CSE20 Beginning Programming in Python

Contents

1 General Information ............................................. 3
  1.1 Class meeting times ........................................ 3
  1.2 Prerequisites ............................................... 3
    1.2.1 Course website .................................... 3
  1.3 Instructor information ...................................... 4
    1.3.1 Teaching Assistant ................................ 4

2 Course Description and Materials ............................... 4
  2.1 Learning Outcomes ......................................... 5
  2.2 Reading Materials ......................................... 5
  2.3 Course assignments and grading ............................. 6
    2.3.1 Programming assignments ............................ 6
    2.3.2 Feedback assignment ................................ 7
    2.3.3 Module review assignment ............................ 8

3 Class policies .................................................. 8
  3.1 Attendance ................................................ 8
  3.2 Assignment deadlines ...................................... 8
  3.3 Accessibility ............................................. 8
  3.4 Academic integrity ....................................... 9
    3.4.1 On academic integrity and coding ................... 10

4 University Resources ........................................... 10
  4.1 Learning Support Services (LSS) .......................... 10
  4.2 Basic Needs ............................................... 10
  4.3 Student Success Centers ................................. 10
  4.4 Undocumented Student Services (USS) ..................... 11
  4.5 CARE (Campus Advocacy Resources and Education) .... 11
  4.6 CAPS (Counseling and Psychological Services) .......... 11
  4.7 Student Health Center .................................... 13
5 Title IX & CARE Syllabus

6 Planned Schedule
1 General Information

This class will be an introduction to programming using the Python language, for people who have never previously had programming experience.

1.1 Class meeting times
Mondays, Wednesdays and Friday, 10:00 to 11:35 AM

1.2 Prerequisites
No programming knowledge is required, but the class will require access to a computer in which you can install programs and basic computer skills:

- Logging into your computer account;
- Connecting to the internet;
- Using a browser to open web pages;
- Downloading and installing software;
- Browsing files in your computer;
- Creating and deleting files in your computer;
- Creating text documents in Word or Docs or similar programs.

1.2.1 Course website
Canvas: https://canvas.ucsc.edu/courses/44507
1.3 Instructor information

- Name: Daniel S. F. Alves, PhD Candidate at UCSC
- E-mail: dalves “at” ucsc.edu
  Due to high volume of emails that I receive, please allow at least one work day for a reply.
- Office hours: Tuesdays and Thursdays, 10:00 to 11:59 AM
  This is a moment to come and ask any questions that may arise. It will be hosted on Zoom (link on Canvas). There will be a waiting room to allow one-on-one questions.

1.3.1 Teaching Assistant

- Name: Harikrishna Kuttivelil
- E-mail: hkuttive “at” ucsc.edu
- Discussion sections: TBD
- Office hours: TBD

2 Course Description and Materials

From [https://courses.soe.ucsc.edu/courses/cse20](https://courses.soe.ucsc.edu/courses/cse20):

Provides students with Python programming skills and the ability to design programs and read Python code. Topics include data types, control flow, methods and advanced functions, built-in data structures, and introduction to OOP. No prior programming experience is required. Students may not receive credit for CSE 20 after receiving credit for CSE 30. Students with prior programming experience (especially in Python) are encouraged to take CSE Testout Exam to be evaluated for their readiness to take CSE 30 directly: [https://undergrad.soe.ucsc.edu/cse-20-testout-exam](https://undergrad.soe.ucsc.edu/cse-20-testout-exam) (Formerly CMPS 5P, Introduction to Programming in Python.)

This class will be an introduction to programming through the learning of Python. The objective is to help students that have no programming experience to become familiar with programming concepts, while at the same time
learning a general-purpose programming language that can be applied outside of class. Classes will consist of lectures to present programming concepts grounded in the Python programming language, with time for questions, and assignments to exercise the ideas seen in lectures.

2.1 Learning Outcomes

By the end of this course the students will:

- have read many samples of Python code to learn its features. This includes recognizing data types, flow commands, functions, basic data structures and the basics of Object-Oriented-Programming (OOP), and how these concepts connect to each other;

- understand how to execute existing Python code, either through the interpreter, through an IDE or online platforms such as Colab;

- have applied Python to solve small problems, using techniques with general application. This will involve writing new code and using existing libraries. An example of an application would be reading some files and doing calculations according to the content read;

- have expressed what their code does by writing small documentation samples.

2.2 Reading Materials

- **Textbook**: Introduction to Python Programming
  - Authors: Gowrishankar S. and Veena A.
  - Available online at the UCSC library: [https://library.ucsc.edu/](https://library.ucsc.edu/)
  - It should also be available at the Bookstore if you want a physical copy

- **Optional** textbooks:
  - zyBooks: CSE 20 Introduction to Python Programming
    * Paid interactive online book
    * Instructions in Canvas
    * Another free reference for Python Programming
2.3 Course assignments and grading

The class will have the following assignments:

2.3.1 Programming assignments

The class will have programming assignments every week, but most assignments will not be graded and will be provided with instructions on how to verify the correctness of the solution found. Assignments will also start with most of the code provided, gradually progressing with more parts missing so that by the end you will be writing the entirety of the requested code. That way you will be exposed to code to build familiarity, as you learn to write your own code, and you will also learn how to test your code to validate it. Graded programming assignments will follow the same model, but will not be provided with the validation method. Self-validation of the code will be part of the assignment and you can indicate what testing you have done yourself for up to 5% extra credit in the assignment (keeping the limit of 100%).

Every week will have non-graded and graded programming assignments, with some weeks having lighter graded assignments. These assignments are planned to take at most a few hours of your time, although later assignments might require more. The assignments will be published on Monday and are due on Friday at the end of the day, but will be graded in the other they are submitted, so you can receive feedback earlier if you submit earlier. This frequency is intended to help you verify that you are understanding the concepts well and enable you to ask for assistance early if necessary. Programming assignments will cover 80% of the total grade.

1. Code validation Code validation will happen through more code, which will execute the written code with some input and compare observe output with expected. The validation will involve a series of input/output pairs that can be classified in four categories:

   obvious examples provided with the instructions;
   basic input that is simple to guess the output without executing the program;
   advanced input that is harder to guess without some calculations;
   “tricky” input that might require special consideration to work properly.
For examples of what the latter three categories look like, consider a piece of code that was written to calculate the value of $x$ to the power of $y$. In that case, $2^2 = 4$ would be a basic case $24^4 = 331776$ would be an advanced case and $(-1)^0 = 1$ would be tricky because of both the exponent zero and negative base.

2. Grading

Grading will be proportional to how the code performs with the given tests, following roughly this scale:

- **90-100%** the code pass almost or all tests;
- **80-90%** the code passes all advanced tests, and some tricky tests;
- **60-80%** the code passes most advanced tests, and maybe some tricky tests;
- **40-60%** the code passes only basic or obvious tests, and maybe some advanced or tricky tests;
- **10-40%** the code passes only obviously tests, and maybe some basic tests;
- **0-10%** the code does not run, or is not syntactically correct.

Some of your graded assignments may also be examined for written feedback on your coding style. Later assignments will also be required to have documentation in the form of comments (to be explained in class) and that will also be examined during written feedback. This feedback will not influence your grade, the purpose is to help you think about your coding style and consider how you can write readable code. After receiving feedback, you will be allowed to resubmit your code with modifications based on the feedback to recover up to 10% of the grade.

### 2.3.2 Feedback assignment

After receiving feedback on your code, you will be provided with guidelines and given access to a submission from another student. You will then have to write some feedback, which will be reviewed. This will be graded by participation and will correspond to 10% of the final grade. The main purpose of this assignment is to provide you with more feedback and to help you get used to giving feedback to the code of others.
2.3.3 Module review assignment

Near the end of the quarter you will be asked to review an external Python module from a provided list and describe some of the functionality it provides. More details will be provided once you are more familiar with programming, but this should be a two-three pages assignment at most. The purpose of this assignment is to encourage you to explore existing Python modules that might be useful to you in the future. This assignment will be made available on a Monday and will be due Friday of the next week. This will be graded by participation (provided that the content is valid) and will correspond to 10% of the final grade.

3 Class policies

3.1 Attendance

Class attendance is not mandatory, however it is encouraged as it enables you to ask questions that may arise during presentation of new material. Some classes will have conceptual quizzes so that we can track the learning of the class, and it will also help you check your understanding of the concepts.

3.2 Assignment deadlines

Assignments are due at 11:59PM PST on the day specified. The main purpose of the deadlines is to ensure timely feedback, but we understand that complications can arise, so we will also use a timebank of five days. That means that you have five days that you can distribute to extend deadlines (with a limit of two days per deadline) from assignments across the quarter, no questions asked. If you require further accommodations to help with deadlines, please feel welcome to contact us so we can arrange a solution as soon as possible.

3.3 Accessibility

UC Santa Cruz is committed to creating an academic environment that supports its diverse student body. If you are a student with a disability who requires accommodations to achieve equal access in this course, please submit your Accommodation Authorization Letter from the Disability Resource Center (DRC) to me privately during my office hours or by appointment, preferably within the first two weeks of the quarter. At this time, I would also like us to discuss ways we can ensure your full participation in the course.
I encourage all students who may benefit from learning more about DRC services to contact DRC by phone at 831-459-2089 or by email at drc@ucsc.edu.

3.4 Academic integrity

You are a member of an academic community at one of the world’s leading research universities. Universities like ours create knowledge that has a lasting impact in the world of ideas and on the lives of others; such knowledge can come from an undergraduate paper as well as the research of an internationally known professor. One of the most important values of an academic community is the balance between the free flow of ideas and respect for the intellectual property of others. Researchers don’t use one another’s research without attribution because we want to recognize the foundational work we have drawn on in order to create new knowledge; scholars and students always use proper citations in papers; professors may not circulate or publish student papers without the writer’s permission; and students may not circulate or post materials (handouts, exams, syllabi—any class materials) from their classes without the written permission of the instructor.

Any test, paper, or report submitted by you and that bears your name is presumed to be your own original work that has not previously been submitted for credit in another course unless you obtain prior written approval to do so from your instructor. In all of your assignments, including your homework or drafts of papers, you may use words or ideas written by other individuals in publications, web sites, or other sources, but only with proper citation (generally, quotation marks or an indented quotation) and correct attribution.

If you are not clear about the expectations for citing sources, completing an assignment, or taking a test or examination, be sure to consult the library website on citing sources (https://guides.library.ucsc.edu/citesources), and if you are still confused, seek clarification from your instructor or TA before turning in your assignment.

Finally, you should keep in mind that as a member of the campus community, you are expected to demonstrate integrity in all of your academic endeavors and will be evaluated on your own merits. The consequences of cheating and academic dishonesty—including a formal discipline file, possible loss of future internship, scholarship, or employment opportunities, and denial of admission to graduate school—are simply not worth it. Consequences and process for academic misconduct appear here: https://ue.ucsc.edu/academic-misconduct.html
3.4.1 On academic integrity and coding

When coding it is very common to look for online references on how to do certain tasks. Entire websites have been built around this concept, for example StackOverflow. While that is normal, and learning to search for the right information can be a valuable skill, finding the complete solution to a problem and submitting that, even with small modifications, presents an issue of academic integrity. There are automated tools that can go through large amounts of code to look for similarities that indicate plagiarism, and we have to apply them to submitted code.

But more than that, there’s the issue of learning: even if you manage to find the specific solution to a problem you are trying to solve as part of an assignment, try to solve it on your own to exercise your programming skills. If you are having difficulties to find the solution on your own, please feel welcome to ask me or the TAs for support, we are here to ensure your success.

4 University Resources

Here is a list of resources that UCSC makes available to you to help you with the class material and other support you might need.

4.1 Learning Support Services (LSS)

https://lss.ucsc.edu/about/index.html

LSS provide students with learning opportunities facilitated by peers to help support your studies. This FAQ provides more information on the services provided and how to sign up: https://lss.ucsc.edu/faq/programs-faq.html

4.2 Basic Needs

https://basicneeds.ucsc.edu

If you are experiencing challenges related to basic needs, such as food, housing, health & wellness, or financial security, visit the Basic Needs hub for information about food pantries, accessible housing, mental health support, and financial aid options.

4.3 Student Success Centers

https://studentsuccess.ucsc.edu/resource-centers/index.html
UC Santa Cruz has a variety of resources to support your overall success at UC Santa Cruz, ensure accessible living and learning environments, help you when you’re experiencing personal or academic challenges, and support you in building community. If you do not find what you’re looking for on the list at the link above, you are encouraged to contact a college advisor, or to go directly to Slug Support.

4.4 Undocumented Student Services (USS)
https://eop.ucsc.edu/undocumented_student_services/index.html

Under the umbrella of the Educational Opportunities Programs, Undocumented Student Services (USS) provides personal, academic, financial, and legal support to all UCSC undocumented students, including scholarship and fellowship information, community-building meetings, legal assistance and workshops, and access to fresh produce.

4.5 CARE (Campus Advocacy Resources and Education)
https://care.ucsc.edu/

CARE is a confidential space to discuss issues of dating violence, sexual assault, and stalking. CARE advocates provide support in a variety of ways depending on your needs, such as by supporting you in your decision-making; understanding the complexities that can arise from these issues; providing emotional support and free services; and providing resources and referrals.

Make an appointment:

1. https://care.ucsc.edu/services/advocacy-appointment-request1.html

Give CARE a call: (831) 502-2273
Visit CARE: Oakes Administration Building, Room 221 (as of Fall 2019) [If using Google navigation, search 223 Oakes Road]

4.6 CAPS (Counseling and Psychological Services)
https://caps.ucsc.edu/

If you are in distress, managing heightened stress and anxiety, or want to get more support and a counselor’s perspective on something you’re going through, CAPS provides a variety of services for your needs—including immediate crisis support, scheduled individual appointments, group counseling, and workshops led by peer advisors.

Crisis Assistance:
• [https://caps.ucsc.edu/counseling/crisis-assistance.html](https://caps.ucsc.edu/counseling/crisis-assistance.html)

• Walk in: Location at the Student Health Center in the East Wing on the 2nd floor

• Call during operating hours: M-F 8am-5pm: (831) 459-2628

• Call after operating hours: (831) 459-2628, then select option “3” (talk to a counselor) to get connected to an after-hours counselor

Drop-in Services:

• [https://caps.ucsc.edu/drop-in-services-no-left-menu.html](https://caps.ucsc.edu/drop-in-services-no-left-menu.html)

• Drop-in services include “Let’s Talk,” a service offered at a variety of campus locations where you can stop by for a one-time talk with a professional counselor to get information or support, and “Relaxation Station,” a designated area in the CAPS office designed to help you de-stress.

LGBTQ+ Services:

• [https://caps.ucsc.edu/counseling/lgbtq-counseling-services.html](https://caps.ucsc.edu/counseling/lgbtq-counseling-services.html)

• Resources and counselors with specific awareness of intersecting and non-binary identities.

Scheduled Services:

• [https://caps.ucsc.edu/scheduled-services-no-left-menu.html](https://caps.ucsc.edu/scheduled-services-no-left-menu.html)

• Includes individual and group counseling.

Self-Support Library:

• [https://caps.ucsc.edu/resources/self-help.html](https://caps.ucsc.edu/resources/self-help.html)

• Online resources on time management, managing stress and anxiety, and learning more about mental health.
4.7 Student Health Center

https://healthcenter.ucsc.edu/services/index.html

In addition to providing medical and health care, the Student Health Center provides resources and information for overall wellness and sexual health at Student Health Outreach and Promotion (SHOP). SHOP also provides a safe, confidential, and nonjudgmental space where you can talk about and get information about alcohol and other drugs. The COVE offers a space for sober students or students questioning their relationship to alcohol and other drugs. If you’re seeking more information, strategies, and opportunities to engage in honest dialogue about safer social and party experiences, visit Party Like a Slug.

5 Title IX & CARE Syllabus

UC Santa Cruz is committed to providing a safe learning environment that is free of all forms of gender discrimination and sexual harassment, which are explicitly prohibited under Title IX. If you have experienced any form of sexual harassment, sexual assault, domestic violence, dating violence, or stalking, know that you are not alone. The Title IX Office, the Campus Advocacy, Resources & Education (CARE) office, and Counseling & Psychological Services (CAPS) are all resources that you can rely on for support.

Please be aware that if you tell me about a situation involving Title IX misconduct, I am required to share this information with the Title IX Coordinator. This reporting responsibility also applies to course TAs and tutors (as well to all UCSC employees who are not designated as “confidential” employees, which is a special designation granted to counselors and CARE advocates). Although I have to make that notification, you will control how your case will be handled, including whether or not you wish to pursue a formal complaint. The goal is to make sure that you are aware of the range of options available to you and that you have access to the resources you need.

Confidential resources are available through CARE. Confidentiality means CARE advocates will not share any information with Title IX, the police, parents, or anyone else without explicit permission. CARE advocates are trained to support you in understanding your rights and options, accessing health and counseling services, providing academic and housing accommodations, helping with legal protective orders, and more. You can contact CARE at (831) 502-2273 or care@ucsc.edu. In addition to CARE, these resources are available to you:
• If you need help figuring out what resources you or someone else might need, visit the Sexual Violence Prevention & Response (SAFE) website, which provides information and resources for different situations.

• Counseling & Psychological Services (CAPS) can provide confidential counseling support. Call them at (831) 459-2628.

• You can also report gender discrimination and sexual harassment and violence directly to the University’s Title IX Office, by calling (831) 459-2462 or by using their online reporting tool.

• Reports to law enforcement can be made to the UCPolice Department, (831) 459-2231 ext. 1.

• For emergencies, call 911.

6 Planned Schedule

This is a tentative schedule with the goal of covering all topics in at a leisurely pace. It might be modified according to unplanned events.
<table>
<thead>
<tr>
<th>Weekday</th>
<th>Date</th>
<th>Lecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>June 21</td>
<td>Introduction, syllabus, requirements</td>
</tr>
<tr>
<td>Wednesday</td>
<td>June 23</td>
<td>Installing Python, using the UCSC Unix timeshare</td>
</tr>
<tr>
<td>Friday</td>
<td>June 25</td>
<td>Introduction to Python</td>
</tr>
<tr>
<td>Monday</td>
<td>June 28</td>
<td>Identifiers, variables, operators</td>
</tr>
<tr>
<td>Wednesday</td>
<td>June 30</td>
<td>Data types, casting</td>
</tr>
<tr>
<td>Friday</td>
<td>July 2</td>
<td>Control-flow: if, else, elif</td>
</tr>
<tr>
<td>Monday</td>
<td>July 5</td>
<td>Holiday</td>
</tr>
<tr>
<td>Wednesday</td>
<td>July 7</td>
<td>Control-flow: for and while loops</td>
</tr>
<tr>
<td>Friday</td>
<td>July 9</td>
<td>Review and consolidation</td>
</tr>
<tr>
<td>Monday</td>
<td>July 12</td>
<td>Functions</td>
</tr>
<tr>
<td>Wednesday</td>
<td>July 14</td>
<td>String manipulation</td>
</tr>
<tr>
<td>Friday</td>
<td>July 16</td>
<td>Review</td>
</tr>
<tr>
<td>Monday</td>
<td>July 19</td>
<td>Data Collections</td>
</tr>
<tr>
<td>Wednesday</td>
<td>July 21</td>
<td>Array and list</td>
</tr>
<tr>
<td>Friday</td>
<td>July 23</td>
<td>Dictionary</td>
</tr>
<tr>
<td>Monday</td>
<td>July 26</td>
<td>Tuple, set and frozenset</td>
</tr>
<tr>
<td>Wednesday</td>
<td>July 28</td>
<td>Exceptions</td>
</tr>
<tr>
<td>Friday</td>
<td>July 30</td>
<td>Files</td>
</tr>
<tr>
<td>Monday</td>
<td>August 2</td>
<td>Review</td>
</tr>
<tr>
<td>Wednesday</td>
<td>August 4</td>
<td>OOP</td>
</tr>
<tr>
<td>Friday</td>
<td>August 6</td>
<td>Classes</td>
</tr>
<tr>
<td>Monday</td>
<td>August 9</td>
<td>Class methods</td>
</tr>
<tr>
<td>Wednesday</td>
<td>August 11</td>
<td>Inheritance</td>
</tr>
<tr>
<td>Friday</td>
<td>August 13</td>
<td>Polymorphism</td>
</tr>
<tr>
<td>Monday</td>
<td>August 16</td>
<td>Modules</td>
</tr>
<tr>
<td>Wednesday</td>
<td>August 18</td>
<td>Review</td>
</tr>
<tr>
<td>Friday</td>
<td>August 20</td>
<td><em>The rest of this schedule is left open to</em></td>
</tr>
<tr>
<td>Monday</td>
<td>August 23</td>
<td><em>for more advanced topics that we will decide on</em></td>
</tr>
<tr>
<td>Wednesday</td>
<td>August 25</td>
<td></td>
</tr>
<tr>
<td>Friday</td>
<td>August 27</td>
<td></td>
</tr>
</tbody>
</table>