CS 100 Roadmap to Computing
Course Syllabus – Fall 2025

Instructor: Junilda Spirollari
Email: Junilda.Spirollari@njit.edu
Office: GITC 4405
Office Hours: MW, 10:00am – 11:20am

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

- computer components and binary representation
- computational thinking and pseudocode
- basic strategies for problem solving
- python basics: variables, int, float and Boolean data types, expressions
- strings, lists, tuples and dictionaries
- conditional statements (if, elif, else)
- modules
- loops: for and while
- functions and namespaces
- file handling: reading and writing files
- 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 Materials

- The course uses zyBooks, an interactive online textbook, as a core resource for learning. Instructions for accessing zyBooks will be provided on Canvas.

- Supplementary notes will be provided on Canvas.

- Python language version 3.13 can be gotten at python.org/downloads. 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 3.13 *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 pythontutor.com.
  Wingware 101 IDE, which can be downloaded for various platforms at wingware.com/downloads/wingide-101.
  Thonny IDE, downloadable from thonny.org.
  The Community Edition of the PyCharm IDE jetbrains.com/pycharm.
  The Mu development environment, available at codewith.mu and various other online resources.

Course Policies

- **Course Communication**
  Canvas (canvas.njit.edu) will be the platform for posting lecture notes, submitting assignments and engaging in course related activities. 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.
  From day one, only registered students are permitted to attend classes, either in-person or remotely.

- **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. Recitations will be conducted by the Course Assistants and the recitation schedule will be available on Canvas by the end of the first week of classes. You may also meet with the instructor for your section. All instructors have posted office hours.

- **Homework**
  Homework must be submitted through Canvas/zyBooks 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.

- **Generative AI**
  The use of AI tools to generate solutions for homework, exams, or other individual assignments is **strictly prohibited** in this course. Students are only permitted to use AI tools when explicitly authorized by the instructor for specific tasks or activities. Any unauthorized use of AI to complete assignments will be considered a violation of the academic integrity policy. If you are unsure whether the use of AI is allowed for a particular assignment, please consult the instructor before proceeding.

- **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 Participation**
  You are expected to actively engage in weekly participation activities on zyBooks. These activities are designed to foster an interactive learning environment. Participation will not only help deepen your understanding of the course material but also enhance your overall success in the class.

- **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: Tutoring | Ying Wu College of Computing (njit.edu).

- **Student Absences for Religious Observance**
  NJIT is committed to supporting students observing religious holidays. If you anticipate any conflicts between course requirements and your religious observances, you must notify the instructor in writing by **the end of the second week of classes**, and **no later than two weeks before** the anticipated absence. Reasonable accommodations will be provided for the missed coursework if the religious observance directly prevents you from completing or attending these activities.
  However, no accommodations will be provided for assignments with deadlines that allow sufficient time for completion (e.g., assignments with a week or more to complete). Students are expected to plan ahead and ensure timely submission of their work in these cases.

Course Educational Objectives

By the end of this course, students will:

1. Develop foundational knowledge of Python programming.
2. Acquire problem-solving skills through devising algorithms and implementing solutions.
3. Learn to design, implement, and test software using core programming constructs, including functions, parameter passing and return values.
4. Understand and apply basic data types in computational problem-solving.
5. Build competence in file handling.
6. Understand the basics of object-oriented programming.
7. Investigate current topics in computing, producing a written report and presentation.

Course Learning Outcomes

These are specific, measurable skills students will gain by the end of the course:

1. Utilize basic data types (int, float, bool, str, list, dict, tuple) to represent and manipulate information.
2. Design algorithms using control structures (loops, conditionals) to solve computational problems.
3. Implement modular solutions by writing functions with parameters and return values.
4. Perform file input/output operations to process persistent data.
5. Construct simple classes to model real-world entities using object-oriented principles.
6. Read a program specification and develop a program that meets the requirements.
7. Trace program execution to identify errors and understand program behavior.
8. Communicate findings of a computing-related investigation through a written report and oral presentation.

Overall Course Score Formula

Homework	15%
Online Participation	8%
Midterm 1	18%
Midterm 2	22%
Final Exam	25%
Roadmap Project	10%
Discretionary	2%

The letter grade is based on the overall course score.

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

Exam Policies
There are two common midterms: Monday, October 13, 2025, and Monday, November 10, 2025, 4:00pm–5:30pm. The final exams period is December 14 – 20, 2025. The CS100 final exam will be during this period, but the date has not yet been set. All exams will be conducted in person, using Canvas, and will require Lockdown Browser. ***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
Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree that you are working on. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the academic code of integrity policy that is found at: NJIT Academic Integrity Code.

Please note that it is my professional obligation and responsibility to report any academic misconduct to the Dean of Students Office. Any student found in violation of the code by cheating, plagiarizing or using any online software inappropriately will result in disciplinary action. This may include a failing grade of F, and/or suspension or dismissal from the university. If you have any questions about the code of Academic Integrity, please contact the Dean of Students Office at dos@njit.edu

Tentative Agenda

Week	TENTATIVE AGENDA	Material	Due Work
1	Introduction to Computing and Abstraction	Module 1	HW01
2	Computational thinking + Intro to Python basics	Module 2	HW02
3	Strings, Lists and Tuples	Module 3	HW03
4	Conditional Statements	Module 4	HW04
5	For-loop	Module 5	HW06
6	For-loop + Review for Mid1	Module 5	
	Midterm Exam 1		
7	Function Definitions	Module 6	
8	Functions + Namespaces	Module 6	HW06
9	Str Methods	Module 7	HW07
10	While loop + Review for Mid2	Module 8	HW08
	Midterm Exam 2		
11	Files	Module 9	HW09
12	Dictionaries	Module 10	HW10
13	User-defined classes	Module 11	HW11
14	Exceptions + Review (Final)	Module 12	HW12

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