CS 151: Introduction to programming in Python

Willamette University, Fall 2022

Instructor:

Fred Agbo, PhD
friday.agbo@uef.fi
Office:
Office Hour:

Web:
Lecture: MWF (TBD)
Lecture Hall: (TBD)

(This syllabus is subject to change as the semester progresses - particularly the schedule. Text highlighted in yellow color are just placeholders and would be updated subsequently)

Brief Course Description

Introductory programming course provides the foundation for computational and computer science knowledge by introducing students to the fundamentals of programming and problem solving in general. Problem solving and computational skills are necessary to communicate with computers, therefore, this course is imperative for contemporary computer science students. This course focuses on introducing concepts of computer science and programming to students through the use of the programming language Python. Fundamental concepts such as understanding variables, logic and loops will be covered as well as object-oriented programming and basic visualization. Students would learn from varieties of short exercises that allow them to practice by writing Python scripts. After completing this course, students should leave the course having a comfort and familiarity in writing Python scripts to solve computational problems.

Prerequisite(s): There is no prerequisite for enrolling in this course. As a matter of fact, the course is open to all students.

Note: A minimum grade of C- is required for this course to count toward university credit.

Credits: 1.0

Resources for this course:

Mainly, we will be using the book entitled “Programming in Python” authored by Professor Eric S. Roberts throughout the semester to supplement what we talk about in class and other online resources that I may provide as the course progresses. The book is available for free via PDF on the course website <here>.

Course Objectives:

The objective of this course is to see that students are able to gain the following knowledge:

1. The basics of problem solving through coding in Python
2. The fundamentals of programming concepts such as variables, logic, and control structures
3. Think algorithmically and apply stepwise approach to problem-solving
4. Understand the basics of object-oriented programming, when to use it, and why it is useful
5. Debug and test programs to ensure they are working as intended

Obviously, programming and the field of computer science in general is vast and the entirety of what is possible in Python could not be covered in a single course. Therefore, this course seeks to provide solid fundamentals to students so that they might be motivated to continue their learning to becoming future experts in computing.

Student Learning Objectives (SLO):

This course has been designed to ensure that at its completion, students should be able to:

- apply stepwise approach to describe, design, and implement processes of problem solving; test and debug programs to solve a problem. Indeed, problem solving skill is a core aspect of programming, and students must be able to take a given problem, break it down into solvable steps, and implement each individual piece to achieve their solution.
- understand concept of algorithm, how to design algorithms and transform them into computer programming. Many types of problems share similar solutions. Therefore, knowing how to design and use algorithm, when to use them, and the benefits of one algorithm over another is important.
- recognize and construct common programming concepts, including variables, loops, functions, I/O, and logic. These are important tools that students can use to construct all their programs efficiently and creatively.
- understand different data structures suck as lists, dictionaries and tuples and recognize the proper ways and times to use each. Storing data is very crucial in programming and Python offers a variety of ways to store data. By understanding the data structure paradigm, one can decide on the correct one to address a problem when coding.

Feedback and Course Grading

I will ensure that feedback is provided on each homework, projects, mid-term exams, and final exam. Aside from the feedback, certain percentage of the grade will be awarded to students for active participation in the classroom, lab session, and via other communication channels created for this course. The weighting of the grades include:

- Participation 5%
- Homework (Problem sets) 25%
- Projects 20%
- Lab work 10%
- Mid-term test 20%
- Final Exam 20%

Course Structure and Assessment

This course will consist of lectures, labs, home-works, projects, mid-term exam, and a final exam. Each course component is very important to gain expected learning outcomes including grades. Looking at its comprehensive nature, students are encouraged to give their best in all the components of the course. Full attendance and participation is mandatory in all the components of the course.

- **Lecture**
  Lectures will be held every Mondays, Wednesdays, and Fridays (except for holidays or other events from the university that may override). The time for the lecture is 9:10 am in "TBD venue". Please check out the course syllabus <here>. Slides and other resources for the lecture can also be accessible from the course webpage. While I explore how to conform to accessibility best practices, I will appreciate your feedback and let me know if some of the materials are inaccessible.

- **Lab session**
  On Wednesdays after the class hour, students proceed immediately to lab session which last for an hour. Generally, most of this time will be devoted to an exercise that emphasizes and provides practice for what we talked about in class that day. The lab session can also provide opportunity for students to ask questions to student peer mentors in a smaller group.

- **Homework**
  There will be homework in most of the weeks, which will be shared by the middle of the week. Each homework would contain problem sets that focuses on previous classes, and the aim is to allow students to demonstrate their understanding of topics covered. It also allows students to master programming and problem-solving through practical experience. Each week's homework is due at 11:59pm on Sunday. Problem sets may consist a few theory/essay type questions and some problems requiring programming. If the solutions are written using any word processing software such as Microsoft Word or LaTeX, they should be saved as pdf. Both pdfs and any code written will be submitted through GitHub Classroom before the deadline. We shall demonstrate the submission process in lab.
  In the words of Professor Jed: "Programming is very hands-on, and the odds are high that you will not do well in the course if you do not practice! As far as I know, the best method to gain proficiency is solving problems and writing code. There are no real 'shortcuts.' …. Working in groups and helping others is very encouraged, though students must turn in their own work. I highly recommend helping and instructing other classmates if you feel proficient on a topic, both to help them and because there is no better way to identify gaps in your own knowledge than when you attempt to teach something". I strongly advice students to submit each homework 'as-is' before the deadline. The reason is that I will be interested to see the attempt made and even provide practical feedback to the student. And because my grading style takes every step taken to address a problem into consideration, it is possible for students who could not complete the homework but had made some concrete efforts to earn some grade points.

- **Projects**
  This course consists of 5 projects that are extensive compared to home-works. The aim is to allow students to demonstrate different concepts of programming in order to implement a

functional program. The average time and efforts required to deal with the problems might be more than what is required for home-works.

- **Mid-term and Fina Exams**
  There will be 1 mid-term exam and 1 final exam this semester. The mid-term exam will try to examine students' understanding of the topics covered by solving programming problems on papers. I have deliberately structured the exams this way because it is possible for a student who could not obtain good score in the first exam– due to some reasons – to make it up in the second exam. Besides, I will provide feedback to the mid-term exam so that students can learn from their mistakes. The final exam also will be paper based. This mean that in both mid-term and final exam, the use of computer is not allowed. The reason why I would prefer a paper-based exams is to allow the students to think big and implement their solution without concentrating on testing and debugging their codes on computers, which may waste time and cause anxiety in exam condition and may make the student to lose substantial grades points.

Course Policies:

Late Submission and Incomplete Policy

I am adapting Jed's course policy which I found very generous enough! I totally understand that as human, things can sometimes come up or go wrong and you are unable to get an assignment turned in on time. This kind of situation calls for some flexibility where I could consider accepting of late submission. However, this flexibility MUST be subject to my awareness and approval. Therefore, if any student is in this kind of unfortunate situation and would need more time to submit homework or project a bit late, please, contact me immediately. I must receive an email and reply to it in order to implement this policy. As a matter of rule, no lateness beyond 1 day (24 hours) can be tolerated for any given homework/project.

In the case where a student could not complete tasks in this course due to prolonged illness or family emergency or other genuine factors that severely affects the academics, I will see how to address such a case by awarding incomplete grade. However, if reasons for not completing the course are based on falling behind through lack of motivation, understanding, or time management skills, there will not be incomplete grade awarded to such a student. Please, feel free to come to me and discus any concern you may have that could affect your completing the course. I'm very much available to provide supports for you to succeed in this course!

Willamette Policies:

This section has been largely developed/adapted from the Willamette University Academic Policy which can been accessed via this link. As a result, the instructor employer's view constitutes the information represented here except for minor edits made by the instructor to adapt the context to the specific case study of this class. I will appreciate if students can reach out to me on any issues that have not been represented regarding policies that guides this class.

Academic Honesty

Cheating is defined as any form of intellectual dishonesty or misrepresentation of one's knowledge. Plagiarism, a form of cheating, consists of intentionally or unintentionally representing someone else's work as one's own. Integrity is of prime importance in a college setting, and thus cheating, plagiarism, theft, or assisting another to perform any of the previously listed acts is strictly prohibited. An instructor may imposed penalties for plagiarism or cheating ranging from a grade reduction on an assignment or exam to failing the course. An instructor can also involve the Office of the Dean of the College of Liberal Arts for further action. For further information, visit:

http://www.willamette.edu/cla/catalog/resources/policies/plagiarism_cheating.php.

Time Commitments

Willamette's Credit Hour Policy holds that for every hour of class time there is an expectation of 2-3 hours work outside of class. Thus, for a class meeting three days a week you should anticipate spending 6-9 hours outside of class engaged in course-related activities. Examples include study time, reading and homework, assignments, research projects, and group work.

Diversity and Disability

Willamette University values diversity and inclusion; from my background, I have promoted diversity in several ways and now that I'm in Willamette where there is commitment to diversity and inclusion, we will ensure a climate of mutual respect and full participation. Our goal is to create learning environments that are usable, equitable, inclusive, and welcoming. If there are aspects of the instruction or design of this course that result in barriers to your inclusion or accurate assessment or achievement, please notify the professor (in this case Fred) as soon as possible. Students with disabilities are also encouraged to contact the Accessible Education Services office in Matthews 103 at 503-370-6737 or accessible-info@willamette.edu to discuss a range of options to removing barriers in the course, including accommodations

Tentative Course Outline:
The weekly coverage might change as it depends on the progress of the class. However, I highly recommend you follow along with the reading, as it makes a large difference!

Week	Monday	Wednesday	Friday	HW&PS
1 Introduction	1st Syllabus	1st Karel the Robot v1	1st Karel the Robot v2	1st HW 0
2 Python	2nd Introduction to Python	2nd Control statements v1	2nd Control statements v2	2nd HW 1
3 Functions	3rd Function v1	3rd Function v2	3rd Graphics v1	3rd HW 2
4 Graphics	4th Graphics v2	4th Graphics v3	4th 1st Mid-term exam	4th HW 3
5 Interactive programs	5th Interactive programs v1	5th Interactive programs v2	5th Interactive programs v3	5th HW 4
6 Strings	6th Strings v1	6th Strings v2	6th Strings v3	6th HW 5
7 Lists	7th Lists v1	7th Lists v2	7th Mid-semester day	7th Project 1
8 Lists	8th Lists v3	8th Lists v4	8th 2nd Mid-term exam	8th -
9 Classes and Objects	9th Classes and Objects v1	9th Classes and Objects v2	9th Classes and Objects v3	9th Project 2
10 Inheritance	10th Inheritance v1	10th Inheritance v2	10th The Engigma Machine	10th Project 3
11 Dictionary	11th Dictionary and Hashing v1	11th Dictionary and Hashing v2	11th Intro to Data Structure v1	11th Project 3 due
12 Intro to Data Structure	12th Intro to Data Structure v2	12th Sets v1	12th Sets v2	12th -
13 Fall break	13th Fall break	13th Fall break	13th Fall break	13th -
14 Algorithmic analysis	14th Algorithmic analysis v1	14th Algorithmic analysis v2	14th Algorithmic analysis v3	14th -
15 Reviews	15th Reviews	15th Reviews	15th Final exams	15th -