**Course Title**: *Computer Science I*  
Course Number: CIS *210*

# Course Description

This first course in the CIS introductory sequence covers fundamental approaches to computational problem solving (software development) and introduce other computer science topics. Computational concepts will be explored using the Python programming language.

# Prerequisites

* Math 112 and prior experience with a high level programming language such as Python, JavaScript, or Java.

# Instructor Contact Information

Prof Brittany Erickson  
Email: bae@uoregon.edu  
Office hours: Tues/Thurs 9-10am, and by appointment.

# Graduate Teaching Assistants

|  |  |
| --- | --- |
| Alexandre Chen Email: chern@cs.uoregon.edu Labs: Tues 3:30pm and Wed 3:30pm | Isaac Anderson Email: igeroni3@uoregon.edu Labs: Tues 2pm and Wed 12:30pm |
| Cody Rucker Email: crucker@cs.uoregon.edu Labs: Wed 11am and Wed 2pm |  |

# Undergraduate Learning Assistants

|  |  |
| --- | --- |
| Kalyn Koyanagi Email: kalynk@uoregon.edu  Labs: Wed 2pm Office hours: Thurs 8-10am, Thurs 12:30-1:30,  Fri 2-3:30pm | Brandon Dodd Email: bdodd@uoregon.edu  Labs: Wed 11am Office hours: Mon 12-2pm, Fri 12-1pm |
| River Veek Email: riverv@uoregon.edu  Labs: Tues 3:30pm Office hours: Tues 5-7pm, Thurs 4-6pm | Nick Johnstone Email: nsj@uoregon.edu  Labs: Tues 2pm  Office hours: Sun 4-6pm |
| Alex Anderson Email: Aanders8@uoregon.edu  Labs: Wed 12:30pm Office hours: Mon 8:30-10:30am, Mon 2-4pm, Wed 8:30-10:30am, Fri 8:30-10:30am | Ginni Gallagher  Email: ginnig@uoregon.edu  Labs: n/a  Office hours: Thurs 2:30-4pm, Fri 1-2pm |
| Brandon Bower Email: bbower@uoregon.edu  Labs: Wed: 3:30pm Office hours: Mon: 4-6pm |  |

# Office Hour Summary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Sun | Mon | Tue | Wed | Thu | Fri |
| 8-8:30am |  |  |  |  | Kalyn |  |
| 8:30-9am |  | A. Anderson |  | A. Anderson | Kalyn | A. Anderson |
| 9-10am |  | A. Anderson | Instructor | A. Anderson | Kalyn, Instructor | A. Anderson |
| 10-10:30am |  | A. Anderson |  | A. Anderson |  | A. Anderson |
| 10-11am |  |  |  |  |  |  |
| 11am-12pm |  |  |  |  |  |  |
| 12-12:30pm |  | B. Dodd |  |  |  | B. Dodd |
| 12:30-1pm |  | B. Dodd |  |  | Kalyn | B. Dodd |
| 1-1:30pm |  | B. Dodd |  |  | Kalyn | Ginni |
| 1:30-2pm |  | B. Dodd |  |  |  | Ginni |
| 2-2:30pm |  | A. Anderson |  |  |  | Kalyn |
| 2:30-3pm |  | A. Anderson |  |  | Ginni | Kalyn |
| 3-3:30pm |  | A. Anderson |  |  | Ginni | Kalyn |
| 3:30-4pm |  | A. Anderson |  |  | Ginni |  |
| 4-5pm | Nick | B. Bower |  |  | River |  |
| 5-6pm | Nick | B. Bower | River |  | River |  |
| 6-7pm |  |  | River |  |  |  |
| 7-8pm |  |  |  |  |  |  |

# Textbooks and Readings

(**Required**) Python Programming in Context, 3d ed., Miller and Ranum. Jones and Bartlett Learning, 2021. It is available for reserve in the Price Science Commons (Science Library). If you purchase a hard copy, a previously-used (2d. edition) copy is ok. Also: the textbook is available online to [ACM](https://www.acm.org/membership/membership-benefits) members through the ACM Learning Center. Student membership is $19 and you can access the textbook through the [O’Reilly learning platform](https://learning.acm.org/e-learning/oreilly) (you will need to enter your ACM credentials).

# Expected Learning Outcomes

# This course sets the foundation for the basic concepts and practices of computer science. The primary topics are familiarity with the Python programming language.

# Upon successful completion of the course, students will be able to:

* understand, develop, and implement algorithms for computational problem solving;
* use structured design and testing methods to develop and implement programs;
* read, write, revise, document, test, and debug code;
* demonstrate robust mental models of data representation and code execution;
* demonstrate good understanding of a high level programming language;
* introduce/implement a sampling of classic computer science problem domains and algorithms;

# Acquired Skills

Upon successful completion of the course, students will have acquired the following essential skills:

* be able to effectively use IDLE in the development and testing of moderate-sized Python programs;
* be able to understand, develop, and implement algorithms for computational problem solving;
* be able to read, write, revise, document, test, and debug code;
* and be able to demonstrate robust mental models of data representation and code execution;

# Estimated Student Workload

Each student will:

* read assigned readings in the required textbook;
* attend the live lectures delivered over Zoom;
* attend a weekly lab session, led by one of the graduate teaching assistants, at which you will be introduced to tools and techniques to help you be proficient in developing your projects;
* submit solutions for weekly programming projects – these must represent your own work;
* sit and submit the midterms and the final.

Workload expectations are 12-13 hours per week.

|  |  |  |
| --- | --- | --- |
| Activity | Total hours | Remarks |
| Attend lecture View recorded lectures | 30 | Each lecture is recorded and made available for subsequent viewing. |
| Assigned readings | 20 | Do the assigned readings ***BEFORE*** it is covered in lecture. |
| Attend labs | 10 | Lab attendance is mandatory – and you must attend the lab to which you are assigned. |
| Develop and submit projects | 50 | Start projects early; take advantage of plentiful office hours; use lab help hours to solicit additional assistance. |
| Revision | 15 | Studying for midterms and final |
| Total | 125 |  |

# Course Requirements and Grading

This REMOTE course will be taught entirely using Zoom and Canvas.

Grading will be based on the following criteria:

|  |  |
| --- | --- |
| 15% | programming projects |
| 5% | lab attendance (10 labs @ 0.5%) |
| 10% | code demo |
| 30% | midterms (two exams @ 15%) |
| 40% | final |

CIS majors and minors, and DSCI majors, must take CIS 210 graded; others may take it graded or P/NP.

## Grading rubric

|  |  |
| --- | --- |
| A | Excellent performance. Very well prepared to continue on in the major. Solid grasp of the concepts and approaches introduced in CIS 210 |
| B | Solid performance. You have shown a generally good grasp of the concepts and approaches introduced in CIS 210. |
| C | Acceptable performance, but your grasp of the concepts and approaches will require improvement if you choose to continue in the CIS major. |
| D | Your grasp of concepts and approaches, and/or programming skills, are not yet sufficient to continue in the CIS or DSCI major. |
| F | Little or no demonstrated grasp of concepts and approaches introduced in CIS 210, and/or failure to carry out much of the required work. |

## Grading Scale

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A+ >= 100 | B+ 86.7-89.9 | C+ 76.7-79.9 | D+ 66.7-66.9 | F 0.0-59.9 |
| A 93.4-99.9 | B 83.4-86.6 | C 73.4-76.6 | D 63.4-66.6 |  |
| A- 90.0-93.3 | B- 80.0-83.3 | C- 70.0-73.3 | D- 60.0-63.3 |  |

# Schedule

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lecture | Topic | Readings in Text | Page #s | | Projects and assignments | |
| 01 (1/5) | Intro to Python | Ch 1 | | n/a | | r |
| 02 (1/7) | Intro to Python (cont’d), algorithms |  | |  | | Project 1 due 1/11@6pm |
| 03 (1/12) | Style guidelines, functions | Ch 2.3, 2.4.2, 2.5.1 | | n/a | |  |
| 04 (1/14) | Variable scope |  | |  | | Project 2 due 1/18@6pm |
| 05 (1/19) | Loops, namespaces, booleans | Ch 2.6-2.7  Ch. 7.4.4 (indef iteration, while loop)  Ch. 3.2-3.2.6 (2d ed. 3.2-3.2.5), 3.4.3 | | n/a | |  |
| 06 (1/21) | Conditionals, monte carlo methods, strings |  | |  | | Project 3 due 1/25@6pm |
| 07 (1/26) | Testing and debugging, midterm review |  | |  | |  |
| 08 (1/28) | MIDTERM 1 | MIDTERM 1 | |  | | Project 4 due 2/1@6pm |
| 09 (2/2) | Testing and debugging (cont’d) |  | |  | |  |
| 10 (2/4) | Tracing, assignment (dynamic, strong typing) |  | |  | | Project 5 due 2/8@6pm |
| 11 (2/9) | Pointers, mutable data types | Ch. 4 | | n/a | |  |
| 12 (2/11) | Collections |  | |  | | Project 6 due 2/15@6pm |
| 13 (2/16) | Python file processing,  Data Mining | Ch. 5.1-5.2.5 | | n/a | |  |
| 14 (2/18) | MIDTERM 2 | MIDTERM 2 | |  | | Project 7 due 2/22@6pm |
| 15 (2/23) |  | Ch. 7 | | n/a | |  |
| 16 (2/25) | K-means cluster algorithm |  | |  | | Project 8 due 3/1@6pm |
| 17 (3/2) | Binary search, recursion |  | |  | |  |
| 18 (3/4) | Intro into user-defined classes |  | |  | | Project 9 due 3/8@6pm |
| 19 (3/9) | User-defined classes (cont’d), programming environments | Ch 13 | |  | |  |
| 20 (3/11) | Summing up, final review |  | |  | |  |
| 3/16 0800-1000 | FINAL | FINAL | |  | | FINAL |

# Course Policies

# Attendance:

# CIS 210 is a UO "mandatory attendance" class. If you do not attend the first class, you will be dropped from the class. At the start of each lecture please have your video on. You may turn it off as class resumes, however I encourage (and prefer) you leave it on.

* Exercises, individual and small group problem solving, and other practice material (posted on Canvas) will be given in lab each week. You must attend the lab to which you are assigned. To receive credit *you must be present AND participate in lab*. If you cannot attend lab during a particular week, a subsequent video of it, given by a GE, will be posted to Canvas. Three lab grades will be automatically dropped (not included in the final grade calculation) at the end of the term for all students.
* The midterms and final exam will be given during a regular class period:
  + Midterm 1: Thursday, January 28th
  + Midterm 2: Thursday, February 18th
  + Final Exam: Tuesday, March 16th: 8:00am-10:00am

By registering for CIS 210, you are agreeing that you can take the midterms and final exams at their scheduled times. If you cannot make the scheduled exam times, you should plan to take CIS 210 another term.

# Programming Projects and Code Demos

* We will be using Python version 3.x in CIS 210. (Any version of Python 3 is ok; 3.6 or higher is recommended.)  IDLE is an integrated development environment for Python. When you download Python, you will also be downloading IDLE. IDLE is the official development environment for CIS 210. You are welcome to use other environments for writing and running Python code, but everything you write for class feedback (e.g., lab help hours, code reviews) should run using IDLE. Go to the [official Python website](https://www.python.org/downloads/) to download Python/IDLE.
* Programming projects will be assigned each week, beginning in week one. Each week, several projects will be posted on Canvas (by Tuesday) and will be due the following Monday at 6pm. You are responsible for checking Canvas for all of the projects due that week. The final version of your project solutions should be submitted on Canvas. Submit all projects by the date and time they are due, i.e., when the Canvas submission option is available. Each project that reflects a reasonable, good faith effort will receive a project point. Four projects will be automatically dropped (not included in the final grade calculation) at the end of the term for all students.
* Code demonstrations are individual demonstrations of your computational problem solving skills, for individual feedback. Each student must complete one code demonstration with one of the 210 GEs during weeks 5 through 10 of the term. Code demonstrations should be scheduled using the CIS 210 Calendar on Canvas, and will take place on Zoom.
  + During your code demo, you will be asked to write and discuss a solution to a coding problem chosen from CIS 210 projects, labs, and class, using the CIS 210 structured approach to computational problem solving. The code demonstration will be graded using the rubric that is included with each weekly problem set.

# Consulting, Collaborating and Individual Work:

Consulting and collaborating on weekly projects is encouraged - share ideas, not code. Students are free to approach working on weekly projects as they see fit. If you learn best engaging with a problem on your own, that is fine. If you prefer to work with one partner, or in a group, or some combination of these, that is also fine. What you turn in must reflect your own understanding.

We strongly encourage you to discuss problems and problem-solving approaches with CIS 210 instructional staff; indeed, this is the main purpose of CIS 210 lab help hours!

The problems in CIS 210 are designed to give you an opportunity to practice the topics, concepts, and skills covered in CIS 210. *Solutions to problems will always rely ONLY on Python tools and skills that have been covered in class, lab, or the text up to the time the problem is assigned. Using sources from outside CIS 210 may result in solutions that are off-topic, or don't meet the problem specification, or are overly complex, or do not align with CIS 210 style guidelines, for example.*

# Late Work

Late project submissions will ***not*** be accepted unless prior arrangements have been made with the instructor at least 24 hours before the due date and time and must be turned in within 24 hours of the original due date.

## Extra Credit

This course does not offer extra credit.

## Barriers and Accommodations

Our goal is a fully inclusive class, accessible to everyone. If you encounter or anticipate barriers to full participation and fair evaluation for any reason, please communicate your needs to the instructor so that we can find a suitable accommodation. If you encounter or observe other impediments to full participation, for yourself or others, please share your concerns with the instructor. You are also encouraged to contact the Accessible Education Center in 360 Oregon Hall at 541-346-1155 or [uoaec@uoregon.edu](mailto:uoaec@uoregon.edu).

It is particularly important that you inform the instructor in the first week of the quarter if you require accommodation. Delayed notification of such a requirement may make it impossible to provide the accommodation.

## Academic Honesty

The University Student Conduct Code (available at [conduct.uoregon.edu](file:///Users/brittany_other/Desktop/Courses/Syllabi/conduct.uoregon.edu)) defines academic misconduct. Students are prohibited from committing or attempting to commit any act that constitutes academic misconduct. Students should properly acknowledge and document all sources of information (e.g., quotations, paraphrases, ideas) and use only the sources and resources authorized by the instructor. If there is any question about whether an act constitutes academic misconduct, it is the student’s obligation to clarify the question with the instructor before committing to attempting to commit the act.

All projects turned in for the course must be your own work. Copying from other class members or other sources is not acceptable. If you collaborate with someone else on an assignment, you must indicate such on the work you turn in. Collaboration that is not explicitly credited is plagiarism.

Academic honesty is expected and cases of suspected dishonesty will be handled according to university policy. In particular, copying someone else's work (including material found on the web) will not be tolerated. If solutions to assignments are obtained from outside sources, the source must be cited.

You are also responsible for protecting your work. That is, you must take reasonable precautions to prevent your work from being copied. This means that if you store your assignment solutions on a shared server, the file permissions must be set to keep others from accessing your files. If you are working on assignments in the lab, you must remove any of your files from the lab machine before you leave.

Turning in someone else's code is collusion, and is a particularly heinous form of plagiarism; if collusion is detected, all individuals involved (i.e., the copyee and all copyers) will be given a grade of F in the course.

## Academic Disruption due to Campus Emergency

In the event of a campus emergency that disrupts academic activities, course requirements, deadlines, and grading percentages are subject to change. Information about changes in this course will be communicated as soon as possible by email and on Canvas. If we are not able to meet face-to-face, students should immediately log onto Canvas and read any announcements and./or access alternative assignments. Students are also encouraged to continue the readings and other assignments as outlined in this syllabus or subsequent versions of the syllabus.

## Prohibited Discrimination and Harassment Reporting

I am a *designated reporter*. For information about my reporting obligations as an employee, please see [Employee Reporting Obligations](https://investigations.uoregon.edu/employee-responsibilities#employee-obligations) on the Office of Investigations and Civil Rights Compliance (OICRC) website*.*Students experiencing any form of prohibited discrimination or harassment, including sex or gender-based violence, may seek information and resources at [safe.uoregon.edu](http://safe.uoregon.edu/), [respect.uoregon.edu](https://respect.uoregon.edu/), or [investigations.uoregon.edu](https://investigations.uoregon.edu/) or contact the non-confidential Title IX office/Office of Civil Rights Compliance (541-346-3123), or Dean of Students offices (541-346-3216), or call the 24-7 hotline 541-346-SAFE for help. I am also a mandatory reporter of child abuse. Please find more information at [Mandatory Reporting of Child Abuse and Neglect](https://hr.uoregon.edu/policies-leaves/general-information/mandatory-reporting-child-abuse-and-neglect).