

CS 262: Object-Oriented Design & Programming II

Spring 2023

Instructor:	Michael P. Rogers
Office:	Halsey 214
Office Hours:	MWF 2:00-3:30 PM, Tu 3:15-4:00 PM, Other hours by appointment
Email:	rogersm@uwosh.edu
Phone:	920-424-1388 / 309-825-6454
Class Times:	MWRF 11:30AM-12:30 PM (MWTh, Halsey 202; F, Halsey 101C)
Credits:	4

Description:

A second course in problem solving, software design, and computer programming using an object-oriented language. Problem solving/software design topics include: abstract data types, **generic types**, universal modeling language (UML), simple recursion, ~~unit testing, event-handling, simple concurrency~~. Data structures and algorithms include: binary search, simple sorting algorithms, use of collection classes and their iteration protocols, sequential file processing. Additional topics include: inheritance, polymorphism, **exceptions**, graphical user interfaces, simple use of threads, **database queries**, **simple network I/O**. [Note: the catalog description, with **additions** and **deletions** to reflect current realities.]

Prerequisites: Mathematics 108 or equivalent with a grade of C or better, or qualifying for a higher level mathematics course via the Mathematics Placement exam, and Computer Science 221 or equivalent with a grade of C or better.

Course Website: if it happens in this course, it will be posted on UWO's [Canvas](#) site. Set up notifications to be alerted when announcements are posted, new assignments, quizzes, or notes are posted and graded, etc.

Required Textbook:

COMP SCI 262: Object-Oriented Design and Programming II, Online book by Zybooks.

Subscription Instructions:

1. Sign in or create an account at learn.zybooks.com
2. Enter zyBook code: UWOSHCOMPSCI262RogersSpring2023
3. Subscribe.

A subscription is \$58. Students may begin subscribing on Jan 17, 2023 and the cutoff to subscribe is Apr 30, 2023. Subscriptions will last until May 26, 2023.

Course Outcomes :

Upon successful completion of the course, students will be able to:

1. use UML notation to describe the structure of a class and the relationship between classes
2. identify relationships between classes, including composition, association, and aggregation
3. create subclasses and superclasses for use in inheritance and polymorphism
4. create and use abstract classes and interfaces
5. identify generic types and their uses

6. handle exceptions
7. read from and write to files
8. read from and write to files over a network
9. write simple queries to retrieve data from a database
10. create an event-driven program
11. write recursive methods and identify the parts of a recursive method
12. identify when to use different data structures

Grading Criteria:

Category	%
Exams (3)	55
Labs (≈10)	15
ZyBooks - Participation & Challenge Activities	5
Projects (≈4)	15
Quizzes (≈10)	10

While ZyBooks is nominally only 5% of your grade, statistics indicate that overall grade performance in this class is highly correlated with the amount of time you spend with it.

Grade Scale:

%	≥ 92	90-92	88-90	82-88	80-82	78-80	72-78	70-72	68-70	62-68	60-62	< 60
Letter	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F

Grade Disputes: If there is a mistake in your grade (an inappropriate assessment, math mistake, etc.), you must let your instructor know **within 1 week** of when the assesment was returned if you wish it to be regraded.

Exam Dates (tentative, subject to change):

Exam 1 - March 10

Exam 2 - April 21

Exam 3 - May 12

Late Policies: Douglas Adams once said, "I love deadlines: I love the whooshing sound they make when they fly by". Hopefully you'll agree with the first part, and never have to experience the second. To encourage the latter, in this course exams and quizzes must be taken at the designated times, and late assignments will not be accepted, unless you have an excused absence:

1. illness - a note a medical professional or Dean of Students Office will be required
2. death in the family - send me a link to the obituary as proof
3. UWO excused activity - get your professor / coach to send me a letter
4. weather - you will get one day's credit for each day the university is officially closed

To avoid late assignments is easy-as-pie: just start them when they are turned in, and that way if there is a problem you can discuss it with your instructor or a tutor to get it resolved.

Absences: It has been scientifically proven that the most significant factor for predicting student success is attendance (although whether this is truly causation or merely correlation is another question). Students should attend each and every class, and notify the instructor ahead of time if you will be absent. Attendance may be taken and factored into the Participation category.

Academic Integrity: The purpose of this course is to broaden your programming skills, and more importantly how to think systematically about a problem, how to *problem solve*. For that reason, the work that you turn in must be your own. Note that the goal here is not to force you to work in a complete vacuum: you may have *general* conversations with students to clarify the nature of an assignment, and you can ask for help with debugging, but that second-set-of-eyes-student should not be looking at their code while they assist you.

Sometimes students are unaware of whether or not they have committed plagiarism, but here are some tips:

1. if your problem solving begins with ctrl-C and ends with ctrl-V, you have committed plagiarism.
2. if your problem solving starts at Google and ends at Chegg, you have committed plagiarism.
3. if your problem-solving involves surreptitiously glancing at the exam of the student next to you and doing a *virtual* copy-and-paste, you have committed plagiarism.
4. if that tiny voice inside your head, the same one that inconveniently shuts up entirely while you are working on a tricky assignment, starts making "ahem" noises, you have committed plagiarism.

Let us consider the pros and cons of committing plagiarism.

Pros:

1. You have completed the assignment. Rah!

Cons:

1. You will have missed that exhilarating, ego-boosting, delicious "aha!" moment that everyone experiences when they have, on their own, solved a difficult puzzle.
2. You will be caught, receive a 0 on the assignment/exam, and may face disciplinary action in front of a bevy of grim-faced administrators who you do *not* want to meet.
3. You won't know anything!

So what do you do when you can't solve a problem? The answer is simple, ask your instructor (or lab assistant) for help. We will use the time-tested Socratic method, asking questions that will lead you to the correct answer. Failing that, just write "I have no idea" in bold letters in the comment block at the top of the assignment, turn it in, and you will gain *some* points (more than 0).

Exams and quizzes are to be done entirely on your own. These are closed book, closed notes, because this material is so absolutely fundamental that you **must** have it at your fingertips. See the Dean of Students Office web page [for more information](#).

Also, for an even more detailed discussion of what constitutes academic misconduct, please see the discussion of [UWS Chapter 14, Student Academic Disciplinary Procedures](#).

Coding Standards: In industry, companies use a set of coding standards that all programmers are expected to adhere to. We do the same. A coding standards document, on the Canvas site,

goes into detail (and may evolve over time: if so, changes will be announced in class). Unless otherwise specified, all labs, assignments, etc., must adhere to those standards.

Accessibility: Your instructor is committed to ensuring a fair playing field. If you have a disability and need assistance (e.g., a note taker, certain seating, extra time to take tests, adaptive technology, etc.), please register with the Accessibility Center, and we work hard to accommodate your needs.

Non-discrimination and Anti-harassment: Your instructor is committed to maintaining a harassment-free, welcoming classroom, and to that end will not tolerate discrimination on the basis of race, religion, creed, color, sex, gender, gender identity/expression, ancestry, national origin, age, marital status, preferences in streaming services (just snuck this in to see if anybody's reading :-), relationship to other employees, sexual orientation, disability, veteran's status, membership in the military, arrest/conviction record, political affiliation, or any other protected status.

Feedback: Your instructor thrives on feedback. If a concept doesn't make sense, ask, and if the answer didn't make sense, ask again! There are no stupid questions (and any stupid answers are my responsibility 😊). It may seem intimidating to put yourself out there and admit that you don't understand something, but:

1. if you didn't understand something, there is a good chance that your peers didn't either, and they will (silently, or better yet, by banging on desks like they do in parliament) applaud your act of inquiry
2. you needn't worry about "slowing down the class". Learning never slows down anything.
3. your instructor is intimately familiar with being in situations where things don't make sense: you will receive a very sympathetic hearing.

If you do feel uncomfortable asking in class, please inquire during office hours, on our Discord server, or during a tutoring session (hours TBA)