If you would prefer, I can also meet over Zoom. Let me know when you want to meet and I will email you a Zoom link that we can use to connect. I may be available over Zoom in the evening if you run into problems. The best approach would be to text me to see if I am free to meet.

COVID Statement: In accordance with current Central College campus policy, our classroom is a mask-friendly, but not required, environment. I have a mask in my office – just let me know if you would like me to where a mask when you visit. If you are not feeling well please do not come to class. I will work with you to make sure you do not fall behind.

Everyone is encouraged to consider getting vaccinated or boosted. As circumstances dictate, Central College policy may be adjusted and I may change my expectations for the classroom.

COURSE DESCRIPTION: Introduces current design methodologies for developing computer programs. Object-oriented programming concepts will be covered. Designing classes and

interaction between objects will be emphasized. In addition, API's for common data structures and programming tasks will be included.

COURSE OBJECTIVES: The objectives for this course include:

1. Students will understand and be able to apply current design techniques for programs with moderate complexity.
2. Students will understand APIs provided for a variety of uses and will be able to incorporate them into their code.
3. Students will be able to implement a simple user interface.
4. Students will be able to implement solutions to moderately complex problems involving multiple interacting components.
5. Students will learn the Java programming language.

Given the above objectives, these are the primary topics for this course:

1. Inheritance and Abstract Classes
2. Interfaces
3. Developing GUIs using the Swing API
4. General UI design principles
5. Exception handling and testing
6. Unit and System testing with JUnit
7. Introductory UML and other object-oriented design techniques
8. Developing classes in Java
9. A variety of APIs (possibly including: data structures, networks, database, graphics)
10. Version Control Systems

COURSE PROCEDURES: This course will include lecture, discussion, and hands-on work. Class time will be spent lecturing and discussing material from the texts, homework questions, and programming projects related to the current topic. This course will typically have homework that will include some reading, written exercises, and hand-on programming exercises. In addition, a significant design and programming project will begin during the second half of the semester and will be worked on throughout the rest of the semester.

This course is being set up to involve students in the course through homework, labs, programming assignments, and class discussions **EVERY DAY**. In this course it is almost always better to spend a little bit of time every day with the course, rather than trying to complete the course requirements using large blocks of time less often.

GRADING PROCEDURES: Students will be evaluated on their understanding of the concepts being covered in class, and their ability to apply those concepts in homework problems and programming projects.

The final grade will be determined by the following distribution:

Programs, Homework, Attendance	30%
Software Project	40%
4-6 Quizzes	30%

and the following TENTATIVE scale will be used to determine the final grade

94 - 100	A	73 - 76	C
90 - 93	A-	70 - 72	C-
87 - 89	B+	65 - 69	D+
83 - 86	B	60 - 64	D
80 - 82	B-	55 - 59	D-
77 - 79	C+	00 - 54	F

Homework and exercises will be due by class time on the day they are due so that they may be discussed in class that day. The software project assignments will be due by the end of the day on the day they are due. Late work **will** be accepted, but **will lose points** at the discretion of the instructor. After an assignment or project has been discussed in class, the late penalty will be severe (at least 1/2 the points).

Notification of participation in college sanctioned events. Mock Trial participants, choir tour participants, athletes, and others who must miss a class for participating in a college sanctioned event are expected to notify me in advance and complete work including tests in advance of the absence. It is the student's responsibility to communicate with me in advance regarding their absences and determine a schedule for make-up work.

Academic Honesty. Collaboration in Computer Science, as in almost any field, is very important. It is important that individual students learn the material. When working on homework assignments, you are encouraged to discuss your answers with other students. **However, you should understand the answers you are turning in!** When developing a program it is often beneficial to talk with others to get their input, however you should not be turning in the work of another individual or group. It is acceptable to look at another individual's code if you are assisting them. You should not, however, let someone look at your code in order to show them how you did it, or to give them specific instructions on how they should change their code (other than to find syntactical errors). If you are getting help, **you should not be looking at someone else's code!**

Plagiarism and cheating of any form are serious offenses and may result in an F for the assignment, the course, or expulsion from the college. The details of Central's Academic Integrity policy are found in the Student Handbook, on the web. A copy will be sent to you via e-mail during the first week of the semester. It is your responsibility to read and understand the

contents of that policy before you submit work to be graded. Questions regarding the policies and enforcement of the policies may be addressed to me during class or during office hours.

ADA Policy
Central College abides by interpretations of the Americans with Disabilities Act and Section 504 of the Rehabilitation Act of 1973 that stipulates no student shall be denied the benefits of an education "solely by reason of a handicap." Disabilities covered by law include, but are not limited to, learning disabilities, hearing, sight, or mobility impairments, and other health related impairments. If you have a documented disability that may have some impact on your work in this class for which you may require accommodations, please see me and Nancy Kroese, Director of Student Support Services and Disabilities Services Coordinator, (x 5247) during the first two weeks of the semester so that such accommodations may be arranged.

COURSE SCHEDULE: The following is a TENTATIVE order in which the topics of the class will be covered. Changes may be made during the semester as needed. Check the webpage for links to exercises and assignments for more details and information about a specific assignment.

Week	Topic	Reading	Homework and Project Work
1 (2 days)	Java Review Java Program Structure OOP Terminology	Chapters 1 - 3	Java Review Program
2 (4 days)	OOP Terminology Java Class Structure, Java API UML introduction – Use Case and Class Diagrams	Chapter 4	Game Use Case diagram
3 (3 days)	GUI Development Swing Components **OOP Quiz**	Chapter 10 - 11	Game GUI
4 (4 days)	Java Generics, Collections and Wrappers	Chapters 8-9	Project Use Case and GUI
5 (4 days)	Programming w/ multiple classes Interfaces, Inheritance and Abstract Classes **Collections Quiz**	Chapters 5-6	Game class diagram and implementation
6 (4 days)	Sequence Diagrams Interacting classes **Inheritance Quiz**		Game Sequence Diagrams

7 (4 days)	Event Handling GUI – MVC Pattern	Chapters 10 - 11	Game Event Handling Design Patterns
8 (4 days)	Java Streams & Sockets Design Pattern Presentations **Events Quiz**	Other material	
9 (1 day)	Java Networking Design Pattern Presentations		Project Registration and Login/Logoff
10 (4 days)	JUnit & Testing **Sockets Quiz**	Other Material	
11 (4 days)	Agile Development	Other Material	Project Follow/Unfollow
12 (4 days)	**JUnit Quiz**		Project Post Message
13 (4 days)	Java Threads	Chapter 12	Project Retrieving and Searching
14 (2 days)	**Threads Quiz**		Project Private Messaging and Notifications (online user)
15 (4 days)	Code Presentations		
16 (4 days)	Project final steps		Project documentation
17	**Final Project Meeting**		