

University of Montana

## ScholarWorks at University of Montana

---

University of Montana Course Syllabi, 2021-2025

---

Spring 2-1-2022

### CSCI 152.00: Interdisciplinary Computer Science II

Jesse V. Johnson

*University of Montana, Missoula*, [jesse.johnson@umontana.edu](mailto:jesse.johnson@umontana.edu)

Follow this and additional works at: <https://scholarworks.umt.edu/syllabi2021-2025>

**Let us know how access to this document benefits you.**

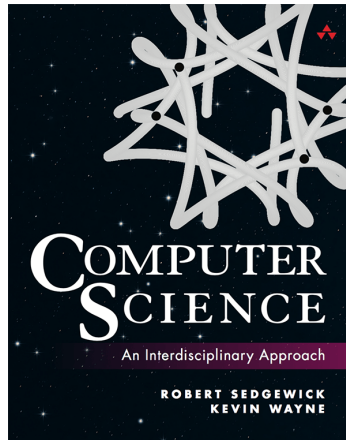
---

#### Recommended Citation

This syllabus is used for all sections of CSCI 152.

This Syllabus is brought to you for free and open access by ScholarWorks at University of Montana. It has been accepted for inclusion in University of Montana Course Syllabi, 2021-2025 by an authorized administrator of ScholarWorks at University of Montana. For more information, please contact [scholarworks@mso.umt.edu](mailto:scholarworks@mso.umt.edu).

# CSCI 152: *Interdisciplinary Computer Science II* *Spring 2022 Syllabus*



---

This class will transition you from a programmer to a problem solver. You will use data structures to store information and construct algorithms to process data. As is the case with the previous courses, we emphasize learning through the practice of programming. You will complete a sequence of assignments using the strongly typed programming language, Java. The use of Java will introduce many new concepts not taught in previous courses. Additionally, in this course you will learn some of the ideas that ignited the digital age. Ideas from thinkers like Turing, von Neumann and Shannon. Finally, we'll look at the essential components of a modern computer and experiment with how these simple components complete complex computing tasks.

---

## **Administrative**

*Instructor:* Jesse Johnson | [jesse.johnson@umontana.edu](mailto:jesse.johnson@umontana.edu) | SS417

*Office hours:* Monday 1-3pm and Wednesday 2-3pm *or by appointment*

*Teaching Assistant:* Anthony Marozzi | [anthony.marozzi@umconnect.umt.edu](mailto:anthony.marozzi@umconnect.umt.edu) | ISB 406A

*Office hours:* Tuesday and Thursday 1-3pm; *or by appointment*

*Prerequisites:* CSCI 151 or CSCI 135. Prerequisites will be waived only for students that can demonstrate clear mastery of CSCI151.

*Textbooks:* [Computer Science, An Interdisciplinary Approach by Robery Sedgewick and Kevin Wayne](#)

*Website:* In addition to the [Moodle](#), the text for the course has a [well developed website](#), and a pair of Coursera courses: [Computer Science: Programming with a Purpose](#) and [Computer Science: Algorithms, Theory, and Machines](#).

*Software:* We will use the [IntelliJ](#) platform for development. Student licenses are available [here](#).

---

## **Meeting Times and Places**

*Lecture:* Wednesdays | 11:00-11:50 | Social Science 254 | *Attendance required*

*Labs:* Monday, Tuesday, Thursday | 11:00-11:50 | Social Science 344 | *Must attend 2 per week*

*Final Exam:* Friday May 13 | 10:10-12:10 | Social Science

---

## **Learning Outcomes**

Upon successful completion of this class, students should be able to:

1. Make informed choices about the appropriate data type for a particular computing task.
  2. Apply the scientific method to determine the cost of computing.
  3. Recognize the need for and an ability to engage in continuing professional development.
  4. Use current techniques, skills, and tools necessary for computing practice.
- 

## **Topics Covered**

1. Basic Programming Concepts
  2. Conditionals and Loops
  3. Arrays
  4. Input and Output
  5. Functions and Libraries
  6. Recursion
  7. Performance
  8. Abstract Data Types
  9. Creating Data Types
  10. Programming Languages
  11. Sorting and Searching
  12. Stacks and Queues
  13. Symbol Tables
-

## Evaluation & Grading

Your grade for the course will be determined by the following elements.

### *Attendance: (20%)*

Students are expected to attend all lecture periods and two lab periods per week. Attendance will be taken and will count towards your final grade. If you are late for class, attendance points will be deducted.

### *Quizzes: (10%)*

Each unit for the Coursera material includes a quiz. We will do these before each lecture as a pre-test. After the lecture, students are responsible for completing a quiz and turning in a screen shot of the grade received. Students can do these as many times as they like, but note the software only allows three attempts per 8 hour period. There will be about 13 of these.

### *Programming Assignments: (30%)*

Each unit for the first ten units of Coursera material includes a programming assignment. Students will submit these on time and get a screenshot of the online assessment. Those will be uploaded to Moodle and the grade recorded. On several occasions, there will be unique assignments that are formulated by the instructors. There will be about 11 programming assignments.

### *Midterm examinations: (20%)*

Two examinations will be given during class meeting times. They will directly cover the material in the book and in the assignments. They will be timed. They will be spaced approximately equally, through the semester. Only the material since the previous midterm will be covered - they are not comprehensive.

### *Final Examination: (20%)*

A comprehensive final examination will cover all material introduced in the class. It will be timed.

---

## Success

Based on twenty years of teaching computer science at UM, here are some tips on how to succeed in CSCI 152:

1. **Put in the work:** Plan on 2 hours outside of class for every hour in class. There is nothing easy about computer science, it takes everyone time to learn the concepts. Put in the time, and use the time wisely.
2. **Don't cheat:** Your instructors are keenly aware of and monitoring sites like [chegg](#), [Geeks for Geeks](#), and [github](#). We will be looking for work that isn't yours when we grade your assignments. In addition to being academically dishonest, you don't learn when you cheat. Do your own work. There's nothing wrong with *looking* at other implementations, but make sure the one you turn in is your own, and *acknowledge* where you got inspiration from.
3. **Show up:** It's 20% of the grade, and both the instructor and the TA are committed to making our time together worthwhile. Nearly 100% of students that fail have stopped coming to class. A student that comes to class for the entire semester very, very rarely fails.
4. **Be social:** Get together with other students. Talk through solutions. Learn what your fellow students know. Teach them what they don't. You learn by explaining. Use the student lounge.
5. **Use office hours:** They are for you. Don't be shy. Don't think your questions are silly or unimportant. I almost always get to know the best students through office hours and other means.
6. **Study by doing:** Don't stare at the book or memorize terms. Do extra programming problems. Study the portions of the book that help you solve problems. That's the best way to learn.
7. **Enjoy what you do:** Embrace the problem solver in you. Relish in figuring things out. Be motivated to learn

more. Be curious about how things work. Ask a lot of questions.

8. **Know the rules:** You can withdraw from a class without a “W” until 7 February. You have until March 30 to withdraw without a “WP” or “WF”. Know when you are in over your head, and get out without damaging your GPA.
- 

## Academic Dishonesty

All students must practice academic honesty. Academic misconduct is subject to an academic penalty by the course instructor and/or a disciplinary sanction by the University. All students need to be familiar with the Student Conduct Code. The Code is available for review online at <http://www.umt.edu/student-affairs/community-standards/default.php>

The nature of programming sometimes muddies the distinctions between your work and someone else's. Let me be as clear as I can.

- Every line of code should be written by you and you alone.
- Often, you'll code that serves a similar or identical purpose. When this happens it's fine to:
  - Study it until you understand it.
  - Write your own version of it.
  - Include comments that tell the grader where you got the idea from.
- The same applies to code collaborated on. If you work with another student:
  - Each student's code should be unique.
  - Comments should indicate who you worked with.

If a programming assignment is found to be similar to other work, a grade of zero will be given for the first offense. Subsequent offenses will result in a failing grade in the class.

---

## Additional class policies and information:

- If you miss a class, you and you alone are responsible for the material covered. This includes handouts, schedule changes, and lecture notes.
- Our campus has its [COVID policies online](#).
- Be aware of important dates and [deadlines related to classes](#), especially regarding drop or withdrawing from courses.
- Also in the University catalog, review the policy on **incompletes**. In particular, note that incompletes can only be assigned when the student has “been in attendance and doing passing work up to three weeks before the end of the semester.” Incompletes will not be issued simply to prevent a failing grade.
- Students with disabilities will receive reasonable modifications in this course. Your responsibilities are to request them from me with sufficient advance notice, and to be prepared to provide verification of disability and its impact from Disability Services for Students. Please speak with me after class or during my office hours to discuss the details. For more information, visit the [Office for Disability Equity](#).