Course Syllabus

Instructor Information

Instructor: Sami Rollins (please call me Sami!)
Contact: s.rollins@northeastern.edu, (mailto:s.rollins@northeastern.edu,) but the quickest way to reach me is via Teams
Zoom: https://northeastern.zoom.us/j/91464027899 (https://northeastern.zoom.us/j/91464027899)
Office Hours: Wednesday 4:45pm - 5:30pm (in person and Zoom); Friday 11am - noon (Zoom only); and by appointment
Location: Arlington Tower 15th floor

Teaching Assistant

TA: Jessica Teurn
Contact: teurn.j@northeastern.edu (mailto:teurn.j@northeastern.edu) or via Teams
Zoom: https://northeastern.zoom.us/j/97727632396?pwd=VEtTTkF3MSs0NkQxRnBpbkwxbFJQZz09 (https://northeastern.zoom.us/j/97727632396?pwd=VEtTTkF3MSs0NkQxRnBpbkwxbFJQZz09)
Office Hours: Wednesday 10am - 11am; Saturday 1pm-2pm
Tutoring Appointments: TBA

Class Hours

Monday and Wednesday 3:00pm - 4:40pm
Recitation Tuesday 3:00pm - 4:30pm

In-person attendance is expected for both lecture and recitation.

Students are expected to arrive on time and attend all classes; however, attendance will not be taken nor will it be part of the grading rubric. ***Material is not guaranteed to be available on-demand or remotely.***

Students who miss class for reasons of illness or personal emergency should communicate with the instructor early and often to arrange appropriate flexibility. Students who miss class for other reasons should not expect the instructor or TA to provide extra assistance to help them catch up on material covered in a missed class.

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.

Course Prerequisites: The course is suitable for students in the ALIGN MS in CS program. It assumes no previous programming experience. We're happy you're here!

Course Objectives:

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

Course Resources

The **Course Schedule (https://northeastern.instructure.com/courses/136732/pages/course-schedule?wrap=1)** provides links to a library of videos specifically produced for Align students. These will serve as a useful foundation for your learning.

There is no required text, but the following *freely available* books may be useful:

- **How to Think Like a Computer Scientist (http://openbookproject.net/thinkcs/python/english3e/)** by Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers
- **Welcome to Python for you and me (https://pymbook.readthedocs.io/en/latest/)**
- **Think Python, 2nd Edition (https://www.oreilly.com/library/view/think-python-2nd/9781491939406/)** by Allen B. Downey
  - Free if you connect to O'Reilly Online via the NU library page for computer science: **https://subjectguides.lib.neu.edu/compsci (https://subjectguides.lib.neu.edu/compsci)**

Announcements

Announcements and class discussion will occur on the Microsoft Team set up for the class. Join the team here: **CS 5001 - Spring 23 - ARL (https://teams.microsoft.com/l/team/19%3aw4kjR1fRq_P32GxZdN6c9h6CxKwlsFv9Hnb_K-XQsb01%40thread.tacv2/conversations?groupId=1c9e1128-693a-4132-8b71-b08277893048&tenantId=a8eec281-aaa3-4dae-ac9b-9a398b9215e7)**

Students are responsible for staying current on all announcements made via the class Team. You are expected to participate in discussion on Teams. Make sure to remain professional in your communications.

Class Workload

It is important for you to be highly 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, to ask for help when they need it (or when feeling overwhelmed), to complete assignments on time, to actively participate in class discussions on Teams. Students are highly encouraged to help and support each other in appropriate ways. We can also often arrange for individual tutoring in cases where there is particular need, so please let me know if you feel you need additional help staying on top of the material.

While this course is designed to 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. This means that **you should plan on spending a minimum of 12-16 hours per week on this course**.

Assessment

- 15% Labs (weekly)
- 45% Projects (5)
- 30% - Exams (3)
- 10% Quizzes and Participation

Labs: Labs will be due weekly. Students are expected to begin labs during the recitation period and complete the exercises before the deadline. Unless otherwise noted, labs are to be completed ***individually.*** Labs generally consist of 4 - 8 shorter programming questions that focus on one or two computing concepts.

Projects: Students will be assigned five ***individual*** projects Projects are generally longer programs that require students to synthesize what they have learned over the previous weeks. The last project will be due during finals week in lieu of a final exam. Most projects will require an interactive demonstration and grading session with an instructor. Projects may not be resubmitted. All grades are final.

Exams: There will be three (3) exams over the course of the semester: February 8, March 15, and April 17. Since programming is, by nature, cumulative, so are the exams, although the emphasis will be on recent material. Exam make-ups will only be permitted due to extraordinary circumstances and should be communicated as soon as possible.

Quizzes and Participation: You are expected to attend class, participate in class discussion, and participate in online discussion on Teams. Quizzes and other in-class exercises may be administered at any time. Failure to complete them as required during your assigned class period will result in a 0 on the assessment.

Grading Scale:

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

Policies

Seeking 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 Zoom. 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.

Code Review: Any student may be asked to come in for code review for any assignment. A student who does not meet with the instructor for a code review in a timely manner will be given a 0 on the assignment in question. A student who is unable to explain their code; answer the instructor's questions about their code; or reproduce their code as expected may receive a deduction of up to 100%, resulting in a 0 on the assignment.

Late Policy: All deadlines are firm. All Labs and Exams must be completed by the deadline. Projects must be completed by the deadline with the following exceptions:

- All students will be allowed ***one*** 24-hour extension without penalty to be used on a ***project*** of their choosing. This dispensation may only be used for ***one project***.
- All students will be allowed ***one*** 24-hour extension with 15% penalty to be used on a ***project*** of their choosing. This dispensation may only be used for ***one project***.

If a student wishes to use both exceptions for the same project they may submit that project late by up to 48 hours for a penalty of 15%.

To use your exception send the professor a Teams message or email *in advance of the deadline.*

Further exceptions to this policy are made only in the case of a *verifiable* medical or family emergency. Extensions and makeup exams must be arranged *PRIOR* to the original deadline unless in case of extreme emergency (such as an emergency room visit).

Students who have legitimate, personal circumstances that warrant special consideration should speak with the instructor as early as possible to arrange for appropriate flexibility. ***Job assessments and interviews are not a legitimate reason for additional flexibility.***

Citations: Everybody is aware of 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 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. 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.

Academic Honesty: 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:

- Giving your code to another student.
- Submitting code written by anyone other than yourself as your own.
- Copying text from the web.
- Discussing implementation details of a solution with another student.
- Discussing code-specific details of a solution with another student.
- Soliciting help on websites such as Stack Overflow.

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

Students With Disabilities: 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.