

# Genesee Community College

## Syllabus – CIS127 – Computing in Math, Science and Engineering

### Spring 2020

**Instructor:** Ken Mead

**Email:** [kjmead@genesee.edu](mailto:kjmead@genesee.edu)

**Office location:** D395 Math Science Area, Building D, Main Campus

**Office hours:** To Be Announced.

**Phone:** 585-343-0055 x 6381

**Homepage:** <https://faculty.genesee.edu/kjmead>

**Catalog description:** An introduction to computer programming and problem solving, with special emphasis on problems found in mathematics, science, and engineering. Students will use software such as Maple and Excel, and/or a programming environment such as MatLab or Python, to efficiently and effectively solve problems by developing a strategy, applying appropriate techniques, and validating results.

**Prerequisite:** MAT140 or higher.

#### **Course Learning Outcomes (CLOs):**

Upon successful completion of this course, the student will be able to:

1. Determine the best tool to use to solve various mathematical, scientific, or engineering problems.
2. Use a variety of software packages (spreadsheet, MATLAB, octave, python shell, etc.) to solve mathematical, scientific, or engineering problems.
3. Use various programming languages (python, MATLAB, etc.) to solve mathematical, scientific, or engineering problems.
4. Use variables, functions, or advanced data structures to effectively solve problems as needed.
5. Solve problems using decision making blocks.
6. Solve problems using looping structures.
7. Solve problems using advanced data structures (lists, arrays, dictionaries, etc.).
8. Use multi-dimensional arrays and matrices to model and solve various problems as needed.

## **Course Overview**

### Part 1 of Course – Python

Python is an easy to learn programming language that is used extensively in mathematics, science and engineering. It is open-source and completely free to use. There are two versions of Python in heavy use today: python 2 and python 3. We'll focus on python 3.

Lesson 0: Installing Python. We'll be using the Thonny and Anaconda Python 3 distribution on PC and the standard python3 distribution on the [csnlinux.genesee.edu](http://csnlinux.genesee.edu) server.

Lesson 1: Why we Program – in this lesson, we'll look at the big picture as to why we write computer programs, and why python is a good tool to use.

Lesson 2: Variables and Expressions - in this lesson, we cover how a program uses the computer's memory to store, retrieve and calculate information.

Lesson 3: Conditionals – in this lesson, we move from sequential code that simply runs one line of code after another to conditional code where some steps are skipped.

Lesson 4: Functions – in this lesson, we will learn about what functions are and how we can use them to solve problems.

Lesson 5: Loops and Iteration – in this lesson, we examine loops and iteration. Loops are the way we tell Python to do something over and over. Loops are the way we build programs that stay with a problem until the problem is solved.

Lesson 6: Strings – in this lesson, we look at how Python stores and manipulates textual data using string variables and functions.

Lesson 7: Files – in this lesson, we learn how to open data files on your computer and read through the files using Python.

Lesson 8: Lists – in this lesson, we look at Python's simplest data structure - the list. Lists can store more than one item in a variable.

Lesson 9: Dictionaries – in this lesson, we look at dictionaries. The dictionary data structure allows us to store multiple values in an object and look up the value given a specific key.

Lesson 10: (time permitting) Tuples – in this lesson, we look at tuples. The tuple is a Python data structure that is like a simple and efficient list.

Lesson 11 (time permitting): Object Oriented Python – in this lesson, we take a quick look at how Python supports the Object-Oriented programming paradigm.

## Part 2 of Course – MATLAB / Octave

MATLAB is a popular programming language and computational toolbox used by engineers and scientists. Octave is open source software that strives to match MATLAB's capabilities feature by feature. In this course, we'll use Octave, but students can optionally use MATLAB if they wish. In the following descriptions, we'll use MATLAB and Octave interchangeably.

Lesson 12: Introduction to MATLAB – in this lesson, we'll familiarize ourselves with the Matlab user interface.

Lesson 13: Matrices and Operators – in this lesson, we'll learn about the basic unit with which we work in Matlab – the matrix, along with many of the operators to perform arithmetic on these matrices.

Lesson 14: Functions – in this lesson, we'll learn how to create Matlab functions so we can break complex problems into manageable and reusable components.

Lesson 15: The MatLab Toolbox – in this lesson, we'll learn about a bunch of Matlab's built in functions. We'll also learn about polymorphism, random numbers, how to get input from the keyboard, and how to plot graphs.

Lesson 16 (time permitting): Conditionals – in this lesson, we will learn how to make Matlab make decisions using the if statement. We will also examine relational operators and logical operators.

## Required materials

**Access to an internet-enabled computer with Python 3 and Octave software installed.**

**Python for Everybody – Exploring Data in Python 3.** Author: Charles Severance. Free downloadable PDF at <http://www.py4e.org/book>

### Optional materials:

**Computer Programming with MATLAB, first revised PDF edition**, June 2015. Authors: Fitzpatrick and Ledeczi. Download for \$10 at <http://cs103.net/buy/>

**Introduction to Python for Computational Science and Engineering (A beginner's guide to Python 3)**, July 2017. Author: Hans Fangohr, Faculty of Engineering and the Environment, University of Southampton. United Kingdom. Free downloadable PDF at <https://github.com/fangohr/introduction-to-python-for-computational-science-and-engineering/blob/master/Readme.md>

## Proctored Testing

Students in this class will take exams in the GCC testing center within 48 hours either way of the dates listed below. Please inform your instructor the date and time you wish to take the test.

## Grading

Final grades are assigned according to the following scheme, with the final average rounded to the nearest integer (in %): 92 or higher = A, 90-91 = A-, 88-89 = B+, 82-87 = B, 80-81 = B-, 78-79 = C+, 72-77 = C, 70-71 = C-, 68-69 = D+, 62-67 = D, 60-61 = D-, 59 or less = F.

Grades will be weighted as follows:

- 50% - **three exams on 2/19, 4/1, 5/6**
- 50% - Programming projects and homework assignments. Homework assignments will collectively count as a single project grade.

Breakdown of Projects and Assignments:

There will be many short homework assignments that must be completed throughout the semester. Generally speaking, these will be graded as either complete or not, all or nothing. However, there may be exceptions. You should complete homework assignments as soon as possible after they are assigned. The deadline for submitting homework assignments for full credit will be the end of the day of the next scheduled exam.\* After this date, the homework score will be reduced by 50%.

There will also be between 4 and 5 larger programming projects (most likely written in Python and/or MATLAB) which must be completed to specifications provided by the instructor. These larger projects will be mathematical, scientific, or engineering focused in nature. Projects must be submitted on time or will be assessed a penalty of 10% if submitted within a week of the due date, 20% if submitted

between 8-14 days late, 50% if submitted later than 2 weeks. If a project is not submitted at all, you will receive a grade of zero (0) AND will have 2 points deducted from your final average!

## Student Responsibilities

- View the class videos as soon as possible. All are posted.
- Take your exams! If you must miss an exam due to extenuating and unavoidable reasons, you need to contact your instructor beforehand to re-schedule. Missing an exam without permission and a good reason will result in a grade of zero. If you are taking exams at a campus center or some other locale, please contact your instructor with your plans at least 10 days before the scheduled date.
- Complete homeworks and projects on time! Late penalties will apply. See above.
- Stay engaged! Come up with ideas for problems to solve! When viewing the videos, make sure you solve any problems as you view. Enjoy the material because it's really good stuff!!!

## Plagiarism and Cheating

***Cheating*** is obtaining or intentionally giving unauthorized information to create an unfair advantage in an examination, assignment, or classroom situation. ***Plagiarism*** is the act of presenting and claiming words, ideas, data, programming code or creations of others as one's own. Plagiarism may be intentional – as in a false claim of authorship – or unintentional – as in a failure to document information sources using MLA (Modern Language Association), APA (American Psychological Association) Chicago or other style sheets or manuals adopted by instructors at the College. Presenting ideas in the exact or near exact wording as found in source material constitutes plagiarism, as does patching together paraphrased statements without in-text citation. The purchasing or sharing of papers or projects between students or the re-use of papers or projects submitted for more than one assignment or class also constitutes plagiarism.

In short - **do your own work!** This means no cutting and pasting code from the internet. Students are encouraged to discuss programming assignments and projects, and even develop overall strategies for solving problems together, but must ultimately write their own code. Any evidence of plagiarism, or of cheating or sharing answers on a test, will result in a grade of zero for the assignment, project or test. Multiple offenses will result in a grade of F for the course.

## Accessibility Statement

If you have a physical, psychological, medical or learning disability that may impact your course work or participation in this class, please contact the Assistant Dean of Student Services/Disabilities Coordinator, Success Coach, or Academic Advisor who will arrange an intake meeting. The Assistant Dean/Coordinator will determine with you what accommodations are necessary, appropriate and reasonable. All information and documentation is confidential.

## General Schedule for the Semester (subject to change)

- Week 1 - Introductions, installing software and accessing csnlunix, introduction to Python, using the python shell, writing a python program.
- Week 2 - Variables, mathematical operators, built-in functions, conditionals.
- Week 3 - Conditionals continued.
- Week 4 - User-defined functions, returning values, loops.
- Week 5 - Loops, review for test.
- Week 6 - Test 1, exceptions, range(),
- Week 7 - Randomness, monte carlo simulations, nested loops.
- Week 8 - String processing and functions.
- Week 9 - Lists.
- Week 10 - Lists, review for test, Test 2.
- Week 11 - Lists, numpy, and matplotlib, dictionaries.
- Week 12 - Dictionaries and tuples, intro to MATLAB.
- Week 13 - Matrices, MATLAB, and MATLAB arithmetic operators.
- Week 14 - MATLAB functions and conditionals.
- Week 15 - MATLAB odds and ends, review for test.
- Week 16 - Test 3, finish semester.

*The instructor reserves the right to make any reasonable and necessary modifications to the statements above. This document is subject to change.*