C S 3A Course Syllabus

Introduction to Object Oriented Programming - Python

Information and Policies

S.1 Course Description and Prerequisite

CS 3A is an introduction to computer programming using the Python language. Absolute beginners or students already familiar with other programming languages will learn how to write Python programs that cover a wide range of applications. The prerequisites are:

- working knowledge of basic algebra
- good written English comprehension skills
- ability to work with computers
- access to the Internet

For success, you will also need a desire to learn, a substantial time commitment, and a positive attitude.

S.2 Instructor

My name is Mike Murphy I am the instructor for this class. Typically you will ask questions through the **Discussions** page, or through the **Inbox** (private message center) here in the course. Only use email if you have trouble logging in. My email address is murphymike@fhda.edu .

S.3 Text and References

All of the concepts will be covered in my modules. You do not need to buy a textbook. There are many free online references and tutorials. When you search for these, be sure to search for "Python 3", not the earlier version "Python 2" which has significant differences.

If you're one of the few people still buying (or reading) hardcopy books, these two Python books are OK. But you don't need them.

- **The ZyBooks ebook "Programming in Python 3", you can get free access to Chapter 1 of the ebook by signing up at https://learn.zybooks.com Links to an external site.. Enter the book code: FOOTHILLCS03AMurphyWinter2019, and click *Subscribe***

- **Python in a Nutshell, 3rd** edition by Alex Martelli et al - this is for people who might take Python further, or who want descriptions of everything in Python, including things not part of this course (this is the book I use). Costs $70 before discount!
- **Python for Everyone,** 2nd edition by Cay Horstmann et al - this is for people who like lots of pictures and icons. Costs $125 before discount!

You can use any of the several Python texts that you find, or I point out, online. You can order books through the Foothill Bookstore at http://books.foothill.edu/ (Links to an external site.)Links to an external site., phone: (650) 949-7305.

S.4 Python Interpreter/Compiler

You will need a (free) software package called an ***IDE***,
or **I**ntegrated **D**evelopment **E**nvironment. In this class we will be using a products combination that runs on Windows PCs, Macs and Linux called **Python/IDLE** (or just ***Python***).

An IDE integrates program creation and testing into a single environment. At a minimum an IDE includes a source code editor and a connection to the Python environment. Most IDEs include a variety of features to increase your productivity.

This Programming Language was named after the comedy group, Monty Python's Flying Circus, and IDLE was named after Eric Idle, one of the stars of the show.

I also will introduced another IDE, PyCharm a little later in the course.

If you are familiar with another ***IDE***, you are welcome to use that, instead. However, my assistance in the forums regarding compiler specifics will be limited to **Idle and PyCharm.**

You may stay with Python and or use PyCharm, it is your choice, IDLE is easier to learn but PyCharm had many additional features.

S.5 Communication

Public Forums

Questions and comments should be posted to **Discussions** which you can reach by clicking on Discussions on the left menu. I will usually reply within 24 hours.

Please use the public Discussions, Unless your question is of a private nature (i.e. grades, registration issues). 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 (Links to an external site.)Links to an external site..

First Week Required, Afterwards Recommended

There are ***no weekly Discussion requirements***, but it's good collegial form to participate, inquire and assist. Each student in the class can gain an extra 1 bonus point (total) by helping others in the Discussion forum. I'll email you when I award this to you. ***Also, you must post an introduction*** in the first week of class or you may be ***dropped as a "no show"*** according to the college requirements.

Do Not Post Homework Code

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

Private Messages

Please use ***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 Inbox by first clicking on Inbox at the far left, then selecting this course and your intended recipient (usually me.)

Steps needed to post your public questions and comments for this course can be found on the Canvas Inbox Instructions Page (Links to an external site.)Links to an external site..

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 (Links to an external site.)Links to an external site..

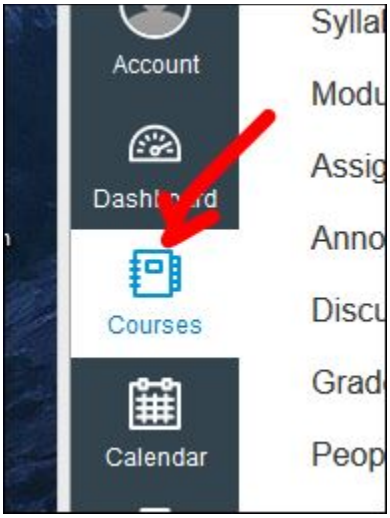
Do not post *entire programs* and ask "what's wrong?" or "is this good?" That's frivolous and indicates you have not tried to narrow down the problem. Find exactly what you want to know about and post only that part of the code.

STEM Success Center

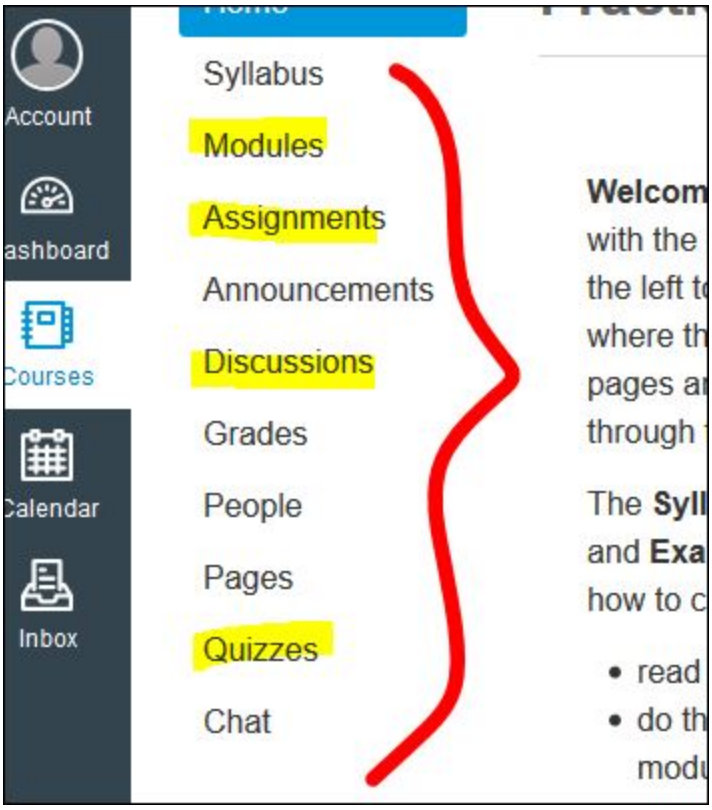
If the online forums here are not enough, please visit the STEM Success Center page (Links to an external site.)Links to an external site. and click **Schedule and Available Instructors**. These experts are qualified to help you with assignments or modules without giving you an answer that will short-circuit 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.

S.6 Where Everything Happens

Access the various areas of your course by first selecting this course through the ***Canvas*** Courses choice in the ***far left column.*** That's the icon that looks like a book...



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

S.7 Grades

Your grades are based on programming **lab assignments** (180 points = 75%) and **exams** (20 + 40 = 60 points = 25%).

If you fail to submit two or more assignments over the course, you may be dropped from the class.

If you fail to submit an assignment by the due date, you have 3 days more to submit it, and it will be graded at 50% of the grade you would have received for an on time submission. Assignments more than 3 days overdue are not accepted, and will be counted as a failure to submit the assignment.

Absolute Grading Scale

% needed for	**this grade**

97	A+
91	A
88	A-
86	B+
80	B
78	B-
75	C+
70	C
60	D
< 60	F

S.8 Drops and Withdrawal

For a complete reference of all withdrawal dates and deadlines refer to the Foothill College Calendar. (Links to an external site.)Links to an external 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 classes must possess in order to maintain their transferability and accreditation.

You may be dropped by me for any of the following:

- Missing a scheduled exam without prior notice will result in an automatic 0 grade for the exam.
- If you do not login for **five (5)** 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.

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."

If you decide to drop the class, please let me know. College policies do not allow anyone who has dropped a class to continue to have access to the material.

S.9 Collaboration

Working with any other person **on homework** results in a ZERO grade + a visit to the Dean of Student's Office.

Spouses, roommates, and friends taking the course together: **Don't discuss ungraded homework** with each other outside the public forums. Instead, direct all of your questions to the public forums where everyone can comment and I can moderate the discussion. Do not look for answers on cheater web sites or pay-for-help web sites.

Any variation of **collaborating on or copying programming lab assignments is prohibited**. The assignment must be 100% your own work. Changing a few variables around to make them look different won't fool me. And if it does fool me, you probably had to change so many things that you take longer than if you just do it yourself in the first place.

You can talk about the ***modules*** all day long off-line if you wish. The "no collaboration/discussion" rule only applies to lab assignments. There is a place to ask for help with homework: the Discussions labeled for that purpose, or the ***STEM Success Center***. I will spend hours helping you each week, both individually, and in groups.

You are encouraged to answer each other's questions in the Discussions, including questions about the homework. If I think you are giving too much information away, I'll edit your post. So there is no reason to ask people not on the course. Ask your course-mates in the Discussions forum.

If you accept help from someone who is not trained to teach without giving away the answer, it will damage your learning process -- you will actually become weaker in the subject. Now, you don't have to agree with me - but you do have to follow the rule. If you want to take a class where you get to solve problems in groups, there are other sections with instructors who have that option. But if you stay in *this* class, you are agreeing to do the lab assignments on your own or with help from us, here, in this course's public forum.

For those students ready to give help, please do not give away the answer. Either post 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.

S.10 How to Ask a Question

"There's no such thing as a bad question" is a myth. There are good questions and poor 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? Gretel"* Gretel is *lazy*. 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? Hansel."* Hansel made an attempt to organize and isolate the problem prior to asking for help. When he gets my answer, he is sure to remember it because he is prepared to hear exactly what he needs to know.

Another example: BAD: *"I don't understand the assignment. I'm lost. Please help. Jack."* The reason this is a bad question is that there are a million things that might get Jack on the right track, but I can't know which ones to focus on because I don't know where Jack's misunderstanding lies. Jack hasn't given me any help to help him. GOOD: I *understand the homework description up until you say 'recursive'. But I'm not sure what you mean by 'recursive'. In the lectures 'recursive' 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? Jill."* Here, Jill has told me exactly the first point at which she is confused so I know what to tell her to set her straight.

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.

S.11 To Obtain Disability-Related Accommodations ...

... 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

S.12 Expanded Content

Post your intro in Discussion forum

Week 1 modules - Compilers, IDLE, "Hello World", anatomy of a program, syntax ("compiler") errors vs. run-time errors, critical style rules and indention requirements.

Week 1R modules - download and install Python 3 onto your computer

Lab assignment 1

Week 2 modules - Numeric expressions, Python type "freedom," strings, ints, floats, lists, console output, type compatibility, playing computer. **Week 2R modules** - lab homework requirements **Lab assignment 2**
Week 3 modules - User input, selection (*if/else, elif*), logical expressions, deeper into str(ing) and list classes, relational expressions. **Week 3R modules** - lab homework requirements **Lab assignment 3**
Week 4 modules - Repetition (*for* and *while*), break, emulating do-while loops in Python, formatting numeric output. **Lab assignment 4**
Week 5 modules - Tuples, Dictionaries, file I/O, built-in Python algorithms and data structures, the interactive Python shell. **Lab assignment 5**
Week 6 modules - Program modularity, functions, methods, parameter passing. the functional return, local variable scope, default parameters, immutable and mutable types. **Lab assignment 6** **Mid-term exam**
Week 7 modules - OOP programming, instance data and methods, the "cls" parameter, *constructors, accessors* (getters), *mutators* (setters), protection of data, inner/nested classes, separation of I/O and computation. **Lab assignment 7**
Week 8 modules - Static data, static and class methods, the "cls" parameter, reference assignment, OOP program design strategies, object parameters. **Lab assignment 8**
Week 9 modules - list-simulated Arrays, user-defined array classes, using arrays with loops, simple sort algorithms, compound data types (arrays of objects and objects containing arrays), index bounds testing, introduction to exceptions. **Lab assignment 9**

Week 10 modules - Linear search algorithm, stack and queue data structures, recursion, binary search algorithm. **Lab assignment 10**
Week 11 modules - GIT repositories, review, gateway to advanced topics.
Week 12 Final Exam

- **Week 1** - Compilers, IDLE, "Hello World", anatomy of a program, syntax ("compiler") errors vs. run-time errors, critical style rules and indention requirements.
- **Week 2** - Numeric expressions, Python type "freedom," strings, ints, floats, lists, console output, type compatibility, playing computer.
- **Week 3** - User input, selection (*if/else, elif*), logical expressions, deeper into str(ing) and list classes, relational expressions.
- **Week 4** - Repetition (*for* and *while*), break, emulating do-while loops in Python, formatting numeric output.
- **Week 5** - Tuples, Dictionaries, file I/O, built-in Python algorithms and data structures, the interactive Python shell.
- **Week 6** - Program modularity, functions, methods, parameter passing. the functional return, local variable scope, default parameters, immutable and mutable types, optional examples in physics, medicine and climate modeling.
- **Week 7** - OOP programming, instance data and methods, the "cls" parameter, *constructors, accessors* (getters), *mutators* (setters), protection of data, inner/nested classes, separation of I/O and computation.
- **Week 8** - Static data, static and class methods, the "cls" parameter, reference assignment, OOP program design strategies, object parameters.
- **Week 9** - list-simulated Arrays, user-defined array classes, using arrays with loops, simple sort algorithms, compound data types (arrays of objects and objects containing arrays), index bounds testing, introduction to exceptions.
- **Week 10** - Linear search algorithm, stack and queue data structures, recursion, binary search algorithm.
- **Week 11** - GIT repositories, review, gateway to advanced topics.
- **Week 12** - Final exam.

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

http://www.foothill.edu/schedule/catalog.php (Links to an external site.)Links to an external site.

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

Student learning outcomes are:

- A successful student will be able to write and debug Python programs which make use of the fundamental control structures and function-building techniques common to all programming

languages. Specifically, the student will use data types, input, output, iterative, conditional, and functional components of the language in his or her programs.
- A successful student will be able to use object-oriented programming techniques to design and implement a clear, well-structured Python program. Specifically, the student will design and use classes and objects in his or her programs.

2 - Calendar and Due Dates

This is the second of your two-page syllabus. The first page has the general policies and rules of the course.

S.14 Weekly Activities

Every week you have

- two lessons, termed *Modules*, to read and study over Monday to Wednesday (recommended)
- one Lab Assignment, usually started on Wednesday, and due on Sunday night before 11:59pm. You will submit a file containing your code through the Canvas website.

There are exceptions, but this is the basic drill. This course is a lot of fun, and a lot of hard work. To pass the course you have to be disciplined and make time to do both of these activities.

Weekly Time Estimate

- **Allow about 10 hours each week for this course**. This includes reading the modules, pasting code into your compiler and trying the examples (essential to understanding). It also includes about 4 hours to complete lab assignments. Some of the assignments may take you longer than this. Software development aptitude, like musical or artistic talent, varies greatly in individuals.

Typical Week

Here is the day-by-day breakdown of a typical week. Some weeks differ, but this will help you understand approximately what you are facing on a weekly basis.

Typical Week	
by Monday evening	Read resource module R, read modules A and B, and copy/paste run all of the all examples in Python.
by Wednesday evening	Finish reading module B, and copy/paste and run all of the examples. Start work on this weeks assignment

Thursday through Sunday	Master all of the material in this weeks modules.
Sunday	Submit this weeks assignment before 11:59pm on Canvas Get ready to repeat the whole thing again, starting Monday.

S.15 Other Activities: *Discussions*, *Announcements*, *Tests*

Discussions

Please ask questions in the Discussions area of Canvas. I hope you will be active in this area. Anyone who knows the answer or has productive suggestions should respond. Read through the recent Discussions posts every time you log in, to make sure you gain the benefit of other students' questions and replies.

Weekly Posts Recommended (Not Required)

Other than the ***first week's introduction***, you are not required to post every week. However, if you are having difficulty, you should reach out and ask questions.

No Exact Homework Code Allowed

Please phrase questions in plain English or use non-homework code examples to demonstrate your question or suggested answer when posting.

Follow Module 3R When Posting

Code fragments must be formatted according Module 3R to receive an answer. Otherwise, we'll ask you to fix the formatting and we'll check back to answer the question once the formatting is achieved.

You must also ***post an introduction*** in the first week to avoid being dropped as a no-show. See my introduction over in Discussions > "Getting to Know You, Please Introduce Yourself to the Class."

Announcements

You will see an Announcement area in the ***Canvas*** course tools menu on the left. Check that area every time you login for late-breaking news.

Tests

The exams for this class will be online. There is a midterm exam on *Thursday* of the sixth week , and there is a Final Exam on *Wednesday* of the last week of the quarter. These exams will be available online for exactly 18 hours starting 6 AM on the due date and be due by midnight. You must take the tests somewhere in that 18 hour period. I will not accept late midterms or final exams. You must take the midterm in a single one-hour sitting and the final in a single two-hour sitting. Details about whether or not the test will automatically submit and lock-you-out an hour (or two) after you begin will be disclosed in the announcement area prior to the exam date.

S.16 Next Steps

Now that you have the idea, you can match that to the modules in the first week, and see you are supposed to:

- Monday
    - Read this syllabus, and resource R1. Post your introduction and photo in Discussion tool.
    - Read and work through modules A and B
- Wednesday - Finish reading and working through module B, start lab assignment due on Sunday night
- Sunday night, submit this week lab assignment before 11:59pm

Please get started.

S.17 Office Hours

There will be an Office Hours in-person meeting offered each week, on Monday from 5pm to 6:00pm, on the main campus in a classroom to be named in the first week of class. Please attend office hours if you would like some more guidance on a lab assignment, or have a topic to discuss.

Course Summary:

Date	Details	
Sun Jan 13, 2019	Getting to Know You, Please Introduce Yourself to the Class	due by 11:59pm
	LAB Assignment 1 - Hello World!	due by 11:59pm
Sun Jan 20, 2019	LAB Assignment 2 - Arithmetic due by 11:59pm	
Sun Jan 27, 2019	LAB Assignment 3 - Lists, Loops, and Lego due by 11:59pm	
Sun Feb 3, 2019	LAB Assignment 4 - Spongebob and the Table due by 11:59pm	
Sun Feb 10, 2019	LAB Assignment 5 - Functions (mm) due by 11:59pm	
Thu Feb 14, 2019	C S 3A Midterm Exam due by 11:59pm	
Sun Feb 24, 2019	LAB Assignment 7- A Triple String Class due by 11:59pm	
Sun Mar 3, 2019	LAB Assignment 8 - How to Cheat at Vegas due by 11:59pm	
Sun Mar 10, 2019	LAB Assignment 9 - How Fast benchmark due by 11:59pm	
Sun Mar 17, 2019	LAB Assignment 10 - A recursive program (pvdl) due by 11:59pm	
Sun Mar 24, 2019	LAB Assignment 11 - Changing Sort Keys due by 11:59pm	
Wed Mar 27, 2019	C S 3A Final Exam due by 11:59pm	