CS 115: Programming I – Fall 2015

Links: [Course Home] [Assignments] [Schedule] [Resources] [Programming Tools] [Moodle] [CS 115W]

Catalog description

(4 units) Lecture, 3 hours; laboratory, 3 hours. An overview of computer organization; arithmetic and logical expressions, decision and iteration, simple I/O; subprograms; principles of good programming style, readability, documentation, structured programming concepts; top-down design and refinements; techniques of debugging and testing. Use of the above concepts will be implemented in a standard high-level programming language.

This course is currently taught using Python.

Prerequisites

GE Math eligibility (satisfaction of ELM requirement) and English eligibility (satisfaction of EPT requirement).

Students who do not meet these prerequisites will be dropped from the class unless they obtain instructor consent.

Textbooks

There are two textbooks, both required. One is online and free. The other one is reasonably priced (about $30), and you will need it for the graphical programs we'll be writing.

- Brad Miller and David Ranum, *How to Think Like a Computer Scientist*, 2nd interactive edition: http://interactivepython.org/runestone/static/thinkcspy/index.html
- John Zelle, *Python Programming: An Introduction to Computer Science*, 2nd edition (ISBN: 9781590282410)

Instructor Contact Info

Gurman gill

Email:	gurman.gill@sonoma.edu
Office phone:	707-664-2667
Office location:	Darwin 116G

Meeting Times

MySSU should show that you are enrolled in two consecutive section numbers (for example, sections 3 and 4). The first number is your lecture section. The second number is your lab section.

You are expected to attend your lecture section twice a week, and your lab section once a week, as

shown below:

Lectures

Section	Time	Room	Instructor
1 and 3	Tu/Th 9:20–10:35 AM	Darwin 37	Gurman Gill
5 and 7	Tu/Th 10:45 AM–noon	Darwin 107	Gurman Gill

Labs

Section	Time	Room	Instructor	Student Assistants
2	Tu 5:00–7:50 PM	Darwin 25	Noah Melcon	Nik Rowen, Leah Headd
4	W 5:00–7:50 PM	Darwin 28	Noah Melcon	Sean Cullen, Leah Headd
6	Tu 5:00–7:50 PM	Darwin 28	Matt Hardwick	Tyler Holland, Sean Cullen
8	W 2–4:50 PM	Darwin 25	Kristi Yost	Tyler Holland, Sean Cullen

Extra help

Time	Type	Room	Instructor
M 4:00–7:00PM	Drop-in tutoring	Darwin 28	Tyler Holland
Tu 1:00–2:00PM	Drop-in tutoring	Darwin 28	Sean Cullen
Tu 2:00–3:00 PM	Office hours	Darwin 116G	Gurman Gill
Th 2:00–3:50 PM	Workshop	Darwin 28	Tyler Holland, Nik Rowen
Th 4:00–5:00PM	Drop-in tutoring	Darwin 28	Tyler Holland
F 10:00 AM–noon	Drop-in tutoring	Darwin 28	Sean Cullen
F noon–2:00PM	Drop-in tutoring	Darwin 28	Tyler Holland
F 2:00–3:00 PM	Office hours	Darwin 116G	Gurman Gill

GE Information

This class satisfies the General Education category B3 requirement (Specific Emphasis in Natural Sciences). As of May 2008, the GE B3 objectives are

1. Improve understanding of the concepts and theories of science and technology
2. Understand the interconnected and ever-changing relationships among the natural, physical, and technological sciences
3. Critically assess the social and ethical implications of science and technology in relations to their daily lives
4. Improve problem solving and critical thinking skills through application of scientific knowledge using hands-on activities

More information can be found on the SSU GE homepage.

Course Goals and Objectives

1. Develop algorithms for solving problems.
    - Write programs that interact with the user textually and graphically.
    - Write programs that perform calculations using arithmetic expressions.
    - Write programs that choose which actions to perform using control structures.
    - Apply, program, and evaluate common algorithms for searching and sorting data.
2. Use design strategies for managing complexity.
    - Design programs that are readable and maintainable.
    - Decompose a complex problem into smaller parts through the use and design of functions.
    - Organize data using standard data structures (e.g. lists, strings, files).
    - Design and use specialized data types (object-oriented programming).

Online Resources

Website

The course homepage is http://gill.cs.sonoma.edu/cs115/. There, you will find links to the following:

- The course schedule, including exam dates and links to all active assignments
- The course resources page, which has links to old exams, the textbook's website, and other helpful resources
- Instructions for installing programming tools on your computer
- Descriptions of all course activities and the policies associated with them

Moodle Gradebook

The course gradebook will be kept on Moodle so that you can check your grades at any time.

Email List

Course announcements will be sent to your SSU email address, so you should check that account frequently.

Coursework and Grading

Course Activities

The assignments page (http://gill.cs.sonoma.edu/cs115/assignments.html) describes the different course activities and the policies on collaboration and late work.

Grading Policies

Grade breakdown:

Exams	45%
Programming projects	30%
Labs	20%
Lecture activities and quizzes	5%

You must also pass each of these three components individually (60% or better) to receive a C- or better in the course. In other words, if you fail one or more of these three components, the maximum grade you can earn in the course is a D+.

Your final semester grade will be rounded to the nearest integer.

Grading scale (after rounding)

93-100%	90-92%	87-89%	83-86%	80-82%	77-79%	73-76%	70-72%	67-69%	63-66%	60-62%	Below 60%
A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F

CS majors must take this course for a letter grade.

Up to 3% may be added to your final grade at the instructor's discretion for constructive participation in the class. Constructive participation includes in-class participation; asking good questions via email or during office hours; and doing outstanding or extra work on assignments. No other adjustments of borderline grades will be considered.

Penalties for Academic Dishonesty

Academic dishonesty will be severely penalized; at a *minimum*, you will receive a grade of 0 on the assignment. For more information, see SSU's cheating and plagiarism policy (http://www.sonoma.edu/UAffairs/policies/cheating_plagiarism.htm) and the Dispute Resolution Board website (http://www.sonoma.edu/senate/drb/drb.html).

University Policies

Disability Accommodations

If you are a student with a disability and you think you may require accommodations, please register with the campus office of Disability Services for Students (DSS), located in Salazar Hall - Room 1049, Phone: (707) 664-2677, TTY/TDD: (707) 664-2958. DSS will provide you with written confirmation of your verified disability and authorize recommended accommodations. This authorization must be presented to the instructor before any accommodations can be made. Visit http://www.sonoma.edu/dss for more information.

Other University Policies

There are important University policies that you should be aware of, such as the add/drop policy,

cheating and plagiarism policy, grade appeal procedures, accommodations for students with disabilities, and the diversity vision statement. Go to this URL to find them: http://www.sonoma.edu/uaffairs/policies/studentinfo.shtml.

Consolidated Syllabus (last updated Aug. 24, 2015)

You may download the course information, assignment descriptions, and schedule in a consolidated pdf:
http://gill.cs.sonoma.edu/cs115/syllabus_consolidated.pdf

CS 115: Programming I Assignments

Lecture and Readings

You are responsible for all material presented in lecture and are expected to get the notes from another student if absent.

You are also expected to spend additional time outside of class to master the lecture material. Reading assignments and learning objectives are posted for each week. You should read as much or as little as necessary to meet the learning objectives.

You can expect frequent small assignments or quizzes in lecture. You should bring at least a 4" by 6" notecard or a half-sheet of paper to each lecture for this purpose. Your lowest three grades on these small assignments will be dropped.

Labs

The weekly lab allows you to practice programming with immediate feedback from the instructor and other students. Lab attendance is expected. Each week's lab assignment will be posted online and linked from the course schedule and Moodle.

Unless otherwise specified, each lab assignment has several different components, which total 40 points.

Note: Grades on the labs tend to be high, but you should be sure to clear up any mistakes you make. The course material is cumulative, and small misunderstandings can add up to big mistakes over the course of the semester.

Lab Writeup

This is a Moodle "quiz" to teach and evaluate your understanding of the week's lecture. You should open the lab writeup in Moodle as soon as you sit down to do the lab. As you work through the lab, some of the lab instructions will ask you to go to your writeup and answer a question to check your understanding. There will also be some open-ended questions that the course staff will grade by hand. You can answer each question as many times as you like without penalty.

Lab Demo

Most weeks, you will be asked to demo one or more working programs to the course staff. We will make sure your program works correctly on a set of basic tests.

Demos must be done in person. They will be accepted in any lab section, workshop, or tutoring hours up to the deadline (see the top of the Moodle page for the schedule). Due to space constraints, they will NOT be accepted during office hours.

Submitted Lab Code

Most weeks, you will also be asked to submit one program for more thorough testing and grading. The program will need to be uploaded in Moodle and each lab will carry instructions for the same.

You are expected to have a docstring in each submitted program that contains your name, the lab number, and a brief description of what your program does.

Lab deadlines and points

Component	Posted by	Due by	Points (out of 40)	Late policy
Lab writeup	Mon 5 PM	The following Tu, 4 PM (online)	20	Half credit up to one week late
Lab demo		Lab, tutoring, or workshop by the following Tu, 4 PM (see Moodle for times)	10	
Lab code		The following Tu, 4 PM (online)	10	

Lab Collaboration Policy

You are encouraged to work with other students in the labs. However, you must turn in all lab work individually, and you must be able to explain and stand by all of the work you turn in for credit.

Programming Projects

Programming projects require you to take the skills you've learned in lecture and lab and apply them at a larger scale. They are a rewarding and challenging part of the course, and they require perseverance and time management skills. Projects must be one student's individual work, and the collaboration policy will be strictly enforced.

The programming projects are larger assignments, but they will usually be broken into pieces as follows:

- **Week 1:** An in-person demo during lab, workshop, or tutoring
- **Week 2:** Another in-person demo (for longer projects)
- **Week 2 or 3:** Final code submission online, to be graded for correctness and style

There is an automatic 48-hour grace period associated with the final code submission deadline. No other extensions will be granted.

Project deadlines and points

Component	Due by	Late policy
Demo(s)	In person; last chance is Thu workshop	No late credit
Final project code	Online; see project instructions	Up to 48 hours with no penalty

Programming Project Collaboration Policy

Programming projects must be your own work unless otherwise stated, and academic misconduct is taken very seriously. You may discuss ideas and approaches with other students and the course staff, but you should work out all details and write up all solutions on your own. **The following actions will be penalized as academic dishonesty:**

- Copying part or all of another student's assignment
- Copying old or published solutions
- Looking at another student's code or discussing it in great detail. You will be penalized if your program matches another student's program too closely.
- Showing your code or describing your code in great detail to *anyone*, in person or online, other than the course staff and lab volunteers.

Exams

The exams cover the material from lecture, lab, and the textbook. Exams are cumulative, although they will emphasize recent material.

You may bring one 8.5 by 11-inch handwritten sheet of notes (front and back). This is the only resource that you may consult, other than your brain.

Makeup exams will be given only in extraordinary circumstances, and the instructor MUST be notified in advance.

CS 115: Programming I – Fall 2015 Course Schedule

Links: [Course Home] [Assignments] [Schedule] [Resources] [Programming Tools] [Moodle] [CS 115W]

Except for exam dates, all schedule information is tentative and subject to change.

This schedule is online at http://cs.sonoma.edu/~gill/cs115/schedule.html. The online version will be updated throughout the semester with links to reading assignments, labs, and projects.

	Tuesday	Lab	Thursday	Due
Week 1 Aug 24– Aug 28	Intro to course and programming *Week 1 reading*	Lab 1	Input and output; Variables	
Week 2 Aug 31– Sep 04	Data types and arithmetic *Week 2 reading*	Lab 2	Definite (*for*) loops *Project 1 assigned*	Tu 4 PM: Lab 1
Week 3 Sep 07– Sep 11	Accumulations Intro to graphics *Week 3 reading*	Lab 3	If-statements	Tu 4 PM: Lab 2
Week 4 Sep 14– Sep 18	More conditional statements Graphics: drawing rectangles; handling mouse clicks *Week 4 reading*	Lab 4	Nested loops	Tu 4 PM: Lab 3 Thu 4 PM: Proj. 1-A demo
Week 5 Sep 21– Sep 25	Problem solving: loops and if-statements Graphics: drawing lines *Week 5 reading*	Lab 5	**Exam 1**	Tu 4 PM: Lab 4
Week 6 Sep 28– Oct 02	Indefinite (*while*) loops *Week 6 reading*	Lab 6	Loop review Intro to strings	Tu 4 PM: Lab 5 Tu 11 PM: Proj. 1 final code
Week 7 Oct 05– Oct 09	Intro to lists *Week 7 reading* *Project 2 assigned*	Lab 7	Common list patterns	Tu 4 PM: Lab 6
Week 8 Oct 12– Oct 16	List review; function intro *Week 8 reading*	Lab 8	Function parameters and return values	Tu 4 PM: Lab 7 Thu 4 PM: Proj. 2-A demo

Week 9 Oct 19– Oct 23	Nested lists; Refactoring *Week 9 reading*	Lab 9	Function wrapup	Tu 4 PM: Lab 8 Thu 4 PM: Proj. 2-B demo
Week 10 Oct 26– Oct 30	Review class (bring your questions!)	Catchup	**Exam 2**	
Week 11 Nov 02– Nov 06	File processing; search algorithms *Week 11 reading*	Lab 10	Sorting and searching algorithms	Tu 4 PM: Lab 9 Tu 11 PM: Proj. 2 final code
Week 12 Nov 09– Nov 13	Sorting algorithms *Week 12 reading* *Project 3 assigned*	Lab 11	Object-oriented programming (OOP)	Tu 4 PM: Lab 10
Week 13 Nov 16– Nov 20	OOP *Week 13 reading*	Lab 12	OOP	Tu 4 PM: Lab 11 Thu 4 PM: Proj 3-A demo
Week 14 Nov 23– Nov 27	OOP *Week 14 reading*	Catchup	*Happy Thanksgiving - No class!*	
Week 15 Nov 30– Dec 04	Recursion *Week 15 reading*	Lab 13	Recursion	Tu 4 PM: Lab 12 Thu 4 PM: Proj 3-B demo
Week 16 Dec 07– Dec 11	Review class (bring your questions!)	Lab 14	Review and catchup	Tu 4 PM: Lab 13 Tu 11 PM: Proj 3 final code
Finals Dec 14– Dec 18	**EXAM 3** **Tue. 11:00 AM–12:50 PM, Darwin 29 (Sections 5–8)** **-- OR --** **Thu. 8:00 AM–9:50 AM, Darwin 37 (Sections 1–4)**			Tu 4 PM: Lab 14 due

CS 115 Student Resources

Links: [Course Home] [Assignments] [Schedule] [Resources] [Programming Tools] [Moodle] [CS 115W]

Python Practice and References

- Website for the course textbook with supplemental material AND information about typos and errors.
- Online Python Tutor (be sure to select Python 3.3 from the drop-down menus)
- Official Python 3 documentation, tutorials, and more
- CodingBat online programming exercises
- Michigan State's archive of Python projects
- PEP 8 Python style guide

Exam File

The number of exams and the distribution of topics varies from semester to semester. Contact the instructor if you have any questions about the content covered on this semester's exams.

Python Exams

Exam 1

- Spring 2014 [Form A] [Form B]
- Spring 2013 [Form A] [Form B]
- Fall 2012 [Form A] [Form B]
- Spring 2012 [Exam]
- Fall 2011 [Form A] [Form B]

Exam 2

- Spring 2014 [Exam]
- Spring 2013 [Form A] [Form B]
- Fall 2012 [Form A] [Form B]
- Spring 2012 [Exam]
- Fall 2011 [Exam]

Exam 3

- Spring 2014 [Form A] [Form B]
- Spring 2013 [Form A] [Form B]
- Fall 2012 [Form A] [Form B]
- Spring 2012 [Exam]
- Fall 2011 [Exam]

Exam 4

- Fall 2012 [Exam]

- Spring 2012 [Exam]

C++ Exams

The exams below were given when the course was taught in C++ and not Python, but they should still help you understand the format and style of CS 115 exams.

Exam 1

- Spring 2011 [Exam]
- Spring 2010 [Review Quiz] [Exam]
- Fall 2009 [Review Quiz] [Exam]
- Spring 2009 [Review Quiz] [Exam]
- Fall 2008 [Review Quiz] [Exam Form A] [Exam Form B]

Exam 2

- Spring 2011 [Exam Form A] [Exam Form B]
- Spring 2010 [Review Quiz] [Exam]
- Fall 2009 [Review Quiz] [Exam]
- Spring 2009 [Review Quiz] [Exam]
- Fall 2008 [Review Quiz] [Exam Form A] [Exam Form B]

Exam 3

- Spring 2011 [Exam]
- Spring 2010 [Review Quiz] [Exam]
- Fall 2009 [Review Quiz] [Exam]
- Spring 2009 [Review Quiz] [Exam]
- Fall 2008 [Review Quiz] [Exam]

SSU campus resources

- SSU Moodle
- Moodle training for students (only available from computers on campus)
- CS advising information
- CS colloquium schedule (Thursdays at noon in Salazar 2016).
- CS Club
- Women in Computer Science (WICS) Club

CS 115 - Tools for Working from Home

Links: [Course Home] [Assignments] [Schedule] [Resources] [Programming Tools] [Moodle] [CS 115W]

Introduction

You are encouraged to install programming tools on your home computer. All of the tools recommended below are absolutely free to use. Just remember that your programs will be graded based on their behavior on our server (cwolf), so it's a good idea to periodically copy your code to cwolf and test it there. (Plus, we probably back up more frequently than you do.)

If you want to work on your own computer, you will need software to accomplish the following tasks:

- Remotely log into cwolf (the CS department server)
- Copy files between cwolf and your home computer
- Edit and run Python 3 programs

The next few sections of this document describe the software you will need for each task. Follow the instructions for your operating system (e.g. Windows, Mac, Linux). If you have any trouble with these instructions, you are welcome to bring your computer to office hours, workshop, or lab for extra help.

Grahpics Package

Here is the graphics package that we use in this course. You need to download it and rename it to "graphics.py" before you use it.

Remotely logging into cwolf

Mac or Linux

1. Open Terminal (which comes with your computer).

2. In the Terminal window, type the following command, with your actual CS department username filled in as indicated:

   ```
   ssh yourusername@cwolf.cs.sonoma.edu
   ```

3. The first time you run this command, you may see the following warning:

   ```
   The authenticity of host 'cwolf.cs.sonoma.edu (130.157.166.29)' can't be established.
   RSA key fingerprint is 3d:88:46:69:53:27:d6:9d:7c:ba:b7:f5:bd:1b:88:2a.
   Are you sure you want to continue connecting (yes/no)?
   ```

   Type `yes` to continue.

4. When prompted, type your CS department password. You will not see anything on the screen

-- not even ***** -- as you type.

Windows

1. You will need to download a program called PuTTY. Download PuTTY from this page by following the first link to PuTTY.exe.
2. Fill in `cwolf.cs.sonoma.edu` as the host name and 22 as the port.
3. Use your CS department user ID and password to log in.

Copying files between cwolf and your home computer

Your best bet is to install a graphical FTP client such as FileZilla Client, which works for Windows, Mac, and UNIX/Linux.

1. Follow the link to download FileZilla Client and install the version that is appropriate for your operating system.
2. To configure FileZilla (or any FTP client):
   - *Host:* `cwolf.cs.sonoma.edu`
   - *Protocol/Servertype:* SFTP or "FTP over SSH"
   - *Port:* 22
   - *Username:* Your cwolf (not university) username
3. You can now drag and drop your files between your computer (shown on the left) and cwolf (shown on the right). Lab 1C has more detailed instructions if you need them.

CAUTION! If you keep copies of your project files on multiple computers, you must be very careful not to forget which copy is newer and accidentally destroy your most recent work. You should *either* (1) keep the "official" copy on cwolf, and only download to your home computer to view/print, or (2) keep the "official" copy on your home computer, and only upload to cwolf to test/execute with our compiler before submitting your code.

Installing Python 3

First, you should get Python 3.3 from the official website. Scroll down to the "Download" section. The specific file to download depends on your operating system.

Once you have installed Python, you should be able to find a newly installed program called IDLE. This program allows you to edit and execute Python code. Be sure that you can open and run it.

Mac OS X

Click the Apple menu at the top-left corner of your screen then click About this Mac. The version of your OS will be displayed in a window that gets popped up.

Section 0.4 of the *Dive Into Python* textbook walks you through the installation.

Windows

Follow these directions to figure out if you have a 32-bit or 64-bit version of Windows. If you have a 64-bit version, you can download and run the Windows x86-64 MSI installer from the Download section. If you have a 32-bit version, or you are still not sure, use the Windows x86 MSI installer.

Section 0.3 of the *Dive Into Python* textbook walks you through the installation.

Mounting your cWolf disk on MacOS

Follow these instructions to mount your cWolf disk on the iMacs in Darwin 25 or Darwin 28.