Software Design I — Syllabus

CS120-04 (1210) and CS120-05 (2651) 4 credit units

Regular meeting times  M., Tu., W., F., 11-11:55AM/12:05-1PM

Meeting rooms  Centennial 3204 (lecture), Wing 16 (lab)

Lecture/lab  Normally, labs are Tuesdays and Fridays but see the course website for each week’s plan

Course website  https://cs.uwlax.edu/~jmaraist/120-fall-18

Prerequisites  MTH 151, MTH 175, or equivalent placement

Catalog description  An introduction to the fundamentals of software development; including software classes, objects, inheritance, polymorphism, logic, selection control, repetition control, subprograms, parameter passage, and rudimentary software engineering techniques. Students complete numerous programming projects using a modern programming language.

Instructor’s name  Dr. John Maraist

Office location  209 Wing Technology Center

Email  jmaraist@uwlax.edu

About  https://cs.uwlax.edu/~jmaraist

Open-door times and appointments  My open-door times are listed on the course website. To make an appointment for other times, ask by email at least one school day ahead of time.

Textbook  Programming in Java, zyBooks (see p.7)

Additional references  Java: A Beginner’s Guide, Herbert Schildt, Oracle Press; see the course website for other resources

Important dates

Tuesday, September 4  First class

Wednesday, September 5, 8:00AM  First daily assignment due

In the week of October 1-5  Midterm 1

Monday, October 8  Project 1 due

Friday, October 26  Project 2 due

In the week of October 29-November 2  Midterm 2

Monday, November 12  Project 3 due

In the week of December 3-7  Midterm 3

Monday, December 3  Project 4 due

Friday, December 14, 10:00AM  Final exam

Midterm and project due dates are tentative; I will confirm them by two weeks beforehand. Final exam times are set by the university; do not plan to leave for holiday travel until after that date.
Your takeaways from this class

This class offers the opportunity to master the fundamentals of software development. We will use the Java programming language, but the skills we will convey are applicable to most programming and scripting languages in use today. Over the course of the semester, we will examine topics related to software development, including problem solving techniques, fundamental programming constructs, and their application to algorithm design and to the Java programming language. In particular, we study the design of simple algorithms and their implementation as Java programs executed as a single, sequential thread.

Learning outcomes

The specific learning outcomes of this class are:

1. Implement Java software following a disciplined process for creating correct software from a problem description. Modern industrial software practice requires not just coding skill, but also the rigor of processes for both software management and human workflow. Coding skill must be balanced with these values for sustainable technical achievement.
   - Formulate diverse examples of correct behavior across different scenarios for a problem. The skill of projecting from a problem statement to a set of key cases is important as a general problem-solving skill, not only for constructing correct software.
   - Compose executable programs comparing expected correct behaviors against the result of actual method calls. Our goal is correct software, and we make the computer tell us whether our work is correct in order to focus our efforts on unresolved problems.
   - Design solution procedures for problems as English-language descriptions of mechanics of the computational process and their purpose. There are two motivations for including this outcome. First, composing text which forms a standalone technical narrative forces you to think about the bigger-picture steps of your implementation effort, and how they fit together, rather than focusing only on a code-level understanding of your work. Second, communication is as essential in computer science as in any other field. A description of design steps in English is how you will communicate with colleagues, managers, and eventually the teams which you may lead.
   - Compose Java methods implementing these designs. This outcome refers to the way we must understand aspect of our programming language not only in isolation, but also in terms of the way it combines with other language elements.
   - Diagnose and debug flaws in both your own code and others’ code.

2. Assess the correctness of possible software designs for a given problem, and both the appropriateness and correctness of implementa-
tions for a given design or problem.

- Predict the value of expressions and the outcome of programs under various inputs and environments.

This outcome is a dual to the first outcome: we must be able not only to produce correct code, but also to understand the designs and programs produced by others.

3. Explain code execution using object diagrams; explain class properties such as scope annotations and aggregation and inheritance relations using class diagrams. Interpret these diagrams to explain what they illustrate. These diagrams are another way of communicating technical information, and the consistency of their conventions allows them to be easily read by a broad audience.

Specific technical content

The elements of algorithm design and Java in scope for this class include non-parallel control instructions including assignment, method calls (both void and with value-returning), selection (if and switch) and iteration (while and for) constructs, and their compositions; static and object-dispatched method construction; and simple inheritance and method override. Other specific Java features which we will cover include:

- Expressions using literals, variables, parentheses, mixed types, widening, casts, issues of precedence among operators with and without parentheses, and the following operations:
  - For integer and real numbers, - (unary negation), +, -, /, *, %, ++ (postincrement), – (postdecrement), =, !, <, <=, >, >=
  - For boolean values: !, &&, ||
  - For string values: + (concatenation)
- Variable declarations and the keywords final, private, public and protected
- The principle of information hiding, and its use to select from local and class scope alternatives.
- Debugging, including cases of nonterminating loops, NullPointerException and ArrayIndexOutOfBoundsException, and using the technique of inserting output statements.
- One-dimensional arrays.
- Using import statements for standard libraries.
- Object and class concepts: Compose instructions to assign objects, object binding, constructor overloading, the null, this and super notations, orphan objects and garbage collection.
- Algorithm patterns including variable content swap, cascading if instructions, counting loops, linear search, selection sort, object access shared by multiple classes, method callback.
- Modern programming system make heavy use of standardized provided libraries, so we will become familiar with libraries, including for user I/O, mathematical computation, string manipulation, constructing
and using graphic user interface (GUI) component, and handling user events. Specifically, we study these standard Java classes: `Object` (and its methods `equals`, `toString`), `String` (length, `charAt`), `Math` (random, `abs`, `sqrt`, and trigonometry functions), and the GUI classes `Container` (add, remove, repaint), `JFrame`, `JComponent` (paint), `JButton` and `JTextField`.

- Event handling in GUIs with buttons and text fields.
- Any local culture of programmers (such as a workplace or a community project) will adopt or be assigned programming stylistic conventions for programs and for comments. We will point out and follow a number of these guidelines, including using meaningful identifiers, intelligent inclusion of comments, and proper indentation patterns, as well as general professional habits.

**Your time commitment**

This class is focused tightly on mastering a specific set of skills, and on the knowledge associated with those skills. Mastering any new mental or physical skill requires practice and discipline. You should plan to spend an average of about ten to twelve hours a week (not counting our class meetings) preparing for class, working assignments, and otherwise studying or practicing class material. As with a sport or musical instrument, you will not develop programming skills without committing serious and regular effort to actually programming.

**Cumulative nature of this material**

The focus on skills, and the elementary nature of the material we cover, means that this class is highly cumulative. Topics from later in the class rely very heavily on skill mastered earlier. Even where assignments and exam seem to address only later topics, it is unavoidable that earlier material will be essential components of later work.

It is important to master all of these skills over the course of the semester, but we recognize that some people take longer to master some aspects of algorithmic thinking and programming. So when computing final grades, I will replace earlier grades from a particular skill category with the weighted average of later grades from the same category. The Evaluating assessments section of this document details exactly how this calculation will work. (Miscellaneous and administrative assignments will have separate categories for grading purposes.)

**Forms of assessment**

**Preparatory exercises** are readings and problems on a topic which you complete before a lecture on that topic. Newcomers to university-level work may be surprised to have assignments on a topic before we review that topic together, but this workflow is typical for adult learners: our time together will be more productive if you are prepared with a first look at new
ideas and definitions, and can come to class with questions and next steps in mind.

Our online textbook groups its exercises into two groups, "Participation activities" and "Challenge activities." Most preparatory exercises are these participation activities from the book. There will also be a small number of online quizzes considered to be preparatory exercises. Generally speaking, preparatory exercises examine your mastery of the course's technical material at the level of definition recall and simple application. Most preparatory exercises will be due at 8AM the morning before each lecture, with a handful of others through the semester.

Supplementary exercises are problems which you should work after a lecture on a topic, but before undertaking full programs based on that topic. They build on the material of the lectures; most supplementary exercises are the "Challenge activities" of the online book.

There will be two laboratories most weeks of the semester. The labs assess the learning outcomes associated with implementing Java software. In particular, the labs emphasize each of the individual steps in our software construction discipline, and there will almost always be several deadlines each lab corresponding to these steps. Some labs, usually the first lab after each major new topic, will be demonstration labs where I work most of an exercise along with you, and you are able to see my screen as a window on your lab computer. In later labs on each topic, you will work more independently.

There will be a number of larger programming projects over the course of the semester; I expect there to be four projects, but there may be as few as three or as many as six. Like laboratories, the projects assess your progress on the learning outcomes associated with implementing Java software, but the work will be larger in scope (requiring several weeks rather than one class period), and implemented more independently. In projects you will again be expected to follow the steps of our software construction discipline, and on earlier projects there will be multiple deadlines which serve both to verify your adherence to the process, and to provide you with early feedback.

Partial credit for programs, whether on examinations or projects, may be awarded only for programs whose design is documented via comments. Although we will not have larger-scale writing exercises, I do require and will assess program comments on both the laboratories and larger projects. Moreover, examinations will include short-answer questions, and unclear writing may lead to loss of some or all credit.

Finally, there will be four examinations, three midterms plus a final. The examinations focus on all learning outcomes. Most questions will be of one of two forms: given a problem, work towards a program; or given a program, understand different aspects of it. The midterm exams are not
explicitly cumulative, and cover distinct topics, tentatively:

1. The imperative core
2. Arrays
3. Objects and classes

A definitive list of the book sections covered by each examination will be announced with each midterm date. Although their questions will explicitly focus on different topics, the cumulative nature of this class means that the skills from earlier exams will absolutely be required on later ones. The final examination is cumulative over the entire semester.

**On formative and summative assessments**

Programming work due before the last two weeks of class, as well as the first two midterm exams, are all considered *formative assessments*, whose purpose includes giving feedback to you and shaping your learning. You will receive feedback on all formative assessments, and are expected to use that feedback to improve your future performance. Feedback on work submitted to the electronic textbook is delivered via that book to guide you to a correct solution. Feedback on lab work is delivered in person in the lab session; it is your responsibility to ask questions during labs when you are unable to keep up with or complete the assignments.

Other work (including the final project and the final) is considered *summative*, intended not for feedback or as learning tools but only as measurements of skill. You will not receive detailed feedback on summative assessments. (The third midterm will treated as formative if time allows, but may be deemed summative.)

**Evaluating assessments**

Labs will be graded qualitatively on progress demonstrated in the submitted work product towards mastering the skills exercised in the lab. Most labs will be marked with one of the following assessments, whose incorporation into the calculation of the final grade is as given:

**Expected progress exceeded (star)** Full credit

**Expected progress shown (check)** Low A

**Some progress shown but improvement needed (dash)** Mid-BC

**Not completed, or no progress shown (cross)** No credit

These lab grades relative to letter grade boundaries will be calculated based on final scaled letter grade cutoffs.

For calculating the final grade, we divide the assignments and examinations into the following **phases**: preparatory assignments (which includes quizzes), supplementary assignments, labs, early projects, later projects, mid-term exams and the final exam (in that order). Each marked item will be also attributed to one **category** including: design and programming (which includes book work, most labs, projects and most other independent work, and quizzes), GUI programming (in some of the last few labs), algorithm
design, describing how code will execute, conveying understanding of concepts. The latter three categories are for exams; the first and second categories are for out-of-class homework and projects. The points of items for a particular category in each phase of the class will be adjusted to be no less than the weighted average of items of the same category in the next later phase of the class. So for example, your percentage score on a midterm exam asking you to predict the effect of some piece of code will not be less than the weighted average of your scores on the final exam questions asking you to predict the effect of code. This adjustment will be transitive from the final exam backwards.

Your grade for each class assignment and phase will be calculated as a weighted average. In turn your final grade will be the weighted average of the assessment of your work, adjusted as described above, and weighted as follows:

**Preparatory assignments** 5%
**Supplementary assignments** 10%
**Labs** 10%
**Projects** 20%
**Midterm examinations** 30% (10% each)
**Final examination** 25%
**Participation and professionalism** 5%

I will convert a weighted average percentage $g$ to a letter grade no more strictly than as follows:

<table>
<thead>
<tr>
<th>$g$</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>$95.0 \leq g$</td>
<td>A</td>
</tr>
<tr>
<td>$92.0 \leq g &lt; 95.0$</td>
<td>AB</td>
</tr>
<tr>
<td>$86.0 \leq g &lt; 92.0$</td>
<td>B</td>
</tr>
<tr>
<td>$82.0 \leq g &lt; 86.0$</td>
<td>BC</td>
</tr>
<tr>
<td>$73.0 \leq g &lt; 82.0$</td>
<td>C</td>
</tr>
<tr>
<td>$60.0 \leq g &lt; 73.0$</td>
<td>D</td>
</tr>
</tbody>
</table>

Grades below 60% are non-passing grades. In addition, to get a final grade above C, you must pass the final exam. The university uses annotated F grades for cases of failure with cessation of class activity and attendance; where such grades are appropriate I will draw them from both the assignment results and attendance records.

**Procedures and policies**

**Textbook**

Our textbook is the online *Programming in Java* text from zyBooks, which is customized for this class section. You should subscribe to this book, which will give you electronic access into January, and the ability to make a print copy as well if you like. You should consider this textbook to be required for this class. Much of your assessed work will be done through the book's interface. To subscribe to this textbook:

1. Sign up at [zyBooks.com](zyBooks.com)
2. Enter zyBook code *UWLAXCS120MaraistFall2018*
3. Click *Subscribe*
The course information above lists an additional reference which you might find useful as a source of alternate explanations and additional exercises. A third book which I’ve found useful for ideas for exercises is *Introduction to Programming with Java: A Problem Solving Approach*, John Dean and Raymond Dean, McGraw Hill, second edition.

**Email and web page**

The course website will be the primary means of communication information across the whole class; electronic mail will be our primary means of personal communication.

*Course Website.* The main web page for this class is listed at the beginning of this document. All class announcements will be posted to that page, and you are responsible for checking it regularly. That page also includes an RSS feed for updates. There are several services which will provide email updates from RSS feeds which you can find by a web search; if you choose to use one, pay attention to how often they check the feed and send email.

*Email.* I will expect you to check your email regularly, and to read and understand messages relevant to this class. In particular, my feedback on your work will be delivered by email. By default I will use your school email address which I receive as part of the information about you that the university gives me, but I am happy to also use a different email address if you email it to me from your school email address. It is your responsibility to make sure that I have an email address which you can and will access regularly, and which you check at least once per business day. Note that we will **not** use electronic mail for submitting assignments; see the *Submission and assessment of assignments* section below. My university email account is the only forum which I regularly check; you should not attempt to communicate with me for class business via other email addresses, or other forms of social media.

For assistance with email or other matters relating to university computer and network services, contact [ITS](https://www.uwlax.edu/its) by phone at 608/785-8774, in person on the first floor of Wing Technology Center, or by email to helpdesk@uwlax.edu.

In general, during the semester I will respond to emails within one business day. When you use email, make sure that you:

**Include your full name.** There’s a small army of you, and one of me. Make sure it’s easy for me to know who you are.

**Mention this class by name or number.** All of your instructors are almost certainly teaching more than one section.

**Write professionally.** Observe the forms of casual business writing, write in complete sentences, and use your spell-checker. Keep in mind that email to an instructor about a class is a different medium, and requires a different voice, than texts to a friend.

I have posted links to a number of guides to effective emailing on a [web]
I expect you to attend class. Our class meetings will be the only source for some class material and assessments. There are no "makeups" for in-class participation opportunities. If you miss class, it is your responsibility to get notes from a classmate. We will not use class time, nor prioritize open-door hours and appointment times, to review topics missed due to nonattendance.

Final examination times are scheduled by the university; make sure to plan any end-of-semester travel around them. Should an exam need to be rescheduled according to the university's limit on the number of exams a student may take on the same day, you must give me notice as soon as you become aware of this situation. I will normally reschedule your exam to the first exam slot before our normal class slot in which you are not taking, and I am not giving, another exam; or similarly after, if our normal slot is on the first day of exams.

Admission of latecomers to an examination may be refused after any student completes the exam and leaves the exam room.

I do not expect there to be review sessions for this class outside of regular lecture/lab times.

Submission and assessment of assignments

Each assignment is to be submitted via the electronic submission system detailed in that assignment. I expect that we will primarily use D2L in this course for assignment submission, but you must always check each assignment for the correct procedure. We will not be using email for assignment submission; assignments emailed to me will not be considered validly or on-time submitted unless either the particular assignment specifically calls for email submission, or I have specifically instructed you to email me an assignment. Submissions for programming assignments should consist of fully-functional code which behave as specified in the assignment.

The deadlines for the different types of assignment are as follows (except where a particular assignment specifies otherwise):

- Exercises in the online textbook are to be completed within the textbook, and are due by 8:00am of the deadline day.
- Projects, exercises, and preparation/follow-up work for labs specified on the course site or in slides/notes are due by 8:00am of the deadline day.
- Most lab work is to be completed in the lab class for which it is assigned, and stages of this work will be due at various points during that lab.

My assessment of your coursework will be returned in compliance with FERPA regulations, either directly to you or via email. As described under Email above, I will email you either at your official university email (which only you are authorized to access), or to an alternate email address which...
you designate. In this way only you will have access to your grades unless you take specific action otherwise.

After you have completed the course, copies or records of your graded material that I retain will be accessible up to seven weeks into the next academic term (either Spring after Fall or J-term; or Fall after Spring or Summer).

I plan to provide feedback on formative assessments submitted on-time within 21 days of the final deadline for that assessment, and to notify you when circumstances require delay.

**Assignments submitted late**

No credit will be awarded for homework from the textbook or lab work completed late. Credit for late submissions will be awarded only for late projects:

- If submitted within 24 hours after the time at which it was due, a reduction of 10% of the awarded percentage score will be assessed.
- If submitted over 24 hours but within 48 hours of the due time, the reduction will be 25%.
- If submitted over 48 hours but within 72 hours, the reduction will be 50%.
- No credit is given for assignments submitted after that point.

See the *Accommodations for individual circumstances* section below for extenuating circumstances that impact your ability to meet deadlines or participate in class activities.

**Equity of course execution**

This course will be delivered and assessed fairly, in the specific sense that all students in this section will have equivalent opportunities to demonstrate their mastery of the subject, and will be assessed according to the same criteria. The only assessed work and the only criteria for assessing that work, and thus for the grades derived from it, will be as set forth in this syllabus.

Mindfully attempting to be assessed by more lenient criteria than one’s colleagues, or by criteria other than the work for and conduct in this class as described in this syllabus, is unprofessional and will be considered a form of academic misconduct.

**Errors and regrading**

If you find an error in the evaluation of your work, you have the right to ask for it to be regraded.

- All requests for regrading must be by email.
- All requests for regrading must detail specifically where the suspected error was made, and what the error is.
- All requests for regrading should be made no sooner than 24 hours, but within one week, of the evaluation of the work being returned to you. If the assessment of some piece of work is returned in stages, the deadline for requesting a regrade will be within a calendar week of when the
suspected error is first returned to you.

- To ensure that a uniform standard is applied across the class, all regrading will use the same criteria and rubric applied to everyone else.
- In general, an entire assignment or exam may be regraded in response to a regrading request, even if your request addressed only a proper subset of the original. So make sure that errors to your detriment outweigh errors in your favor.

You will always be notified of errors I find in the evaluation of your work after it is returned to you, as well as any resulting change to your grade, even if you did not request a regrade.

Collaboration

I encourage you to work together to understand course material. Learning together is a great way to learn and share ideas, and is a useful professional skill. However, in order to actually learn something, it is important that you complete the real work of programming on your own. It is acceptable to:

- Discuss the general approach to an assessed problem with each other.
- Discuss and solve other, unassessed problems together.
- Work together to install software we’ll use, or get it to work properly on individual computers.
- Help each other figure out syntax errors when code isn’t compiling.
- Help each other isolate and debug problem spots when code isn’t running correctly.

However:

- It is not OK to write code together, or to copy code from anyone inside or outside of the class.
- It is not OK to simply copy code, whether from online, a book or printed article, other people, or any other source. You can use online references to get additional explanations of how Java works, or to learn programming techniques. But the only way to actually gain the skill of programming is to write code yourself.

Any improper behavior with respect to these guidelines will be dealt with as academic misconduct according to University policy.

Academic integrity and acceptable use policies

Academic misconduct is a violation of the UWL Student Honor Code and is unacceptable. I expect you to submit your own original work and participate in the course with integrity and high standards of academic honesty. When appropriate, cite original sources. Plagiarism or cheating in any form may result in a diminished grade or failure of the assignment or of the entire course, and may include harsher sanctions. As necessary I will use resources provided by the university or other services to verify the originality of submitted work. Refer to the Student Handbook for a detailed definition of academic misconduct.

In general,
• You can share ideas, but you may never share code.
• You must independently write all of the code you submit and never copy code from anyone inside or outside of the course to complete an assignment.
• You are expected to be able to fully explain every line of Java code that you write, and may be asked to do so for any given assignment.

In interpreting these general guidelines, "you" should be taken to mean the unit designated to complete one assignment. Except where explicitly stated otherwise in an assignment, all assignments are individual assignments, and it is individuals who may not collaborate on code. Where an assignment is explicitly deemed to be a group assignment, the individuals within a group may freely share material with each other, but never with individuals in other groups.

The article ‘Avoiding Plagiarism’ on the Murphy Library website offers helpful information on avoiding plagiarism. You may also visit the Office of Student Life if you have questions about plagiarism or cheating incidents. Failure to understand what constitutes plagiarism or cheating is not a valid excuse for engaging in academic misconduct. Acadia University offers a light-hearted ten-minute interactive tutorial on avoiding plagiarism at library.acadiau.ca/sites/default/files/library/tutorials/plagiarism

UWL and UWS policy also mandates responsible use of shared computing resources. In particular, your authorization for the use of administrative server resources such as course management systems (like D2L or Canvas), program submission and autoevaluation systems (like AutoLab or WebCat), the course web site, or other assigned systems is strictly limited to the purpose described in course assignments and other material. Any disruption, exploration and/or exfiltration of system components is strictly prohibited, and may also constitute academic misconduct. More information about the UWS policy on Acceptable Use of Information Technology Resources is available at www.wisconsin.edu/regents/policies/acceptable-use-of-information-technology-resources

Professional conduct

Interacting with peers and with me in a constructive, respectful and professional manner, being a constructive and supportive presence in class, handling difficulties with grace and resilience, operating as an autonomous and responsible adult, fulfilling commitments, and approaching work with enthusiasm are all valuable professional (and life) skills, and are firm expectations of this class. Part of your final grade in this class will be determined by the quality and consistency of your professional conduct, whether online, in class, or in open-door office hours/appointments.

One aspect of being a constructive and supportive presence in class is simply not being disruptive to the class. Attendance carries the obligation of being a constructive presence, or at least, a non-disruptive presence. In
particular:

- Cell phones and other electronics must be silenced for the duration of class. Consider using an app like Shush! or Silent Time (for Android), or AutoSilent (for iPhone) to manage silencing your devices automatically.
- If you need to arrive to class late or leave early, be mindful of creating a minimum of disruption: sit near the exit and on the end of the aisle, pack lightly, and avoid using materials in class which are noisy on packing/unpacking.
- Research has shown that screen use in class is distracting not only to the student using a device, but also to that student’s neighbors. So if you plan to use a screened device in class, I’ll expect you to sit in the back row so that your screen distracts the fewest people. Likewise, if you plan not to use a screened device, you should sit away from the rearmost rows.

In cases of egregious, repeated or persistent disruptive conduct, of mindful discourtesy or of any intimidation of anyone in class, or of isolating or shaming conduct based on gender, race or other identity issues, I may require you to leave class immediately, possibly on an ongoing basis.

Findings of academic misconduct and/or unacceptable use of course resources may also result in loss of graded credit for professional conduct. In particular academic misconduct on a project, major assignment or any examination, as well as multiple instances academic misconduct and/or unacceptable use of course resources, will result in the loss of most if not all credit for professional conduct.

In laboratory sessions,

- Be gentle with lab computers.
- Speak in quiet tones in the lab to avoid disturbing others.
- It is permissible to assist neighboring colleagues with debugging when they are stuck on a particular problem. However:
  - You may not copy any aspect of your lab work from a colleague, nor provide your work as to them for duplication.
  - You may not interrupt colleagues, who have their own work to do, to repeatedly ask for help; raise your hand and I will answer your question as soon as I can. Moreover you should remain at your own computer unless you are leaving the lab; moving about the lab for conversations is disruptive to others.
- Use of headphones in lab is unprofessional and strongly discouraged. I will frequently announce important material, and cannot repeat it individually simply because you excluded yourself from listening with the group.
- Do not touch computer screens; use the mouse when indicating particular items to me or to a colleague.
- Food is not allowed in the lab. Drinks in closed containers are permitted but may not be placed on the same desk as a computer or keyboard — keep them on the floor, where a spill will not destroy equipment.
- We recommend that you use hand sanitizer when leaving the lab; key-
boards and mice are notorious vectors for communicable disease.

**Concerns or complaints**

If you have a concern or a complaint about either the course or me, I encourage you to bring it to my attention. My hope would be that by communicating your concern we would be able to come to a resolution. If you are uncomfortable speaking with me, or if you feel your concern has not been resolved after bringing it to my attention, you can contact my department chair or the Office of Student Life.

The Student Academic Non-Grade Appeals process can be found in the Student Handbook. Information about appeals and petitions for academic matters is in the UWL Catalog.

I normally give anonymized examinations: you will sit at a desk tagged with your name; rather than writing your name on the exam, you will write the number on that tag. The anonymity allows us all to be more confident in the accuracy and uniformity of assessment across the class. However, that anonymity extends only through the completion of assessing the individual exam questions. After marking I will de-anonymize the exam papers to understand both individual and group trends and weaknesses, and to address them through subsequent improvements to the class. So exam papers should not be considered an anonymous forum for suggestions or complaints.

**Sexual harassment**

As an employee of the University of Wisconsin-La Crosse, I am a mandated reporter of sexual harassment and sexual violence (which include sexual assault, domestic violence and stalking) that either takes place on campus or otherwise affects the campus community.

So if I receive detailed or specific information about an incident such as the date, time, location, or identity of the people involved, I am obligated to share this with UWL's Title IX Coordinator in order to enable the university to take appropriate action to ensure the safety and rights of all involved. It does not matter whether the incident took place on- or off-campus; it matters only that a person who is a member of this campus was involved in the incident.

It is possible that course assignments may lend themselves to disclosure, but you should not share any details of an incident with me until you have discussed your options under the new Title IX guidelines. There are confidential reporters available to students at UWL where you can have this discussion.

For students not wishing to make an official report, there are confidential resources available to provide support and discuss the available options. The contact in Student Life is Ingrid Peterson, Violence Prevention Specialist, 608/785-8062, ipeterson@uwlaus.edu. For more resources or to file a
report, please see www.uwlax.edu/violence-prevention.

I am also happy to help you find counseling and support services, or a confidential reporter — just ask.

**Class interruptions and cancellations**

In the event of a campus incident that impacts the availability of teaching spaces, any changes or cancellations will be communicated to you via your university email account. Depending on the incident, some or all of the information might be posted on the UWL home page.

In the event of inclement weather, we will follow the University’s closure policy. If classes are not canceled, I will make every effort to be in class on time, and so should you. Please do not send me email asking whether class is going to meet; instead, check the university website. The university’s emergency readiness plan is available online, that page also describes sign-ups for individual emergency alerts. In the event of a cancellation, consult the course homepage for any alternative assignments or other arrangements.

**Accommodations for individual circumstances**

It is my goal that all students have equivalent opportunities to succeed in this class. This section discusses the general procedures for alternative assessment accommodations in this class, as well as a number of specific situations for which there are standard mechanisms and policies in place to achieve the goal via accommodations for individual circumstances.

**General procedures and constraints.** Students may propose alternative assessments for assignments and exams for matters outside of a student’s control such as documented non-chronic illness, bereavement, unplanned university equipment unavailability, or university program travel or activities.

- In almost all cases, you will work with a campus office (usually one of the ACCESS Center, Veterans Services Office, or Office of Student Life) to design and manage your accommodations. They will have confidential access to the full details of your situation, and so they will be the sole authority who can certify that the accommodations you propose are both necessary and sufficient for your situation. Moreover all accommodations shall be reviewed by the same office: the necessity and sufficiency of the overall accommodation for your situation cannot be accurately assessed otherwise.
- Any accommodation must also ensure that the required objectives for this course are assessed as thoroughly as under standard procedures. It is my role to judge whether any proposed accommodation meets this requirement.
- It is your responsibility to propose assessment alternatives which which are both approved by the overseeing campus office as necessary and sufficient to accommodate your circumstances, and approved by me as appropriate for the original assessment’s objectives.
• All requests for accommodation must be accompanied by appropriate supporting documentation. In most cases this documentation will be reviewed by a separate group on campus such as the ACCESS Center or Veterans Services Office, and I will not see specific details. Where no such campus group applies, the specific form of documentation will be at my discretion.
• Proposals for alternative assessment must be made at least ten calendar days before any relevant major deadline or exam. If a proposal cannot be made in time due to medical or other emergency, the proposal should be made at the earliest possible point.
• Alternative assessment proposals should address relevant big-picture issues in addition to immediate course matters.
• Alternative assessment proposals must be explicit, and must be sent only by email or in writing.
• Students proposing alternative assessments should never simply assume that their proposal will be granted verbatim, and must allow time for thoughtful review of all proposals.
• Extracurricular and student groups/activities, planned personal trips, and similar elective activities are not considered to be outside of a student’s control, and do not qualify for alternative assessment.
• Accommodations are generally not available for the activities of other classes. Do not schedule activities for other classes during the lecture/lab/exam times of this class; you are not "free" at those times.
• Accommodations should enable you to complete the assessments for this class during the regular semester. I will avoid recording incomplete grades as part of an alternative assessment plan for any situation which has previously been addressed by accommodation, whether at UWL or other institution, whether via the ACCESS Center or not. Incomplete grades will also not be used where an advisor’s or other credible recommendation for a reduced load, for a particular semester or on an ongoing basis, was disregarded or avoided; you are expected to design a feasible schedule with your advisor.

Disabilities and medical conditions. Accommodations for documented disabilities or medical conditions are made via The ACCESS Center. Contact them to meet with an advisor who will register documentation of your situation, and help you to develop and propose alternative assessments.
• Examples of the disabilities and conditions for which this procedure applies include, but are not limited to: ADHD; autism spectrum disorder; acquired brain injury; PTSD; and physical, sensory, psychological, or learning disabilities.
• The ACCESS Center is located at 165 Murphy Library, and is reachable by phone at 608/785-6900 and by email at ACCESSCenter@uwlaus.edu. Interactions with the ACCESS Center and with instructors should be initiated promptly. For issues and conditions identified prior to the semester, you should contact the ACCESS Center prior to the semester in order to
propose and confirm an accommodation plan before assignments are due. For issues arising during the semester, you should contact the ACCESS center to initiate their accommodations process promptly after a diagnosis. Accommodations will not be applied retroactively in the case of a delay in initiating the ACCESS Center process. Once some alternative assessment accommodation is arranged for you via the ACCESS Center in this class, any other accommodations for you as well as any changes or extensions to your accommodations, including those arising from changes in your underlying condition or disability, must also be arranged via ACCESS Center procedures (see Changes to accommodations below), and must follow the procedures described elsewhere in this syllabus.

You can find out more about services available to you with disabilities at The ACCESS Center website, www.uwlax.edu/access-center.

Veterans and active military personnel. Veterans and active military personnel with special circumstances (e.g., upcoming deployments, drill requirements, disabilities) are welcome and encouraged to discuss these issues with me, and I expect you to do so as far in advance as possible.

For additional information and assistance, contact the Veterans Services Office, www.uwlax.edu/veteran-services. Students who need to withdraw from class or from the university due to military orders should familiarize themselves with the university’s current military duty withdrawal policy, catalog.uwlax.edu/undergraduate/academicpolicies/withdrawal.

Religious accommodations. Per the UWL Undergraduate and Graduate Catalogs, "any student with a conflict between an academic requirement and any religious observance must be given an alternative means of meeting the academic requirement. The student must notify the instructor within the first three weeks of class of specific days/dates for which the student will request an accommodation. Instructors may schedule a make-up examination or other academic requirement before or after the regularly scheduled examination or other academic requirement."

University athletics. Student athletes are expected to submit their full schedule for the semester, including expected travel times and possible championship tournaments, by the end of the first week of class. I realize that your coaches’ official letter may not be ready by that time: that letter can come later. But you are able and expected to collect and convey the information yourself, and later follow up with the official documentation.

In the event of cancellations or postponed events, I expect you to inform me in email before our next class meeting of the cancellation. In that email, you should also indicate to the best of your knowledge whether the university is attempting to reschedule the event later in the semester.

Changes to accommodations. Accommodations can change by mutual consent to reflect changed circumstances. Changes should follow the same review and implementation mechanism as the original accommodation; in
particular where the ACCESS Center reviewed and recommended original accommodations, I will expect changes or parallel accommodations to be reviewed and recommended through the ACCESS Center.

**Further university policies**

UWL provides a list of policies common across all classes at [https://www.uwlax.edu/info/syllabus](https://www.uwlax.edu/info/syllabus)
How to do well in the course

You want to do well in this course; I want you to do well in this course. Some simple ways to help make that happen are:

**Time management.** I cannot stress enough that programming courses are notoriously more time consuming than most other courses. You should expect the supplementary assignments, before- and after-lab work, and projects to take a significant amount of time to complete. Start assignments early to allow enough time to reflect on and wrestle with problems you encounter. Make sure you are allocating a sufficient amount of time each week outside of lecture to work on assignments and reviewing concepts.

**Do the reading before you come to class.** Aside from the fact that reading and preparatory exercises feed a substantial portion of your final grade, they are essential to getting the most out of class. Arriving prepared allows you bring your own learning goals to class, gives you early warning of how difficult you will find the day’s material, and makes your class time more productive. If you wait until it is time to do a programming assignment to look at the material, it may be too late to help you.

**Stay current with the material.** All of the concepts in this course build on one another. Falling behind early in the course will cause problems understanding and succeeding with the material throughout the remainder of the course. Attending lecture, reviewing course materials each week, and staying current with the assignments will help you stay on track. If you are having trouble with a particular concept then you need to ask questions right away.

**Ask and answer questions in class.** It’s easy to listen to others talk about a subject — so easy that you can convince yourself that you are more fluent in the subject than you really are. Participation is how you find out the limits of your understanding, and helps you build your plan for succeeding with the subject.

**Get help in person.** Get in touch with me when you have questions. Email is a perfectly reasonable way to contact me, but often a topic or question is better answered in person, especially when a longer exchange of several emails seems to drag on. Don’t hesitate to come to open-door hours or to schedule an appointment when this is the case — it’s best to clear up points of confusion quickly, before misunderstanding builds up.

There are other resources on campus besides me where you can seek help: The CS Department offers both informal peer support through the CODERS group (look for bulletin board announcements for "Cookies with CODERS"), and tutoring sessions by your upper-level colleagues several times a week. And UWL offers academic skill building assistance. All of these services are available at no additional cost to you (your fees and tuition have already paid for all of these services).

**Study with others.** Our class policies are clear that the work you submit must ultimately be your own. But you have leeway on collaborating on gen-
eral class work for understanding the material and the assignment state-
ments, for debugging programs which you’ve written, and for working other,
unassessed practice problems. Take advantage of the opportunity to learn
with your colleagues. Study groups will become ever more important as
you move through your degree, and the ability to learn together will serve
you well throughout your career.