

CS 120 section 70 (hybrid), Fall 2020

Instructor: Michael Kowalczyk

Office: 2222 Jamrich Hall

Office Phone: 227-1600

Class Meetings: 12:00noon – 12:50pm Wed / Fri in room 2812 Jamrich Hall

Virtual “walk-in” Office Hours:

8:00am – 10:00am, Thurs / Fri at <https://nmu.zoom.us/my/drkow>

By-appointment Virtual Office Hours:

3:00pm – 5:00pm, Mon / Wed / Thurs / Fri

Reserve a time by going to Starfish in MyNMU (one hour block maximum).

Reserve your appointment by the night before (at latest).

Email: mkowalcz@nmu.edu

Course Website: <https://educat.nmu.edu>

How to succeed in this course:

- 1) Keeping pace with lab exercises is crucial: get at least one lab exercise completed each scheduled lab day. If you get behind, see the tutor or myself outside of class.
- 2) Each “Saturday programming assignment” should take an hour or less to complete, but sometimes just one bug can eat up hours of your time trying to figure it out. Therefore start a week before it’s due. That way if you get stuck you can have me explain it – and you don’t waste a big chunk of your time.

Overview:

This course is an introduction to writing computer programs using Java. Although I assume you have never used a programming language before, you will probably find the course challenging and interesting even if you have.

Prerequisites:

Mathematics Placement recommendation of CS120 or higher, or B- or better in either MA100, CS101, or CIS110.

Textbook:

None required, but I have listed some useful E-books and other resources in Educat.

Equipment:

You will need a laptop computer with a web browser and Internet access. You will also need to do some software installs (see “Software installations and your first program” and “Homework 0” in Educat for full details). If for some reason you don’t have a laptop computer, talk to me as soon as you can, since we will be using them for in-class exercises.

Grading:

Grades will be based on exams, homework assignments, labs, and quizzes. Homework assignments are weighted based on their size and complexity.

30% Homework

10% Labs

10% Quizzes

25% Midterm exam

25% Final exam

Handing in Programs and Late Policy:

Some assignments (programming assignments and quizzes) are handed in electronically while other items (labs, worksheets, and exams) are submitted in person. I urge you to hand in programming assignments 3 days before the deadline (I have Educat set up so that you can revise and resubmit as often as you like until it is due). Programming assignment deadlines are strictly enforced: **one second late is still late and late work gets no credit**. It is your responsibility to pace yourself accordingly. If for some reason you are having trouble handing something in, you can email it to me as an attachment, but I must receive it **before** the deadline. The latest you can submit a lab for credit is on the next lab day (e.g. the last day to get credit on Lab Exercise 4 is on Lab Day 5).

If some circumstance such as illness occurs which might interfere with getting your work in on time, let me know about your concern right away. I can usually work something out if your concern is brought up well in advance of the deadline.

Exam Dates & Schedule Conflicts:

The midterm and final exams are administered *on paper only*; no book, no computer, no notes. The midterm exam will be during our regular class meeting on Friday, October 2. The final exam will be held over two days: Wednesday, November 18 and Friday, November 20. Any conflicts with the exams (due to religious observances, other coursework, intercollegiate athletics, etc) must be made known to me as soon as you are aware of the conflict.

Laptop Use:

Most class meetings will be labs, in which your laptop is required. Other times, we will have lecture or discussion and I will need your complete attention (laptops closed).

You are responsible for keeping your laptop in good working condition and making frequent backups of your work. Note that the helpdesk does not backup your work if they need to fix your laptop (unless you want to pay them a fee), so make frequent backups to hardware external to your laptop *before* a crisis strikes.

Academic Conduct:

I work hard, with honesty and integrity; I expect my students to do the same. Every assignment must be written entirely by you. There are precisely two instances where including program source code from elsewhere is acceptable:

- You may include any code that I give out in my lab tutorials and lecture notes, without citation.
- Any other code that you didn't author **must** be accompanied with a full citation (this includes people, websites, books, etc.). Indicate clearly which lines of code you didn't write, and where it came from (in enough detail that I can find the resource myself).

The best way to help others succeed in the course is by explaining concepts and working through examples – not by giving away source code.

Course objectives:

CS 120 is an introductory programming course. It forms the foundation for later CS courses, but it also satisfies the Quantitative Reasoning Requirement. Upon successful completion of this course, a student should be able to do the following in the Java programming language:

- Solve programming problems through the use of conditionals, loops, and nested control structures
- Write an instantiable class from scratch
- Write code to call constructors and invoke methods on existing objects, including correct use of parameters and return values
- Demonstrate an understanding of commonly used operators (logical, arithmetic, and comparison)
- Demonstrate a basic working knowledge of arrays and their syntax

Evaluation of these learning outcomes will be done through written assessments (quizzes and/or exams).

Quantitative Reasoning Requirement (for students under the 2017-2018 bulletin or later):

This course satisfies the Quantitative Reasoning component of your general education requirements.

Formal Communication Studies Requirement (for students under the 2016-2017 bulletin or earlier):

This course satisfies the Formal Communication Studies requirement. This course is designed to introduce students to the ways in which information and ideas are expressed using a communication system other than English. Such courses should foster the student's ability to conceptualize and communicate in an orderly, rational manner. Characteristics of a communication system include: 1) possession of a grammar; 2) operation from an established set of rules; 3) reasoning properties such as deduction, inference drawing and problem solving. This includes courses in languages and those in which the central focus of the course is on statistics, computers or formal logic.

Disability Services:

If you have a need for disability-related accommodations or services, please inform the Coordinator of Disability Services in the Dean of Students Office at 2001 C. B. Hedgecock Building (227-1700). Reasonable and effective accommodations and services will be provided to students if requests are made in a timely manner, with appropriate documentation, in accordance with federal, state, and University guidelines.