CS 115: Programming I – Spring 2015

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

Suzanne Rivoire	
Email:	suzanne.rivoire@sonoma.edu
Website:	http://rivoire.cs.sonoma.edu
Office phone:	707-664-3337
Office location:	Darwin 116F

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	Suzanne Rivoire
5 and 7	Tu/Th 10:45 AM–noon	Darwin 29	Suzanne Rivoire

Labs

Section	Time	Room	Instructor	Student Assistants
2	Tu 4:00–6:50 PM	Darwin 25	Matt Hardwick	Kelly Morales, Kelsey Rangel
4	W 5:00–7:50 PM	Darwin 28	Noah Melcon	Kelsey Rangel, Helen San, Hayden Zollars
6	Tu 5:00–7:50 PM	Darwin 28	Noah Melcon	Marshall Jankovsky, Niklas Rowen
8	W 2–4:50 PM	Darwin 25	Brian McWilliams	Marshall Jankovsky, Kelly Morales

Extra help

Time	Type	Room	Instructor
M 12:30–3:30 PM	Drop-in tutoring	Darwin 28	Marshall Jankovsky
M 4:00–7:00PM	Drop-in tutoring	Darwin 28	Kelsey Rangel
Tu 2:15–3:45 PM	Office hours	Darwin 116F	Suzanne Rivoire
Tu 3:00–4:00PM	Drop-in tutoring	Darwin lobby	Kelsey Rangel
W 9:00–9:45 AM	Office hours	Darwin 116F	Suzanne Rivoire
Th 2:00–3:50 PM	Workshop	Darwin 28	Kelsey Rangel, Jeremy Swedroe
Th 4:00–5:00PM	Drop-in tutoring	Darwin 28	Kelsey Rangel
F 9:00 AM–noon	Drop-in tutoring	Darwin 28	Marshall Jankovsky

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

   - 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://rivoire.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://rivoire.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 Jan. 14, 2015)

You may download the course information, assignment descriptions, and schedule in a consolidated pdf:
http://rivoire.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.

Pre-Lab

This is a Moodle "quiz" to prepare you for the week's lectures and lab. To prepare, you will be expected to read a preview of the material for the week and a high-level description of the problem to be solved in the lab.

Pre-labs are not supposed to be high-stakes assignments. They're auto-graded, and you can take them as many times as you want without penalty. They are meant to help you get the most out of the lectures and labs.

Lab Writeup

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. Just like the pre-lab, 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. Each lab will provide instructions on how to do this.

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
Pre-lab	Th 9 AM	Tu 7 AM (online)	10	No late credit

Lab writeup	Mon 5 PM	The following Tu, 4 PM (online)	10	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 – Spring 2015 Course Schedule

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

This schedule is online at http://rivoire.cs.sonoma.edu/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 Jan 19–Jan 23	Intro to course and programming *Week 1 reading*	Pre-Lab 1 and Lab 1	Input and output; Variables	
Week 2 Jan 26–Jan 30	Data types and arithmetic	Lab 2	Definite (*for*) loops *Project 1 assigned*	Tu 7 AM: Pre-lab 2 Tu 4 PM: Lab 1
Week 3 Feb 02–Feb 06	Accumulations Intro to graphics	Lab 3	If-statements	Tu 7 AM: Pre-lab 3 Tu 4 PM: Lab 2
Week 4 Feb 09–Feb 13	More conditional statements Graphics: drawing rectangles; handling mouse clicks	Lab 4	Nested loops	Tu 7 AM: Pre-lab 4 Tu 4 PM: Lab 3 Thu 4 PM: Proj. 1-A demo
Week 5 Feb 16–Feb 20	Problem solving: loops and if-statements Graphics: drawing lines	Lab 5	**Exam 1**	Tu 7 AM: Pre-lab 5 Tu 4 PM: Lab 4
Week 6 Feb 23–Feb 27	Indefinite (*while*) loops	Lab 6	Loop review Intro to strings	Tu 7 AM: Pre-lab 6 Tu 4 PM: Lab 5 Tu 11 PM: Proj. 1 final code
Week 7 Mar 02–Mar 06	Intro to lists *Project 2 assigned*	Lab 7	Common list patterns	Tu 7 AM: Pre-lab 7 Tu 4 PM: Lab 6
Week 8 Mar 09–Mar 13	List review; function intro	Lab 8	Function parameters and return values	Tu 7 AM: Pre-lab 8 Tu 4 PM: Lab 7 Thu 4 PM: Proj. 2-A demo

Week 9 Mar 16–Mar 20	*Spring break - no class*			
Week 10 Mar 23–Mar 27	Refactoring; function practice	Lab 9	Function wrapup; nested lists	Tu 7 AM: Pre-lab 9 Tu 4 PM: Lab 8 Thu 4 PM: Proj. 2-B demo
Week 11 Mar 30–Apr 03	*Cesar Chavez Day -- no class*	Catchup	**Exam 2**	
Week 12 Apr 06–Apr 10	Nested lists; file processing	Lab 10	Search algorithms	Tu 7 AM: Pre-lab 10 Tu 4 PM: Lab 9 Tu 11 PM: Proj. 2 final code
Week 13 Apr 13–Apr 17	Search and sort algorithms *Project 3 assigned*	Lab 11	Sorting algorithms	Tu 7 AM: Pre-lab 11 Tu 4 PM: Lab 10
Week 14 Apr 20–Apr 24	Object-oriented programming (OOP)	Lab 12	OOP	Tu 7 AM: Pre-lab 12 Tu 4 PM: Lab 11 Thu 4 PM: Proj 3-A demo
Week 15 Apr 27–May 01	OOP	Lab 13	OOP	Tu 7 AM: Pre-lab 13 Tu 4 PM: Lab 12 Thu 4 PM: Proj 3-B demo
Week 16 May 04–May 08	Recursion	Lab 14	Review and catchup	Tu 7 AM: Pre-lab 14 Tu 4 PM: Lab 13 Tu 11 PM: Proj 3 final code
Finals May 11–May 15	**EXAM 3** **Mon. 10:45 AM–noon, Darwin 29 (Sections 5–8)** **-- OR --** **Thu. 8:00 AM–9:50 AM, Darwin 37 (Sections 1–4)**			Tu 4 PM: Lab 14 due