CS 100 Roadmap to Computing
Course Syllabus for Spring 2024 - Section HS1

Instructor: Trevor Summerfield (trevor.summerfield@njit.edu)

Classroom Assistant: Sarisha Singh (ss4886@njit.edu)

Course Description
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 data types
- lists, sets and dictionaries
- files
- conditional, repetition, functions and other constructs that control the flow of execution of a program
- the design of classes
- the use of high level data types such as lists, strings and dictionaries in problem representation

The course also includes a project in which the student investigates a topic of current interest in computing, writes a report on the topic and presents it in class.

Computing is a profession that requires lifelong learning, which is pursued through activities and using types of materials that are similar to those employed by students. In this course, the student, in addition to mastering the programming and problem solving materials, is expected to learn to effectively use learning strategies and materials — learning how to learn efficiently in preparation for a knowledge intensive profession. This includes effective use of knowledge resources — reading documentation, asking and answering peer questions, consulting with more experienced persons, and searching on-line for answers. It also includes tools and methodology — testing to verify the correctness of code, use of an integrated development environment (IDE) and debugger, writing specifications and documentation.

Learning this material requires extensive hands-on practice. You should plan to spend twice as much time studying and working problems outside of class (that is, about 6 hours per week) as you 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 http://greenteapress.com/wp/think-python-2e/.

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.

Other course materials:

- Python language version 3.9 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 3.9 *cannot* be used on Windows 7 or earlier.**
- Thonny IDE, downloadable from thonny.org.

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.

Homework
Homework must be submitted through Canvas on or before the due date and time. It will not be accepted late except for special circumstances (such as jury duty or medical problem), for which you must provide 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 require that your solution be written, tested, and then rewritten multiple times until it fulfills the specification. Expect that the bulk of your time will be spent getting it right. Remember: only code that is correct is worth anything. During the write-test-debug cycle you may — and are encouraged to — use the debugging facilities in the development environment, pose questions on Canvas, and discuss the problem with others, however, you may **not** discuss solutions or share code with others.

Roadmap Project
Each student will work on a Roadmap project, consisting of a written and an oral presentation.

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. Repeated infractions will be counted as half an absence.

Course Communication
Canvas (canvas.njit.edu) will be used to post lecture notes, to submit homework and for course discussion. You may also email instructors and classroom assistants.

Collaboration and Individual Responsibility
You are encouraged to study and to work on assignments together with others; collaboration is a basic learning technique. You may not take credit for the work of others. You must understand and be able to explain all work that you submit.

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 gives you practice on these concepts and skills, and provides feedback on your progress. You are expected to submit working solutions to every homework assignment. Each element of this course builds on previous material, and 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. Each of the two midterm exams and the final exam test your mastery of the material.

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

Exams
There are two midterms: Saturday, March 9th and Saturday, April 6th. The final exam is Saturday, May 4th.
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 (njit.edu/policies/sites/policies/files/academic-integrity-code.pdf). 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.