Syllabus

CMPSC 100
Computational Expression
Summer 2020

Syllabus

Course Instructors

Prof. Janyl JUMADINOVA  
Email: jjumadinova@allegheny.edu  
Website: http://www.cs.allegheny.edu/sites/jjamadinova/

Prof. Doug LUMAN  
Email: dluman@allegheny.edu  
Website: http://www.cs.allegheny.edu/sites/dluman/

Instructors’ Office Hours

Instructors are available for office hours by appointment. To make an appointment, a student can send a request via course’s Slack workspace or via instructors’ emails. An online office hour meeting with one of the instructors will be scheduled within 24 hours of the request.

Course Meeting Schedule

This course meets online for 5 weeks of instruction, starting on June 23rd and ending on July 28th.

Discussion, Group Work and Practical session: Monday, Tuesday, Wednesday, Thursday, 7pm to 8:30pm IST (9:30am to 11:00am EST)
Laboratory session: Friday, 7pm to 8:30pm IST (9:30am to 11:00am EST)

Course Resources

Course Web page: https://cs.allegheny.edu/sites/jjamadinova/teaching/100
You can find the most up-to-date schedule of the course and the required readings on the course’s page.

Slack channel: https://cmpsc100summer2020.slack.com
Slack will be used for discussion and communication during and outside of the class time.

Github Organization: https://github.com/allegheny-college-cmpsc-100-summer-2020
Github, a cloud based system, will be used for sharing course materials by the instructor and for submitting assignments by the students.

JupyterHub: https://www.cs.allegheny.edu/jupyter
Allegheny’s JupyterHub server will be used for interactive programming and computing, eliminating the need to install software on students’ local machines.

Academic Bulletin Course Description

An introduction to the principles of computer science with an emphasis on creative expression through the medium of a programming language. Participating in hands-
on activities that often require teamwork, students learn the computational structures needed to solve problems and produce artifacts such as interactive games and computer-mediated art and music. Students also learn how to organize and document a program's source code so that it effectively communicates with the intended users and maintainers. Additionally, the introduction includes an overview of the discipline of computer science and computational thinking. During a weekly laboratory session students use state-of-the-art technology to complete projects, reporting on their results through both written reports and oral presentations.

Prerequisite: Knowledge of elementary algebra.
Distribution Requirements: ME, SP.

Course Objectives

The process of implementing and evaluating correct and efficient software involves the application of many interesting theories, techniques, and tools. In addition to learning problem solving and computational thinking skills, this class will teach students how to use, design, implement, and test software developed in a popular programming language.

This course will also demonstrate that computer science can exist outside of theorems, laws, or "rules" and, instead, serve as just another tool by which to express ideas – some of which happen to be technical in service or nature. No different than oil paints, paint brushes, a pack of playing cards, or elaborate vanishing booths, computer code can be used as a tool to express artistic or even impractical ideas via computational thinking, a set of thought processes which we are often unaware that we exercise on a daily basis.

Instead of only relating technical details (of which there are many), this course explores ways to ideate and realize concepts on the path to achieving basic fluency in a new language, enabling abstract ideas to become digital reality.

Learning Objectives

At the completion of this course, students will be able to:

- Develop and realize moderately complex ideas in a digital medium.
- Understand the basic language-independent principles of object-oriented programming (OOP).
- Express basic fluency in the Python programming language.
- Exercise an intermediate grasp of industry-standard tools used to build software (e.g. git, GitHub, et al.).
- Learn to adapt to the important, but often accidental, aspects of creating programs in Python.
- Demonstrate the ability to communicate and collaborate as an integral part of the creative process.
Textbooks

To supplement instructor’s content and hands-on activities, we will assign required readings from the book titled “Think Python: How to Think Like a Computer Scientist”, by Allen B. Downey (2nd Edition). This book is available for free under the Creative Commons License at http://greenteapress.com/thinkpython/html/index.html.

Supplemental readings maybe assigned from “A Byte of Python” by Swaroop Chitlur, which is also available for free under the Creative Commons License at https://python.swaroopch.com/.

Class Policies

Grading

The grade that a student receives in this class will be based on the following categories. All percentages are approximate and it is possible for the assigned percentages to be changed during the course, if a need arises to do so.

<table>
<thead>
<tr>
<th>Category</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Participation</td>
<td>25%</td>
</tr>
<tr>
<td>Laboratory Assignments</td>
<td>35%</td>
</tr>
<tr>
<td>Quizzes</td>
<td>20%</td>
</tr>
<tr>
<td>Course Project</td>
<td>20%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

These grading categories have the following definitions:

- **Class Participation**: All students are required to actively participate during class sessions. Your participation will take forms of active engagement in class activities, asking or answering questions, or involvement in the course’s Slack channel discussions.

- **Laboratory Assignments**: Lab assignments invite students to explore different techniques for designing, implementing, evaluating, and documenting software solutions to challenging problems that often have a connection to real-world concerns.

- **Quizzes**: Each Wednesday, the instructors will release a quiz that assesses the key concepts and terminology of that week. Unless prior arrangements are made with the course instructors, all quizzes have to be completed by all students before the class session on Thursday.

- **Course Project**: This project offers students the opportunity to develop a realized version of an reasonably complex, discipline-appropriate project. Completion of this project will require use of the knowledge accumulated throughout the course. This task will require incorporation of a wide variety of concepts, and will allow students to implement either a project of their chosen discipline/application or a project for an application designed by the instructors.

Attendance

It is mandatory for all students to attend all online class and laboratory sessions. If you will not be able to attend a session, then please notify the instructors as soon as possible. If you need to miss
class due to a religious observance, please speak to the instructors in advance to make appropriate arrangements.

Course Workflow and Schedule

Class sessions: During each class session on Monday through Thursday, students will be invited to participate in a class activity, which may involve discussion, reflection, programming practice, or a completion of a practical exercise using Jupyter worksheets. These activities are designed to give students a practice using the new concepts without being afraid to fail, and are graded only as ‘attempted’ or ‘not attempted’.

Laboratory sessions: A lab session will take place during each class session on Friday. During each lab session, with the guidance of the instructors, students will be invited to complete a hands-on assignment that incorporates concepts introduced during that week.

Quizzes: At the end of each class session on Wednesday, a short quiz will be released, to be completed an hour before the class session begins on Thursday. On Thursday, instructors will use the results of the quiz to provide additional instruction and reflection on the concepts that are in need of follow up.

An outline of the course schedule is included below. A more detailed schedule, including reading assignments and deadlines, is available on the course’s Website.

- **Week 01 (June 23-26)**: What is Computer Science? Computer Basics. Introduction to course’s tools.
  - During the first week, the instructors will introduce the course and its structure, and will delve into the world of computer science. All of the tools used in this course will be introduced and the students will be guided through the set up and configuration of these software tools. Students will then learn about and practice using the command line interface.

  - During this week, students will be introduced to the Python programming language and will learn how to write Python programs that perform simple tasks. Students will also engage with obtaining and incorporating user’s input into their programs interactively and learn how to design basic functions.

- **Week 03 (July 6-10)**: Data structures (sequences). Introduction to control structures and repetition. Input/Output.
  - During the third week, data structures, such as lists and tuples will be discussed. Students will learn about various types of control structures that can be used to design an algorithm and will practice using one type of a repetition structure, a while loop. Programming involving reading and writing to files will also be discussed and practiced.
• **Week 04 (July 13-17):** Data Structures (set, dictionary). Conditional statements and `for` statements. List Comprehensions.
  
  – During this week, instructors will discuss other data structures, including sets and dictionaries. Students will work with conditional statements and will learn an alternative repetition statement, called a `for` loop. A syntactic construct available in Python for creating a list will be discussed and demonstrated.

• **Week 05 (July 20-24):** Object-Oriented Programming (OOP). Classes. Methods.
  
  – During the fifth week, students will engage more deeply with the concepts surrounding object-oriented programming. Development of sophisticated programs consisting of multiple classes and methods will be demonstrated and practiced.

• **Week 06 (July 27-28):** Project work. Code walkthrough.

**Seeking Assistance**

Students who are struggling to understand the knowledge and skills developed in a class or a laboratory session or experience technical difficulties are encouraged to seek assistance from the course instructors. Students who need the course instructors’ assistance should request a meeting via course’s Slack channel or by email.

**Classroom Ethics**

The discipline of computer science, like many others, encourages its members to act according to discipline-specific ethics. We encourage you to take time to review the Association for Computing Machinery (ACM) *Code of Ethics.*

Allegheny College also expects students and faculty to act according to its Statement of Community:

Allegheny students and employees are committed to creating an inclusive, respectful and safe residential learning community that will actively confront and challenge racism, sexism, heterosexism, religious bigotry, and other forms of harassment and discrimination. We encourage individual growth by promoting a free exchange of ideas in a setting that values diversity, trust and equality. So that the right of all to participate in a shared learning experience is upheld, Allegheny affirms its commitment to the principles of freedom of speech and inquiry, while at the same time fostering responsibility and accountability in the exercise of these freedoms. This statement does not replace existing personnel policies and codes of conduct.

Keep both of these standards in mind as you exercise your academic inquiry in this course. These serve as our “fundamental first principles” in pursuit of our shared academic goals.
Honor Code

All students enrolled at Allegheny College are bound by the Honor Code. It is expected that your behaviour will reflect that commitment. To this end, we expect that you will adhere to the following Department Policy:

Department of Computer Science Honor Code Policy

It is recognized that an important part of the learning process in any course, and particularly in computer science, derives from thoughtful discussions with teachers and fellow students. Such dialogue is encouraged. However, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others, and the student who produces assignments that are identical to, or merely variations on, someone else’s work. It will therefore be understood that all assignments submitted to faculty of the Department of Computer Science are to be the original work of the student submitting the assignment, and should be signed in accordance with the provisions of the Honor Code. Appropriate action will be taken when assignments give evidence that they were derived from the work of others.