

CS 5001: Intensive Foundations of Computer Science and CS 5003 Recitation

Fall 2020

with

Dr. Maria Jump

Email:	m.jump@northeastern.edu
Phone:	704.228.1392
Virtual:	Microsoft Teams
Zoom:	https://northeastern.zoom.us/my/profjump

1 -- Course Description

This course introduces students to the principles of systematic problem solving through programming. Topics include analysis of problems, modeling a solution, data types, and control structures. Additionally, it will introduce various ways to organize data including a discussion of their advantages and disadvantages.

1.1 -- Course Objectives

- Develop a basic understanding of how to analyze and break down large problems in order to implement efficient solutions using the Python programming language.
- Determine functionality of code written by oneself and others through reading and tracing short segments of code.
- Write correct and clearly-documented small-to-medium sized programs that others can read, understand, and modify.
- Use generalization for data and functions to limit code duplication.
- Develop tests to exercise implemented code and appreciate the importance of good testing in the software development process.
- Select appropriate data types to represent information including using common library classes.
- Assess the impact of data structure and algorithm choice on the running time and storage space needed to run a program.

1.2 -- Course Prerequisites

The course is suitable for students in the ALIGN MS in CS program. It assumes no previous programming experience.

2 -- Course Structure

Even though this course is being offered online, it is not completely self-paced. Students are expected to interact with modules on a weekly basis similar to a face-to-face course. Weekly materials will be released on Mondays as follows:

Week	Dates		Topics	Lab	Assignment
1	Sep 7	-- Sep 13	Introduction	Mod 1 Lab	HW 1 out
2	Sep 14	-- Sep 20	Variables, expressions & conditionals	Mod 2/3 Lab	HW 1 due/HW 2 out
3	Sep 21	-- Sep 27	Functions & testing	Mod 4 Lab	HW 2 due/HW 3 out
4	Sep 28	-- Oct 4	Iteration with while loops	Mod 5 Lab	HW 3 due/HW 4 out
5	Oct 5	-- Oct 11	Sequences	Mod 6 Lab	HW 4 due/HW 5 out
6	Oct 12	-- Oct 18	For loops	Mod 7 Lab	
7	Oct 19	-- Oct 25	MIDTERM EXAM		HW 5 due/HW 6 out
8	Oct 26	-- Nov 1	Recursion	Mod 8 Lab	
9	Nov 2	-- Nov 8	Handling errors	Mod 9 Lab	HW 6 due/HW 7 out
10	Nov 9	-- Nov 15	Dictionaries & Sets	Mod 10 Lab	
11	Nov 16	-- Nov 22	Classes & Objects	Mod 11 Lab	HW 7 due/HW 8 out
	Nov 23	-- Nov 29	THANKSGIVING BREAK		
12	Nov 30	-- Dec 6	Stacks & Queues	Mod 12 Lab	HW 8 due/HW 9
13	Dec 7	-- Dec 13	Program Efficiency	Mod 13 Lab	
14	Dec 14	-- Dec 20	FINAL EXAM		HW 9 due

2.1 -- Open Mic

To enhance the learning experience, weekly live sessions will be held that will provide students with the opportunity to discuss the material covered in that week's module, get answers to outstanding questions, and to see additional examples and applications. The timing of these sessions will depend upon the availability of students hoping to accommodate as many students as possible. The topics of these sessions will also be driven by the students. While participation is optional, it is encouraged. *The sessions are generally recorded and made available only to students enrolled in the course, the instructor of record, and any teaching assistants assigned to the course; however, if any student in attendance objects, I cannot record the session.*

2.2 -- Student Participation

It is extremely important for you to become engaged in this course. Like most programming courses, we are asking you to change the way you think so that you can express your thoughts to a computer. **This is not an easy task.**

Students are expected to keep up with the material as it is released each week and to ask for help when they need it (or when feeling overwhelmed). To help student's plan, we provide the following weekly cadence for this course:

- Carefully review the weekly material completing the *Check Yourself* questions as you go
- Solve all the *Coding Practice* problems that appear at the end of each module to practice what you have learned in the module. These are small practice problems that provide practice to students similar to those exercises you might do in a live class. Solutions can be submitted to codePost for automatic feedback.
- Complete the quiz posted with the module, due by **Thursday before 11:59pm ET**
- Plan to attend the weekly Open Mic which will be scheduled during the first week.
- Complete the lab assignment by **Saturday before 11:59pm ET**. Be sure to check the automatic feedback from the submission server and correct any issues it may point out.
- Work on the current homework assignment currently assigned, usually due **Thursday before 11:59pm ET**

While this course is designed for you to succeed, you will not unless you put the time in. In general, you should be prepared to spend 3-4 hours per credit hour for a master's level course. In computer science, we have found that it is often more than this. This means that **you should plan on spending a minimum of 12-16 hours per week on this course.**

This sounds like a lot of work and it is. Computer science is likely a brand new topic for you if you are taking this course. One of the goals of this course is to show you that CS can be fun. It can be challenging, it can be difficult, and it can be extremely satisfying when something works correctly. It should not be frustrating because frustrating things are not challenging or fun.

Frustration occurs when you are working on something and not making progress, especially if you are not sure about what you are doing. Frustration does not facilitate learning. If you find yourself getting frustrated, STOP. Go do something else, eat some dinner, take a walk, but most importantly, ask a question!!

We do not expect you to complete the assignments in this course completely on your own!

Please follow the **30-minute rule**: if you have been stuck on a problem more than 30 minutes and have made no progress despite your best efforts, please stop and get help. Message the professor, attend office hours, or consult a peer. If you don't get an answer immediately, do something else for a while. Please do not waste your time on one problem or bug in your code. Asking a question can both get you past the bug quickly and teach you how to fix it in on your own next time.

3 -- Course Assessment

Final grades will reflect students' effort and performance. The course grade will be roughly based on the following (subject to change):

Quizzes	=	6%
Lab Assignments	=	12%
Homework Assignments	=	42%
Exams	=	40%

Quizzes are assigned at the end of each module. The purpose of quizzes is to assess whether students are following the materials found in that module and to help them prepare for the written exams. Students can prepare for quizzes by carefully going through the week's provided materials consisting of video and text, and answering the *Check Yourself* questions found at the end of each video, and completing the *Coding Practice* at the end of each module.

Lab Assignments are assigned in lieu of formal recitation time and are designed as practice problems that illustrate the concepts covered in that week's module. They are designed to be completed in approximately 3-5 hours. They also serve as preparation for the next homework assignment. Labs become available to students who have submitted the Quiz for that week's module.

Homework Assignments will ask students to solve problems that require the application of all of the topics covered in the course thus far. Unlike lab assignments, these tend to be a bit larger and require you to combine the concepts from multiple modules using them in new ways. As the complexity of these assignments grow, so will the time that you will be provided to solve them.

In addition to programming, students will be expected to use good coding style, to include proper documentation, and to design test cases for the problem.

Exams will be given twice during the semester. The midterm exam will be given approximately halfway through the course while the final exam will be given during finals week. Since programming is, by nature, cumulative, so are the exams. Exam make-ups will only be permitted due to extraordinary circumstances and should be communicated as soon as possible.

3.1 -- Late Policy

In general, late work will not be accepted for any credit. However, we understand that sometimes something interferes with your ability to turn in an assignment by the specified time. For this reason, each student gets four (4) free, no-questions-asked late days for the course. The server keeps track of your late day and makes the extension process fair and transparent by getting the instructors out of the extension-granting business entirely. Instead, when you need an extension, you can take one as long as you have a late day remaining. The late days are to be used for contingencies such as needing extra time due to other deadlines, brief illnesses, other commitments, etc. It is not a good idea to use your late days to work for extra time on assignments, and then request more extensions for other circumstances: use them judiciously!

3.2 -- Grade Calculations

Grades will be calculated on an absolute basis: there will be no overall curving. The mapping of raw point totals to letter grades is given below. Please note that these grade boundaries may move slightly at the discretion of the instructor, but the grade boundary for A is unlikely to change.

93.00 -- 100.00	A	86.00 -- 89.99	B+	73.00 -- 76.99	C+	0.00 -- 64.99	F
90.00 -- 92.99	A-	82.00 -- 85.99	B	69.00 -- 72.99	C		
		77.00 -- 81.99	B-	65.00 -- 68.99	C-		

To progress, students are required to meet the grade point average (GPA) requirements for the MS Computer Science -- Align as determined by Khoury College of Computer Sciences (see [Khoury's website](#) for more information).

4 -- Course Materials

There is an associated Canvas page for this course. I will use it to post weekly reading assignments, lecture materials, labs, feedback, and grades.

4.1 -- Textbook

There is no required textbook for this course. However, there are several that might be useful to students who prefer to read materials rather than rely solely on the videos in this course. We have identified several really good resources and listed them on Canvas through the *Resource page*. Students are encouraged to use these resources. While these are not the only resources available on the web, these are the ones that I have identified as being most useful to students. I also encourage students to share resources that they find useful with the instructors.

4.2 -- Piazza

Piazza will be used for Q&A for this course. It provides students with a platform for getting you help fast and efficiently from classmates, the TAs, and the instructor. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. The Live Q&A feature will be available every week before the Open Mic for students to ask questions and upvote questions that other students have asked to help guide what we talk about during the Open Mic. You can find our Piazza page through the course page on Canvas. If you have trouble with Piazza, you can get help from team@piazza.com.

4.3 -- Microsoft Teams

In addition, we will use Microsoft Teams primarily for office hours, text messaging, and video conferencing. Each week, I will hold the Open Mic and, if possible, the recordings will be made available in the General channel on Teams.

4.4 -- Python

This class will be using the Python 3 programming language. You can download this for free from <https://www.python.org/>. By default, Python installs the *Integrated Development and Learning Environment* (IDLE) that can be used to do all of the development in this course. However, you are free to download another Integrated Development Environment (IDE) if you would prefer a different one. Getting started with Python and your chosen IDE is the objective of Lab 1.

4.5 -- codePost

[codePost](#) is used in this course to manage assignments and give students feedback. Each assignment will have a dedicated slot on codePost to accept submissions from students, to provide limited feedback to students before the deadline, and to provide manual feedback after grading. codePost also automatically keeps tracks of late days.

5 -- General Policies

5.1 -- Getting Help

In any course, all students experience some difficulties and frustration and everyone (professor included) makes mistakes. Everybody needs help sometimes and I would love to provide it! When it is your turn to have problems, do not suffer in silence! Talk to your professor immediately, before things become serious. You can do this via phone, email, or Microsoft Teams. She wants to help you solve these problems and does not make any judgments. Because she cannot always know when trouble starts, you must take the initiative of making initial contact.

5.2 -- Digital Etiquette (netiquette)

As an online student, communication is a bit different than in a regular face-to-face class. While you will be provided with several opportunities for face-to-face interactions, the primary form of communication in an online environment is the written word. Because this means you are missing body language cues and immediate feedback from your "listener", it is very important to understand some common rules for good online etiquette¹:

1. **Be respectful.** While it is easier to say hurtful or disrespectful things without standing face-to-face with someone, it is important to remember that your classmates and instructors are real people who are affected by the words you say and write. It is essential to keep in mind the feelings and opinions of others. *If you wouldn't say it to someone's face, you shouldn't say it online either.*
2. **Be aware of strong language, all caps, and exclamation points.** It is easy for written text to be misread and misunderstood. Have you ever sent a text message with good intention but your recipient thought you were being rude? If so, then you've experienced this firsthand. By being aware of strong language, you can identify potential confusions before sending messages. *Tip: Read everything out loud before you send it.*
3. **Be careful with humor and sarcasm.** Certainly you shouldn't avoid being funny. We love to see your personality shine through in online classes. Many of the instructional staff are exceptionally funny too. But make certain that it is clear you are being funny and not being rude. Emoticons and smileys can be helpful when conveying humor or sarcasm so that it is read correctly.
4. **Yes, grammar and spelling matters.** While texting, textspeak can be great for friends. In an educational setting (even online), however, keep it formal. Your written communication should be professional and reflect proper writing style. Save written shortcuts and less than stellar grammar for Snapchat if you must, but follow grammar rules for school.
5. **Don't post or share (even privately) inappropriate material.** Enough said here,

17 Rules for Online Etiquette. <https://achievethecore.org/7-rules-for-online-etiquette/>, Accessed 2019-11-26.

nothing is truly private online.

6. **Be forgiving.** Remember that not everyone will know these rules before posting. Try to be understanding of others when they struggle with written communication. It is very different than simply talking to a person face-to-face.

5.3 -- Citations

Everybody is aware of the vast amount of knowledge that is available on the Internet. While every assignment in this class is designed to be solvable using the methods we have discussed in class, your Professor recognizes that every student will end up using the Internet at some point or another. In general, web searches should be limited to how to accomplish small tasks in Python. As a student, your job is to be honest and forthright with your efforts. It is of utmost importance to your learning that you never just cut-and-paste a solution to a homework problem; instead make the effort to understand the solution well enough to put it into your own words and **be sure to cite your sources**. Citations should include references (paper, website, or other) for any site that you used to research a solution. Proper APA format should be used. For websites this includes name of website, title of the article, the url, and the date of retrieval. If you find yourself spending any significant amount of time searching the web, you should come and see me because it is a sign that something is not working for you in this class.

5.4 -- Academic Integrity

You are expected to read, understand, and follow the University's policies on *Academic Integrity*. Each student is expected to do his or her own work. Violations of academic integrity will result in a zero on the corresponding assignment along with harsher penalties for more widespread problems. Here are a few examples of academic dishonesty:

- Working with one or more partners on an assignment.
- Submitting a copy of work done by another student, with or without their knowledge.
- Submitting work that was primarily found on the web or provided by someone else outside of this class.
- Submitting work by anybody who took this course in the past whether the course was here at Northeastern or at another campus or institution.
- Providing or receiving significant help to another student on an assignment.

General discussions with other students are okay but should be done away from the computer leaving only memories.

5.5 -- Reasonable Accommodations

The goal is that every student should be able to participate in this course. If you require any special accommodations, let me know immediately so that we can work out appropriate arrangements.

5.6 -- Student Feedback

Your opinions are very important to me. In addition to the university-required student evaluation form at the end of the semester, I will be asking for your feedback at least once about halfway through the semester. However, if you have concerns about the course, don't wait until you are asked. You can contact me any time!

5.7 -- Recommendations

Recommendations for coop, employment, or further studies (MS or PhD program) are meaningful and I would like to help when I can. However, I cannot provide a recommendation unless you have come to at least five office hours so that I can get to know you, you've earned at least an A- in the course, and you discussed your project or at least the last practicum with me. Should you wish to ask me for a recommendation letter or reference, please check with me ahead of time, and email me your CV, resume, and other supporting documents, in addition to reminding me when we spoke, what you did as a project, and during which term you took a course with me.