

Syllabus

Instructor

That's me! I am Eric Reed, and you can email me at reederic@foothill.edu (<mailto:reederic@foothill.edu>). It's easiest for me if you ask questions through the private or public message center here in the course, and only use email if you have trouble logging in. I get a huge amount of email and there's a chance that a time critical question will get lost if you email me.

Why do I get so much email? I am also the [STEM Center](http://foothill.edu/stemcenter) (<http://foothill.edu/stemcenter>) director here at Foothill. Which, by the way, is a good place to know. More about that later.

I teach both math and computer science. I hold an M.S. in math from CSU East Bay and an M.S. in computer science from Georgia Tech. Feel free to ask me about either one!

What you really need to know is that I am glad you are here, and my goal is your success in this course.

Office Hours

You can find me online on Sundays from 10am-11am at foothill.edu/stemcenter/onlinecs

I am also available on campus Tuesdays from Noon - 1PM in room 4149 (Exception, we will hold office hours on Monday Feb 4th instead of Tuesday Feb 5th).

If neither time works, feel free to contact me for an appointment.

Oh, what about this course?

CS 3B is an intermediate computer programming course using the Python language. We will explore object oriented programming in depth, including multiple inheritance, abstract base classes, inner classes, operator overloading, and more. We will look at some data structures like queues, linked lists and binary trees. To succeed in this class, you need a solid understanding of Python structure and statements, and some experience with object oriented design. You will also need both a desire to learn and a positive attitude.

This 3B class is going to be challenging for most. In 3A you were guided through most of the planning. In 3B you are expected to implement code with minimal spec. You will need to dedicate at least 15 hours per week to coding, and it is critical that you plan your time well to avoid a last minute scramble to get stuff done.

Python is in high demand right now because of its use in data analysis - boy is that a hot topic. You can code up some models pretty easily, and then use some of the extensions to make your code run super fast.

And where's the book?

There is no required textbook for the course. The ["official" Python tutorial](https://docs.python.org/3/tutorial/) (<https://docs.python.org/3/tutorial/>) should get you through, and it's completely free. If you prefer a textbook, you could pick up "Starting Out with Python" by Gaddis, any edition. But - I will not be referring to any book or requiring work from any book.

So I need to buy some expensive software, right?

You will need a (free) software package called a IDE, or Integrated **D**evelopment **E**nvironment along with a Python Interpreter (which includes a bytecode Compiler). I am going to recommend that you use PyCharm. The free version will give you everything you need for this class. I will insist that every assignment meets PEP-8 formatting standards, and if you code in PyCharm it will tell you everything you need to get there.

This is an online CS class. That means I sit at home alone with a Mountain Dew and get everything done without talking to anyone, right?

Wrong. There are a few places where you can meet and chat with your fellow students. And I think you should!

Public Forums

Questions and comments should be posted to the Discussions Tool which you can reach by clicking on Discussions on the left menu. I will usually reply within a day. Unless a question is of a private nature (i.e. grades, registration issues), please use the public Discussions. Also, feel free to answer your fellow students' questions even if you only have a guess as to what the answer is. It's great to engage in conversation with each other in this manner.

Steps needed to post your public questions and comments for this course can be found on the [Canvas Discussion Instructions Page](https://www.fgamedia.org/faculty/loceff/cs_courses/common/syllabi/cs_all_disc_canvas.html) (https://www.fgamedia.org/faculty/loceff/cs_courses/common/syllabi/cs_all_disc_canvas.html).

You must post an introduction in the first week of class or you will be dropped as a "no show" according to the college requirements. Use the "First Week Introductions" discussion in this Syllabus Module. In following weeks you can file bug reports, share interesting sites or articles you found, or ask and answer questions about the labs.

One thing though - Do Not Post Homework Code

Whether you have a question or suggested answer, never post homework code to forums. Create a separate small program to display your issue or illustration.

Private Messages

Please use public discussions for any question or comment that involves understanding the modules, tests or assignments. If you have a confidential question (grades or registration) use the Message Tool by first clicking on Inbox at the far left, then selecting this course and your intended recipient (usually me).

Steps needed to post your private questions and comments for this course can be found on the [Canvas Inbox Instructions Page](#) (https://www.fgamedia.org/faculty/loceff/cs_courses/common/syllabi/cs_all_INBOX_canvas.html).

Posting Program Code

You can post code to the public discussions that is not directly from your assignment. If you have an assignment question, translate that into a piece of code that does not reveal your answer or submission, exactly.

When posting code fragments (i.e., portions of your program) into questions, make sure these code fragments are perfectly indented and that they are properly formatted. For details see the required resource module [Pasting Code into Questions](#) (https://www.fgamedia.org/faculty/loceff/cs_courses/common/compiler/cs_all_posting_code_CANVAS.html).

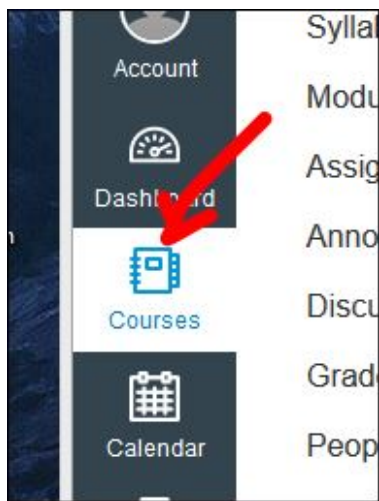
Do not post entire programs and ask "what's wrong?" or "is this good?" That's frivolous and indicates you have not tried to narrow down the problem. Find exactly what you want to know about and post only that part of the code.

STEM Success Center

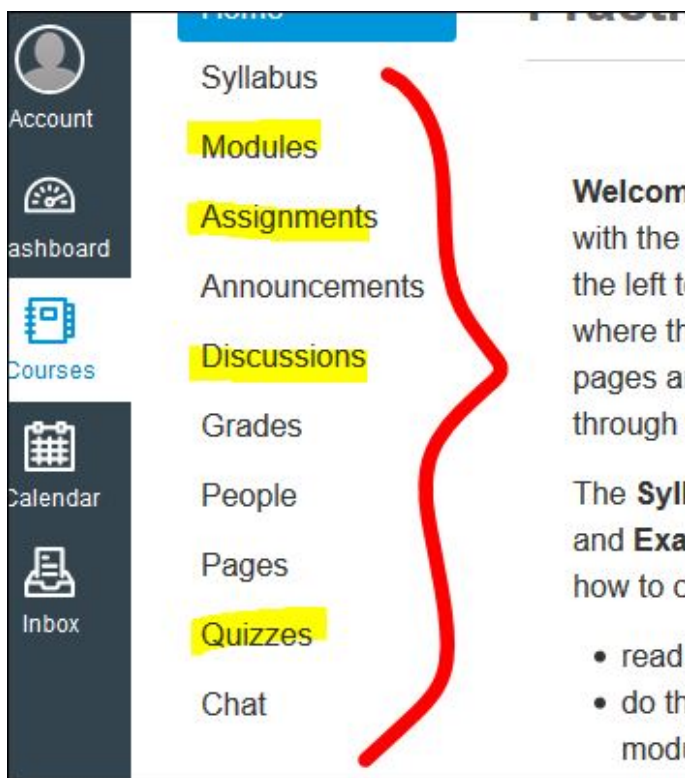
If the online forums here are not enough, please visit the [STEM Success Center page](#) (<http://foothill.edu/stemcenter>) and click **Schedule and Available Instructors**. They even have [online computer science tutoring](#) (<https://foothill.edu/stemcenter/tutoring-schedules/schedule-cs-online.html>)! These experts are qualified to help you with assignments or modules without giving you an answer that will short-circuit your discovery process. Let them know that you are not to receive actual assignment solution code or even fragments. They probably know this already, but it's your responsibility to avoid submitting something that was written by a tutor or another person.

All about Canvas

Access the various areas of your course by first selecting this course through the Canvas Courses choice in the far left ...



... then examining our course choices menu, also on the left side of the screen, but slightly to the right of Courses:



- Assignments: submitted through the Assignments Tool.
- Tests: taken through the Quizzes Tool.
- Questions or comments: posted using the **Discussions Tool**.
- Other areas: You'll find the names self-explanatory, and you can investigate them on your own.

Grades

Your grades are based on programming **lab assignments** (165 points), **exams** (15 + 15 = 30 points), and **participation** (20 points).

To earn full participation points, you must provide meaningful contribution to the discussions at least once every three calendar weeks (Sunday through Saturday, weeks 1-3, 4-6, 7-9 and 10-12).

This may be asking a "good" question, providing a helpful, non-duplicating answer to another student,

posting a bug report, or just bringing in some relevant information from outside the class. Realizing that this is somewhat subjective, I will post progress participation grades after week three so you will know how you are doing. For your first week it is sufficient to post a nice introduction (what's your name, why are you here, what coding experience do you have, etc).

Each lab assignment has its own rubric. Every assignment, though, has the following basic requirements:

- You must include an updated docstring that describes what's new in your program
- You must follow the PEP-8 style guidelines (PyCharm should not identify any PEP-8 violations in your code)
- Your code should be simple and elegant, without redundant code
- You must include a sample run that is exactly what is output by the program you submit

Following is a grading scale for your final grade:

Absolute Grading Scale	
% needed for	this grade
97	A+
91	A
88	A-
86	B+
80	B
78	B-
75	C+
67	C
60	D
< 60	F

No Ghosting!

For a complete reference of all withdrawal dates and deadlines refer to the [Foothill College registration page](https://foothill.edu/calendar/fall2018.html) [_\(https://foothill.edu/calendar/fall2018.html\)](https://foothill.edu/calendar/fall2018.html) at the college web site.

To stay enrolled in this class, you must participate regularly in your lab assignments and exams. This is part of the class participation that online classes must possess in order to maintain their transferability and accreditation.

You may be dropped for any of the following:

- Missing a scheduled test without prior notice may result in an automatic drop.
- If you do not login for **nine (9)** consecutive days I may drop you. (See exception below.)
- If you receive a zero on any two lab assignments, I may drop you. (See exception below.)
- If you do not post an introduction in the first week, you may be dropped for non-participation.

You will notice that I wrote "may" instead of "will." If you do not wish to continue with the course, it is your responsibility to drop yourself. Do not count on me dropping you, even if you are not participating.

Exception to Above Policies:

If the non-participation that has just been described occurs partially beyond the last date to drop, I may not be able to drop you, and you may receive whatever grade that your points dictate. Therefore don't assume that you can simply stop participating late in the quarter and you will be dropped. If you intend to drop please do so yourself, so you don't accidentally end up with an unintended "F."

Sharing is caring? Not always...

I think you will find me to be kind, supportive and patient. But, my patience is very short when it comes to cheating. **Your FIRST violation will result in a referral to the Dean of Academic Affairs.**

I strongly encourage you to help each other - on the discussion board, in the STEM Center, through Skype, wherever you happen to be. You can do flow charts together and strategize how you might logically complete the assignment. However - all the CODE you submit MUST be 100% your own. Therefore, never share any code from your assignments with anyone else. MANY MANY MANY times I have heard a student upset because someone else used their code in a computer science class and both students got a zero. Any variation of collaborating or copying programming lab assignments is prohibited. The assignment must be 100% your own work.

Please familiarize yourself with Foothill's [Academic Integrity Policy](#).

(<http://www.foothill.edu/services/documents/Z-Card.pdf>)

For those of you wishing to give help, please do not give away the answer. Either tell the person where they can look to find the solution, give them a general idea or ask them to ask me. Don't post actual assignment code.

The assignments on this site were created by me and I retain all rights. If you post the material online you are not only violating my copyright, you are also short-circuiting your own learning and the learning of other students. If you believe this is the only way to pass the class (despite the fact that I am available to you online, we have tutors in the STEM Center and online, and your fellow classmates are very generous with their help) then you should drop and try something else.

My strong advice to you is to completely avoid sites like Chegg or CourseHero. If you post assignments or use code there, you will receive a zero for the assignment, and a referral to the dean. Please read the terms and conditions for these sites, the sites will reveal the user information of a student (i.e. name on your credit card or bank account) if they are asked by an academic dean.

StackExchange is a legitimate place to ask small questions about how a particular function works (or why it's not working the way you think it should). Be very careful, though, not to post assignments or copy code even from StackExchange.

Think first, then ask constructive questions

It is easy to make sure your question is a good one: Make it specific. An example of a bad question is, "My program doesn't work. Here it is. Would you please see if you can tell me what I am doing wrong?" An example of a good question is, "My program doesn't work. Through trial and error I have determined that the problem lies in the following five lines, but I can't seem to narrow it down any further. Can you help?" This shows an attempt to organize and isolate the problem prior to asking for help.

Another example: BAD: "I don't understand the assignment. I'm lost. Please help." The reason this is a bad question is that there are a million things I might say to get the student on the right track, but I can't know which ones to focus on because I don't know where the misunderstanding lies. GOOD: "I understand the homework description up until you say 'XYZ'. But I'm not sure what you mean by 'XYZ'. In the lectures 'XYZ' seems to be...but here it seems to mean something different. From that point on, things get hazy because of this mismatch. Would you resolve this apparent difference for me?" Here, The student has told me exactly the first point of confusion so I know how to help.

I am not discouraging questions: I want you to ask. Through them, I get a chance to communicate with you. But narrow down the question. Show me you have tried to answer it and have made some progress. Show me exactly where you seem to be faltering so I can know how to help you. The same holds true if you are posing your question to a fellow student or to the whole class.

If you have (or think you may have) a learning challenge:

... please contact Disability Resource Center (**DRC**) at the start of the quarter. To contact **DRC**, you may:

- Visit **DRC** in Room 5400
- Email **DRC** at adaptivelearningdrc@foothill.edu
- Call **DRC** at 650-949-7017 to make an appointment

Student Learning Outcomes and Course Outline of Record

Our college uses Student Learning Outcomes to guide curriculum and assess the quality of instruction. Student Learning outcomes are broad, measurable, and student-centered. They do not represent everything you are expected to learn in the class.

The learning outcomes for Computer Science 3B are:

- **Python Inheritance:** A successful student will be able to write and debug Python programs which make use of inheritance, i.e. the "is a" relationship common to all OOP languages. Specifically, the student will define base and derived classes and use common techniques such as method chaining in his or her program.
- **Basic Python Abstract Data Types:** A successful student will be able to use the Python environment to define the basic abstract data types (stacks, queues, lists) and iterators of those types to effectively manipulate the data in his or her program.

You can access the official course outline of record for all **CS** courses here:

<http://www.foothill.edu/schedule/catalog.php> (<http://www.foothill.edu/schedule/catalog.php>)

From that page, select **Dept: Computer Science** → **Search**, and from there, select any CS course whose official outline you want to review.