CS 100 Roadmap to Computing Course Syllabus - Fall 2024

Instructor: Junilda Spirollari Office: GITC 4405 Email: <u>Junilda.Spirollari@njit.edu</u> Office Hours: MF, 11:30am – 12:50pm

Course Description

This course offers an introduction to programming and problem-solving skills using Python, a very high level language. Topics include:

- programming environments and tools, including editor and debugger
- basic strategies for problem solving
- integer, floating point, string and logical types
- lists, tuples and dictionaries
- files
- conditional, repetition, function and other constructs that control the flow of execution of a program
- class design and object-oriented programming
- exceptions
- the use of high level data types such as lists, strings and dictionaries, in problem representation.

The course also includes a project where the students investigate a topic of current interest in computing, write a report on the topic and present the findings.

Computing is a profession that requires lifelong learning, which is pursued through activities and materials similar to those employed by students. In this course, the students are expected to master programming and problem-solving techniques while also learning to use effective learning strategies and materials – essential skills for a knowledge-intensive profession. This includes:

- effective use of knowledge resources: reading documentation, asking and answering peer questions, consulting with experienced individuals, and searching on-line for answers.
- tools and methodologies: testing code for correctness, using an integrated development environment (IDE) and debugger, and writing specifications and documentation.

Mastering this material requires extensive hands-on practice. Students should plan to spend twice as much time studying and working on problems outside of class (approximately 6 hours per week) as they do in class.

Course Resources

The textbook is *Think Python* by Allen B. Downey, 2nd edition. This is an open source book. It is available without charge in HTML and PDF formats at <u>https://greenteapress.com/wp/think-python-2e/</u>.

A print format is published by O'Reilly (campus bookstore or online). There is also a Kindle edition. The textbook is required. You may use any one (or more) of the formats.

Please note that a 3^{rd} edition of the book has recently been published and can also be used (<u>link to 3^{rd} edition</u>). However, the homework assignments are based on the 2^{nd} edition. It is your responsibility to ensure you complete the correct exercises from the edition you choose.

Python language version 3.12 can be gotten at <u>python.org/downloads</u>. This includes the IDLE development environment, help files, modules and other parts of the standard distribution. You will need to get Python and install it on your personal desktop and/or laptop computer. You can download Python for Windows, Mac or Linux environments. There is no charge for Python. **Be aware that Python 3.12** *cannot* **be used on Windows 7 or earlier.**

Other course materials:

- PythonTutor, a program for stepping through and visualizing the execution of Python code at <u>pythontutor.com</u>.
- Wingware 101 IDE, which can be downloaded for various platforms at <u>wingware.com/downloads/wingide-101</u>.

- Thonny IDE, downloadable from <u>thonny.org</u>.
- The Community Edition of the PyCharm IDE jetbrains.com/pycharm.
- The Mu development environment, available at <u>codewith.mu</u> and various other online resources.

Course Communication

Canvas (<u>canvas.njit.edu</u>) will be the platform for posting lecture notes, submitting assignments and engaging in course discussions. For any questions and additional support, you are welcome to reach out to instructors and classroom assistants via email.

Class Attendance

Class attendance is mandatory. A student who misses more than five classes will be dropped, without credit. Getting to class late or leaving early counts as half an absence.

Recitation

Attending recitation is an important checkpoint in assuring your grasp of the material being covered and correctly solving assigned problems. Before attending recitation, you should have already read the assigned material and made progress on your current homework. During recitation, you should come prepared to ask any questions you have identified that require clarification. This will help confirm whether your understanding of the material is correct. You may also meet with the instructor for your section. All instructors have posted office hours.

Homework

Homework must be submitted through Canvas on or before the due date and time. Late submissions will **not** be accepted except under special circumstances, such as jury duty or a medical issue, for which you must provide appropriate documentation.

A homework assignment will typically require you to write code that produces a specified output. No credit will be given for code that does not run. Getting a correct solution will often involve writing, testing, and revising your code multiple times until it meets the required specifications. Be prepared to spend the bulk of your time getting it right. Remember: only correct code will earn credit.

During the write-test-debug cycle you are encouraged to use the debugging tools available in your development environment, ask questions on Canvas, and discuss the problem with others. However, it is imperative that you adhere to the university's academic integrity policies. Cheating, plagiarism, and any other form of academic dishonesty will not be tolerated. To avoid misconduct student's work:

- must be original. Students may **not** copy code from external sources without permission of the instructor or course assistant (e.g. online sources, other students' code, previous semester coursework, etc.).
- may **not** include concepts, methods or code in the homework that have not been introduced or taught in the class up until the submission date.
- may not contain code that they cannot understand or explain if asked.

All assignments must be completed independently unless explicitly stated otherwise. Violations will result in serious consequences, including a zero on the assignment, and may lead to further disciplinary action by the university. Ensure that all work submitted is genuinely your own and seek help if you encounter difficulties.

Roadmap Project

Each student will work on a Roadmap project, consisting of a written and an oral presentation, either individually or with a partner. Partners are held to a higher standard than individuals.

Online Discussions

You are expected to actively engage in weekly discussion forums on Canvas. These forums are designed to foster a collaborative learning environment where all students contribute to discussions. Participation will not only help deepen your understanding of the course material but also enhance your overall success in the class. Your contributions should be thoughtful, demonstrate critical thinking, and reflect on the content covered in lectures and readings.

Class Participation

Presenting your homework answers and presenting your projects in class is a regular part of the course. Asking and answering questions, taking quizzes, solving programming problems — individually or in groups — is a regular part of class meetings.

Cell phones must be turned off during class. During class time you may not play games, text, email, browse the web or engage in other activities that are not part of the class.

Collaboration and Individual Responsibility

Collaboration is a valuable learning tool, and you are encouraged to study and discuss coursework with your peers. However, it is essential that all submitted work reflects your individual understanding and effort. You may discuss concepts and approaches with others, but the work you submit must be your own. You should be able to explain and justify all aspects of your submission.

Tutoring hours are also provided by our college. The link to the tutoring page is: <u>Tutoring | Ying Wu College of</u> <u>Computing (njit.edu)</u>.

What You Will Learn

By the end of this course, you will be expected to know and be able to use these pieces of the computing toolkit to compute the solution of a specified problem:

- Devise a problem representation (model) and a sequence of steps (algorithm) that correctly solve the problem posed
- Write a program that implements the algorithm, using
- A core set of Python language elements (keywords, syntax, variables, modules)
- Basic data types (integers, floats, strings, booleans, lists, tuples, dictionaries) and operations on them
- Statements that perform console/file input and output
- Statements that control the sequence of execution (if/else, for, while)
- Statements that are structured into function calls

Each homework assignment is designed to help you practice these concepts and skills, while providing valuable feedback on your progress. You are expected to submit working solutions to every homework assignment. Since the course material builds progressively, any gaps in your understanding will compromise your ability to successfully complete the course. You understand material when you are able to use it to solve problems and to explain your solutions. Both midterm exams and the final exam will assess your overall understanding and mastery of the course content.

Topics to Be Covered

The list of topics to be covered includes the following:

- Getting Started with Python
- Expressions, Variables, and Assignments
- Built-in Data Types
- Sequence Data Types (Strings, Tuples, and Lists)
- Python Standard Library
- Formatted Output and User Input
- Conditional Execution and Boolean Logic
- Iteration
- Functions
- Argument-Passing and Return Values
- Data files
- Dictionaries
- Designing and Using Classes
- Scope and Namespaces
- Exceptions
- Debugging and Testing

Overall Course Score Formula

Homework	12%
Discussions	5%
Midterm 1	20%
Midterm 2	20%
Final Exam	30%
Roadmap Project	10%
Discretionary	3%

The letter grade is based on the overall course score.

Grade Formula						
Grade	Α	B +	B	C+	С	D
Overall Course Score Cutoff	90	85	80	75	70	60

Common Exams

There are two common midterms: Monday, September 30^{th} and Monday, October 28^{th} , 4:00pm-5:30pm. The final exams period is December $15^{th} - 21^{st}$. The CS100 final exam will be during this period, but the date has not yet been set. *Be sure that you will be present for all of your exams*.

You must bring ID to all exams. Students with special needs are advised to make arrangements with the Office of Accessibility Resources and Services, Kupfrian Hall 201.

There are no makeup exams. If you miss a midterm because of a documented special circumstance determined by the Dean of Students you may receive an imputed grade based on the other midterm and the final exam.

Grade Appeals

If you believe that you deserve more credit than you have been awarded on a particular common exam problem, you may request, **at the time the exam is returned**, that it be regraded. Your entire exam will be regraded, which may result in points being added or subtracted.

If you believe that you deserve more credit than you have been awarded on a particular homework problem, you may request, **within 48 hours of the grade being posted**, that it be regraded. Your entire homework will be regraded, which may result in points being added or subtracted.

University Code on Academic Integrity

Read the University Code on Academic Integrity (<u>njit.edu/policies/sites/policies/files/academic-integrity-code.pdf</u>). It describes infractions of academic integrity and penalties for violations, including, for the most serious violations, an XF grade in the course or expulsion. **All work that you represent as your own must, in fact, be your own.** Work done by others must be given proper credit.

You will be informed of any modifications of this syllabus during the semester.