

Official Course Description (Catalog Description - Outdated):

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, 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, graphical user interfaces, simple use of threads.

Updated Course Description (Unofficial):

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), recursion, event handling, 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 database queries, simple network I/O.

Learning Outcomes:

At the end of the course, students will be able to:

1. use UML notation to describe the structure of a class and the relationships 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, and
12. identify when to use different data structures.

Course Grading Policy

Your final grade for this course will be based on five components:

1. Daily programming activities in the textbook that are called “book quizzes”,
2. Weekly labs,
3. Weekly in-class quizzes,
4. Five (or so) homework programming assignments, and
5. Three in-class exams.

Your overall numerical grade for the course will be computed as the weighted sum of the component grades using the following weights:

Component	Weight
Daily book quizzes (BQ)	10%
Labs (L)	15%
In-class quizzes (Q)	15%
Assignments (A)	15%
Exam 1 (E1)	15%
Exam 2 (E2)	15%
Exam 3 (E3)	15%

Finally, your letter grade for the course will be computed as follows:

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

While this overall grading scheme is fixed, I will be happy to discuss any issue you may have with individual grades. If you notice a mistake or have a question regarding a specific grade, please contact me *as soon as possible*. Do not wait until the end of the semester to bring up grading issues. Also, I will *not* be available to discuss grades after the end of the final week.

Attendance and Participation

You are expected to not only attend **every** class meeting but also to come **prepared** for and **participate** actively in it. Necessary preparation requires you to have studied and assimilated the material covered in previous sessions, to have completed the reading assignment for the day, to have discussed with me outside of class any questions you may have, and to have completed the labs and programming assignments on time. **It is hard to imagine how a student could do well in this course while missing classes or attending them unprepared.** On the positive side, I have high expectations for my students and will always support and encourage you. **I strongly encourage you to ask any question** or raise any issue you have with the course either during class or in my office hours. I will also gladly talk with you outside of office hours: send me email to make an appointment. Keep in mind that I may not be widely available just before an exam or the due date for an assignment since you may not be the only one needing help at the last minute.

Late submissions: THERE WILL BE NO LATE SUBMISSIONS ACCEPTED ON ANY COMPONENTS OF THIS CLASS. If your submission reaches me after the due date/time (even if it is late by only a few seconds, as recorded by Canvas), I will not grade it and you will receive a **zero**. Late submissions can easily be avoided by starting to work on each assignment right away and asking questions early if you get stuck.

The zero-score for late submissions can be waived in only one scenario, namely if you give me a signed note from a doctor or a written justification for the extension from the Dean of Students Office. If you miss an exam, you may be able to take a make-up exam provided you give me a valid justification (see above) ahead of time if possible. Only one make-up exam will be given. It will be a comprehensive exam scheduled at the end of the semester. If you miss a quiz, you may be able to take a make-up quiz, provided you have a valid justification for your absence.

Accommodations

The University of Wisconsin Oshkosh supports the right of all enrolled students to a full and equal educational opportunity. It is the University's policy to provide reasonable accommodations to students who have documented disabilities that may affect their ability to participate in course activities or to meet course requirements.

Students are expected to inform instructors of the need for accommodations as soon as possible by presenting an Accommodation Plan from either the Accessibility Center, Project Success, or both. Reasonable accommodations for students with disabilities is a shared instructor and student responsibility.

The Accessibility Center is part of the Dean of Students Office and is located in 125 Dempsey Hall. For more information, email accessibilitycenter@uwosh.edu, call 920-424-3100, or visit the [Accessibility Center Website](#).

Disclosure: Students are advised to see the following URL for disclosures about essential consumer protection items required by the Students Right to Know Act of 1990:

<https://uwosh.edu/financialaid/resources/consumer-information/>

Collaborating versus Cheating

While it is acceptable to discuss the problem statement, premises, goals, constraints, etc., of the assignments with others, you must submit your work exclusively. You may not "borrow" any piece of code or design of any length from anybody else, unless you can live with a zero and the other potential academic sanctions of cheating (see the [UWO Student Discipline Code](#), Chapter UWS 14).

In conclusion, remember that computer science classes require a lot of work in addition to active participation in class. It takes considerable practice to develop the technical and analytical skills targeted by this course. You will need to spend **at least (and typically more than) three hours of effort outside of class for each in-class hour**. Having said this, I expect every hardworking student to do well in this course.

Have fun this semester and good luck!