

Item Frequency Report [Filter Results](#)

Introduction to Python Programming - 2016-01-11 to 2016-01-13

1. Please check the box for each question that represents your opinion (1 to 5 where 5 is the highest rating):






The presentation was informative.

		Response Percent	Response Total
1		0%	0
2		0%	0
3		0%	0
4		13%	2
5		87%	13
Total Responses		15	

The instructor was knowledgeable of the materials.

1		0%	0
2		0%	0
3		0%	0
4		13%	2
5		87%	13
Total Responses		15	

The instructor was effective in delivering the materials.




1		0%	0
2		0%	0
3		0%	0
4		27%	4
5		73%	11
Total Responses		15	

I will be able to use the skills learned in my current job.

1		0%	0
2		0%	0
3		7%	1
4		20%	3
5		73%	11
Total Responses		15	

2. The length of the class was:

		Response Percent	Response Total
Too Short		0%	0
Just Right		100%	15

Too Long		0%	0
		Total Responses	15
4. Would you recommend this class to others?		Response Percent	Response Total
Yes		100%	15
No		0%	0
		Total Responses	15
3. How could the course be improved for future offerings?			
1.	Pair up weaker students with stronger students. We initially paired up with workstation neighbors, but that didn't work for me (a weaker one). I asked stronger ones in the class to pair up with me, that helped a lot.		
1.	Prior to class make clear what prerequisites are needed to be successful/get most out of class. (I'd say 30% of class needed better baseline in object oriented programming)		
2.	The Google Doc notes were helpful! Keep that up! 3. Get information email from prof out sooner to have all required installations ready to go before class started. 4. Teaming during exercises helped me to make the concepts concrete, forcing me to teach my partner my rationale for my algorithm implementations.		
3.	have a few printouts ready. Not all people will need one, but helps to annotate directly		
4.	Printouts of presentation and exercise materials available for note taking.		
5.	At least for me, it would have been nice to have a broader introduction to object oriented programming. Still, the class was excellent at not requiring much prior programming knowledge.		
6.	For those not familiar with python's structure or programming, it would be a good idea to include the hierarchy of classes, objects, functions, etc at the beginning of the course.		
7.	The monitors at the front of the room were outputting at really low resolution and the presenter's materials were harder than necessary to read.		
8.	Some sort of note set would be beneficial. We went through things orally and I took my own notes, but some set of high-level concepts that he really wants to make sure we understand would be a nice take-away.		
9.	Although I felt that this course was at a great pace for me (with background programming experience) I could tell that it was too fast for some who came in with no real OOP or programming concepts. I don't think it's appropriate to advertise that no previous programming experience is required, and I think there should be a prerequisite introduction to programming/OOP course to precede this one. Either that, or extend the length of this course to allow time to introduce the basic concepts of programming in general, before jumping in to how they are applied/handled in Python.		
10.	no comment		
11.	Prepare and provide installation steps for packages (like numpy or even python3) that are referenced during the course.		
12.	More real world engineering modeling and simulation problems/exercises to compliment the pure CS stuff		
13.	Could have of course probably spent a couple of months learning this stuff. But this method gave a broad enough overview for me to go back and re-visit ideas. It would have been good for the intro course to make sure everybody is 'set up' before beginning (i.e., make sure everyone is actually able to run python!). May have been my own mistake but I spent the first bit of the class just trying to figure out how to run it.		
14.	Maybe if we could get some of the basic concepts in advance of the class so that we could be more prepared in advance.		
15.	Overall, Glenn was an awesome instructor. He was enthusiastic, informative, and very engaging. The only comment I would like to leave for him to specifically name which documents he would open. Since the speed of the class is fairly fast (which is okay for what was listed as the recommended prereqs), it would have been nice if he named which scripts he would open before he would start going through them.		

5. General Comments:

1. 1. The style the course was taught was just right to offer chances to learn, ask questions and get hands on practice for the exercises. 2.Recommend to get bugs out of IDE's when running exercise code, to not confuse students
2. To get more out of the class, you should have some background in programming.
3. I appreciated your socratic teaching method. It kept me engaged even though some topics went over my head.
4. Great intro course, the instructor was very effective for me and I really enjoyed his interactive style.
5. Good explanations and exercises to teach concepts. I learned a lot!
6. Great and informative lectures, very energetic and engaging lecturer. I'm looking forward to taking the intermediate course, and I have already recommended this course to various colleagues.
7. Pairing students for class work did not benefit me.
8. Great class. Already recommending it.

Glenn was very engaging. I think the way the course is structured is smart, it requires constant attention by the student,
9. which enables quick progression. Course is definitely set up for program developers and less for scientific application. Will look for a python for science class in the future.
10. Glenn, for the speed of the class, and all the information that was delivered, I feel like the execution was amazing! I learned so much about python within the 5 days, it's incredible! Thank you so much!



Item Frequency Report [Filter Results](#)

Intermediate Python Programming - 2016-01-14 & 2016-01-15

1. Please check the box for each question that represents your opinion (1 to 5 where 5 is the highest rating):

The presentation was informative.

		Response Percent	Response Total
1		0%	0
2		0%	0
3		0%	0
4		15%	2
5		85%	11
Total Responses		13	

The instructor was knowledgeable of the materials.

1		0%	0
2		0%	0
3		0%	0
4		8%	1
5		92%	12
Total Responses		13	

The instructor was effective in delivering the materials.




1		0%	0
2		0%	0
3		0%	0
4		8%	1
5		92%	12
Total Responses		13	

I will be able to use the skills learned in my current job.

1		0%	0
2		0%	0
3		0%	0
4		15%	2
5		85%	11
Total Responses		13	

2. The length of the class was:

		Response Percent	Response Total
Too Short		8%	1
Just Right		92%	12

Too Long		0%	0
		Total Responses	13
4. Would you recommend this class to others?		Response Percent	Response Total
Yes		100%	13
No		0%	0
		Total Responses	13
3. How could the course be improved for future offerings?			
1. More examples of engineering problems and modeling/simulation in addition to the CS stuff			
2. Possibly send a crash course summary of the beginner course to the intermediate folks so they have a quick review to all more questions/materials to be covered.			
3. More examples to support the details of the exercises.			
4. some printouts			
5. N/A			
6. I would like to work more on numerical analysis, however it makes sense that the course was kept as broad as it was. I'm looking out for a python for science course.			
7. Cover fewer edge cases, more exercises.			
8. Maybe if we could get some of the basic concepts in advance of the class so that we could be more prepared in advance.			
9. It could be recorded on JPLtube for future review.			
The examples are great to walk through and very instructional. Sometimes, though, we discuss nuances verbally and unless you are able to write down notes, it's hard to retain that information. More detail in the comments may help with			
10. that (or you could release a version with more comments after we've discussed it in class). It also seemed like the skill level varied in the course (which is fine). Extra exercises or ways to make current exercises more challenging for advanced students would be good.			
11. Some additional challenging activities could have been available for those who finished early. With the large range of experience of the people taking the class an activity may take 5-45 minutes.			
Overall, Glenn was an awesome instructor. He was enthusiastic, informative, and very engaging. The only comment I would like to leave for him to specifically name which documents he would open. Since the speed of the class is fairly fast (which is okay for what was listed as the recommended prereqs), it would have been nice if he named which scripts he would open before he would start going through them.			
12.			
13. I think it would be nice to see additional practical examples. For example, parsing data (from a CSV or other source) and performing analytics on it. Doesn't necessarily need to be scientific analysis - could be searching for terms.			
5. General Comments:			
1. Great class. Already recommending it.			
2. The most effective way to learn in the intermediate course to be in the beginner course (as we just continued materials from beginner). Great course and style of teaching! I learned quite a bit and will sure to recommend!			
Glenn was a very engaging instructor. His methods force students to pay attention (which is great for people like me who			
3. have wandering minds) and understand the material. The way the exercises are set-up are very clever in that they require some higher order thinking rather than just plug and play			
4. Thanks Glenn! Great class			

5. Very informative and instructional course. Thanks for your time!
6. Great instructor and teaching method. Glen also provided tons of resources for further learning and remained open to email questions so we don't have to feel like we are on our own from here on out.
7. Glenn, for the speed of the class, and all the information that was delivered, I feel like the execution was amazing! I learned so much about python within the 5 days, it's incredible! Thank you so much!

8. The Socratic Method of teaching was sometimes confusing. Sometimes the examples were too specialized, and the message taken-away from that example was inadvertently too specific. For example, after introducing the way raising and catching exceptions works in Python, there was no clear explanation of all the different kinds of exceptions that can be caught or raised. It also wasn't clear how to find out when to expect what kind of exception outside the examples shown. More references to the Python documentation (when appropriate as comments in the examples) would have been immensely helpful to understand how the code used in the class can be applied generally.

