CS 1110-001 **Introduction to Programming** - Spring 2017

ENGR (17513)

INSTRUCTORS: **Tychonievich, Luther (lat7h)**

Respondents: 178 / Enrollment: 325

Summary: CS 1110-001 Introduction to Programming - Spring 2017 (17513)

Overall Course Rating

CS-1110-001 Mean 4.00
CS-1110-001 Std Dev 0.99
CS-1110-001 Response Count 878

SEAS, 1000-level courses Mean 3.86
SEAS, 1000-level courses Std Dev 1.02
SEAS, 1000-level courses Response Count 8112

Overall Instructor Rating

INSTRUCTOR: Tychonievich, Luther
Mean 4.43
Std Dev 0.75
Response Count 1223

SEAS, 1000-level courses Mean 4.02
SEAS, 1000-level courses Std Dev 0.94
SEAS, 1000-level courses Response Count 16340

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

1. My grade in this course accurately represents my mastery of the material within this course.

~

Question Type: Likert

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
178	3.72	0.97	39 (21.91%)	77 (43.26%)	37 (20.79%)	24 (13.48%)	1 (0.56%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
178	3.72	0.97	39 (21.91%)	77 (43.26%)	37 (20.79%)	24 (13.48%)	1 (0.56%)

2. How accurate is this statement for you: After taking this class, I personally have a better understanding of fundamental concepts in Computer Science.

~

Question Type: Likert

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
177	4.40	0.71	89 (50.28%)	74 (41.81%)	11 (6.21%)	2 (1.13%)	1 (0.56%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
177	4.40	0.71	89 (50.28%)	74 (41.81%)	11 (6.21%)	2 (1.13%)	1 (0.56%)

3. How accurate is this statement for you: After taking this class, I have a better appreciation for Computer Science.

~

Question Type: Likert

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
176	4.22	0.96	88 (50.00%)	53 (30.11%)	24 (13.64%)	8 (4.55%)	3 (1.70%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
176	4.22	0.96	88 (50.00%)	53 (30.11%)	24 (13.64%)	8 (4.55%)	3 (1.70%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

4. How accurate is this statement for you: After taking this class, I am more likely to major or minor in CS.

~

Question Type: Likert

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
176	3.29	1.42	46 (26.14%)	43 (24.43%)	32 (18.18%)	26 (14.77%)	29 (16.48%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
176	3.29	1.42	46 (26.14%)	43 (24.43%)	32 (18.18%)	26 (14.77%)	29 (16.48%)

5. Which topic/lecture in this course was your favorite and why?

~

Question Type: Short Answer

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-001, Tychonievich, Luther

Total	Individual Answers
146	*See below for Individual Results*

I really liked learning about gamebox because you could produce something really cool from it.

boolean decisions

The material in the beginning of the semester because the content was fundamental information that was key to my future in Computer Science.

My favorite topic was extracting information from a website or file and using that information in some way to return desired information. It was something that was very useful and was pretty cool to see the results.

I enjoyed making the game because it factored in a lot of creativity.

Reading and processing information from the web through urllib. I think this tool will be very useful for my personal projects in the future.

The gamebox stuff was the most fun in my opinion because it's satisfying to make a game

I like the when we learned about functions because they applied to all programs and seem to have the most practical application for the workforce.

Learning how to manipulate csv files --really applicable to real life. I, and most students I think, enjoy when CS is applicable.

Probably coding with pygame. It felt that we got to use the most discretion here which allowed us to be more creative.

Learning about gamebox/pygame

for loops because they were interesting

Gamebox and regex: applied the coding to real-world uses.

Gamebox because its cool

Gamebox/pygame because there is not one right answer. It allows you to test everything you have learned in a fluid and non-restrictive way.

I really enjoyed the discussion on list, dictionaries, and tuples. I feel there is a lot of power that can come from the use of these in your programs.

reading data was my favorite topic because it involved working with data outside of python as well as manipulating it inside python.

Understanding data, because I think it is a very useful topic.

pyGame, it was interesting and fun

N/A

for and while loops

Creating games. I was able to see what I did and play with it. I create a program for a reason.

Enjoyed the early topics, it was easier to follow along and i did not get as lost.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	Regular expressions because they're like a puzzle
	Despite the fact that it was very specific to the class and doesn't have direct application to coding in general, I really enjoyed the gamebox exercises. Building something that you can interact with and show to others was a very satisfying activity and I also found myself looking at simple games and think of how I would build them using that code.
	I liked learning about loops because they were so applicable in so many other coding topics.
	I enjoyed regular expressions because they were interesting and applicable
	My favorite activity of this course was the lab in which we used our current location to determine the nearest Wendy's. I was blown away at the fact that I actually wrote code that could find the nearest Wendy's. To me, it showed just how powerful CS can be.
	learning function use because I believe it is an important foundation of learning.
	Understanding Data. I personally think data analysis is a crucial part of computer science.
	Game creation because it allows students to be creative.
	I really enjoyed doing assignments that pull information from online websites because that is very applicable and makes me feel like I'm doing something meaningful. I also really enjoyed the game unit because who wouldn't love doing that.
	I enjoyed learning the basic computing strategies and learning how to apply them to a wide variety of programs.
	Pygame because I felt it was the most practical
	Regular expressions was my favorite because I found it the most interesting and the most useful.
	learning how to code simple functions and working on the game project
	First one, did not attend many others
	I enjoyed the gamebox topic because it was fun and allowed us to be creative.
	I really enjoyed regular expressions. I think that they were fun and I had to think in a different way then normal which is an enjoyable exercise.
	While loops. An idea that was easier to understand than lists.
	Probably learning how games work. Also, I enjoyed learning about obtaining data from urls and csv files.
	Learning about csv files due to its applicability outside of CS.
	the topics reviewed for the first exam because I actually understood what the code was doing.
	My favorite is the lecture on loops and applying them to lists as it taught me a useful method to use when I program.
	Making games. It's easier to understand
	I liked the gamebox unit because it related to things I want to do when I'm older.
	Either functions or loops because I found it so fun to make programs with which users could interact and programs that could do interesting things
	Regular expressions because of the way they are implemented
	turtle, because it was simple, fun, and felt like I could actually do something real
	Games-it was fun to make them.
	Writing the game was very fun.
	learning about functions and loops because I found that to be the most useful and relevant.
	Pygame and Gamebox because it utilized everything we had learned
	n/a
	Loved Reg Ex, seemed like building a puzzle.
	Writing files is pretty cool

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	I enjoyed the gamebox portion of the class the most. It provided programming in which an immediate result is seen and can be used practically. RegEx's because they were easy gamebox and pygame because it combines material learned throughout the semester and applies it to something appealing and fun to spend time mastering and perfecting Regular expressions - I enjoyed the assignments for them and I think it is cool how they can be used I liked the pygame stuff, because it was cool to see what I could do with games. I thought the creating your own game part of class was my favorite because you could constantly track the progress that you have made and get a sense of accomplishment when finishing a big assignment. I enjoyed creating games using Python, especially creating my own game with my partner. Regular expressions, they are very cool and super useful! Also it's fun to think of a bunch of corner cases and stuff Gaming - it made me feel like I was doing "real-life" application with my code. It is also just super fun! My favorite was URL opening and reading as it allows us to do really cool things I loved all of it but I guess my favorite was turtle just because I got to draw superhero symbols. I liked pygame because I play videogames a lot on my own, and can appreciate how they come together. Gamebox, it was fun to interact with the product of which one created. Also like turtle too, visualizing the work we did gamebox gamebox My favorite topic was the gamebox, because I enjoyed creating games. I preferred regular expressions because it made more sense to me and I found it to be relatively simple. In addition I liked how it was more like a logic problem. learning about loops My favorite topic in class was most likely the game design part just because it was the most fun and I saw the most results from that topic. The gamebox stuff was my favorite because it was interesting to see what goes into making games and it was fun to make our own games. My favorite topic was GameBox. I like how we can use knowledge we had learned throughout the course to create something unique ourselves. I liked learning about regular expressions. probably gaming--it was hard for me to understand but always wondered how they worked I liked learning gamebox and having a visual representation of what I was coding. regular expression Gamebox because we get to be use our creativity. I liked learning how to get information from real websites. My favorite topic to learn about was PyGame and Gamebox, because I like art and using these programs allowed me to make something using computer science. I can't choose! I liked all the loops and learning how to make use of them to make more complex algorithms. Turtle art Creating a set of loops My favorite is doing the games because it is something different and is definitely more satisfying to get a finished product. location finder

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	My favorite topic would probably be about the regular expressions; I found them very intuitive in understanding. Gamebox, because it showed me what I could do with the things I learned. Gamebox lecture because it allowed for creativity game building My favorite topic was learning how to use gamebox. The for loop and regular expressions My favorite topic in this course was creating games with Gamebox. Really interesting seeing how simple games like flappy bird actually are to create. Game coding because it is applicable in the real world My favorite topic was probably web reading python programs. I enjoyed the game project because it was very applicable and fun to create. pygame making functions - I found this useful The game project, as it was really fun to see your ideas become manifested onto the screen. I really enjoyed the first exam material and the regex material. The first exam material was taught in a very calm, relaxed manner, which made it easy to follow along. I found regex and its implications particularly fascinating. The gaming unit was definitely one of my more favorites and that is because ultimately I hope to go into gaming development some day. Creating games -- seems most practical Games I was always very confused as to how loops worked so this topic in the class really helped me. if statements - very versatile I enjoyed gamebox because it incorporated everything we learned. Learning about functions was my favorite part. They just interested me. I really liked the gamebox topic. I think the lab associated with it was a good intro to making games. I think further instruction would have made doing the game.py easier, but overall that project allowed for my own creativity to be utilized. Gamebox It was just nice learning python after already learning a bit about java. Also, liked making paper airplanes that one day. Turtle was a lot of fun (even though we didn't go over it for long) because of its flexibility and the creativity it allowed. I also found the gaming section really exciting, though python isn't much of a gaming program, because it was a really cool way of implementing what we learned and watching it come to life. The most enjoyable lectures were probably Tychonievich's general Q&A sessions. Gamebox and Pygame will be useful for me personally. I think there are a lot of fun applications for them. turtle.py gamebox- freedom and creativity for loops because they are very useful Topic - Gamebox - application of concepts learned and putting them all together I like the game topic because it allowed us to use what we know into a fun activity. The work has more of a reward compared to other assignments, and it was less frustrating. Loops because they were interesting I enjoy using Pygame because it allows for more creativity

print statements because after that it went downhill

I loved learning about the games and creating them. I would spend hours learning about them.

Gamebox- visually pleasing

gamebox was really cool because it gave us a chance to really get 'fluent' in a concept and apply it in new ways ourselves

Python

Probably gamebox? within my area of interest

I really liked the lectures about gamebox and creating our own game. It was the project that was the most rewarding.

I enjoyed the pace we were at for the beginning of the course. After that it seemed like we were going through material without enough explanation.

The games were really cool

My favorite was Loops (while and for) because it was fun to test, easy concept to grasp, and very practical in real life applications.

game

game

loops since it seems to be the most applicable

My favorite topic was learning about how to create games because it allowed us to be creative and unique.

Learning loops, because I think that it's an interesting concept

The regular expressions because it was fun to solve them (like a puzzle)

Definitely the game topics. I'm a gamer and I really learned to appreciate the work that goes into the game.

I enjoyed learning about gamebox - its amazing how many interesting code commands python has with gamebox to add detail and value to creative games.

learning about how to make games

My favorite was the gamebox section of the course. Learning new games as an application of coding increased the feeling of how important coding was in modern society.

My favorite topic would have to be GameBox, as I find game design to be very interesting, though not really the kind of thing I would like to pursuit. It's interesting to code rudimentary games.

Pygame - it was interesting how we could create games and apps using python

I liked the game stuff the most

I really enjoyed gamebox as a way to creatively integrate everything we learned into one assignment. The smaller programs we were assigned each week were also very interesting to write.

6. Which topic/lecture in this class do you think you will find the most useful in the future?

~

Question Type: Short Answer

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-001, Tychonievich, Luther

Total	Individual Answers
144	*See below for Individual Results*

just the basics, so the first half of the semester and a bit after

Basic things such as printing, making functions

lecture on compiling information

I think the regular expressions and unit on functions will be the most helpful.

Concepts of using code for future CS classes

Just the general stuff about CS. Types and stuff like that.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	Reading info from files or documents and getting info out of those. The data reading and analysis unit will probably be the most useful. Doing math with programming I think the whole class in general will be useful. Just the ability to potentially write programs for a job or something. It's hard to say which one will be the most useful because I'm not sure what skills I will use the most in the future regex maybe For/while loops. I think the topics of for loops and if statements will be most helpful to me in the future. Although I found them the hardest, I think regular expressions are the most useful because you can thoroughly search a website a find information without requiring much effort. regular expressions Creating functions regex regex For loops The math ones. Regular expressions were very interesting and helpful. I thought the salary assignment, although hard, was a very important skill that could be used in the future. Strings, tuples, ints, etc. for and while loops for and while loops Loops, this made all of our programs so much easier to do. working with lists Not a specific topic, but the way of thinking about CS and how to communicate with computers. It's the most difficult part and requires a lot of practice, but it can be applied to all languages and is a way of thinking/problem-solving that can be applied outside of CS, as well. Everything!! good start to learning about programming Functions the first couple of lectures that really taught the foundation of CS The mentality of coding and thought process. Creating games Pygame Looping in general Introducing functions earlier on in the class was very valuable to me. Other sections of CS 1110 last semester weren't introduced to functions until the very end, but I definitely appreciated learning functions sooner. I highly recommend keeping functions early in the semester. For loops were used in my statistics class too so that will be most useful. The general important concepts of programming: functions, loops, etc. reading data will be most useful in the future because I can use it for applications in data science and statistics. Probably the url stuff. Also learning about data types like lists and dicts. Probably anything with functions, list, loops, and using websites.

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	Probably writing texts but it was very hard.
	Regular expressions since there are so many useful things you can do with it, especially when it comes to working with the web.
	I will find the first couple lectures the most useful because it explained computer science, its importance, and the basics of how to program.
	interacting with data types
	Regular Expressions
	loops
	loops
	loops
	loops
	basics such as types, lists, for loops, etc.
	Regex because I want to do something with data analysis and manipulation.
	regular expressions because they can be used in a multitude of ways
	either URL or regular expressions
	reading websites and building games
	debugging
	I think a general knowledge of how websites work will be useful to me in the future.
	I definitely would find for loops most useful, or maybe lists/dictionaries.
	I thought that learning simplistic coding and regular expressions were useful.
	While loops
	file writing
	Regular expressions
	General syntax and statements that are prevalent through out computing, such as the for loop.
	The general understanding of how programs and computers are set up to retrieve and read information.
	Overall, I feel like the general concepts of coding will be the most useful as I move on to go into more advanced CS classes and learn other languages.
	I don't plan to continue with programming in the future, so I don't think I will really use any of this stuff.
	I believe all of the topics will be useful
	Csv files.
	Pygame and Gamebox because it utilized everything we had learned
	Loops
	I think just basic coding knowledge will come in handy. I can't really pick one topic because they all relate to each other in some way.
	Probably retrieving data from websites and stuff! Now i can write programs that actually get me the data that I want! In that same vein, the csv stuff is super interesting and useful
	Writing functions. There is just so much usefulness there, even in the simplest of programs.
	All of them are useful.
	the review lectures before exams
	I think that pseudo-code applies the most to everyday life and topics unrelated to Computer Science.
	Regex

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	Regex
	Regex
	Regex
	I think lists and loops will be the most useful in the future because I have already seen them been put to use in the workspace. I spoke with a computer science grad from UVA who is now in the work field and lists and loops seem to be the basis of his work.
	Basic concepts of building programs.
	The logical thought process required for problem-solving was really beneficial.
	I'm not sure which will be most useful in the future because I'm not sure how much CS I will use in BME, but I think a lot of things could be useful.
	I feel most of the course provided and intro to programming that can later be used for future CS classes.
	not sure
	just general knowledge of coding
	Regular expressions
	regular expression
	I believe that reading and using URLs will be very helpful in future applications
	Maybe regular expression or the game programming
	Regular expressions.
	Functions, regular expressions, and loops
	Regular expressions/writing functions
	for loops
	regex/ loops
	I assume that I will find file writing the most beneficial in the future.
	The regular expressions lecture.
	reading from data files
	My favorite topic would probably be the one about lists, just because lists are super useful in most of the tasks we had to code. They are much more applicable as opposed to things like tuples and dicts, but these are still essential to learn about.
	Loops, regular expressions
	loops and lists
	defining functions and variables because that is used all the time in computer science.
	Gamebox lecture because it allowed for creativity
	Just the general knowledge acquired of python some of the other more specific things will likely need to be brushed upon when I need to use them in the future
	All of it.
	Understanding data
	Understanding data
	Honestly just basic programming skills such as if statements, for loops, and while loops.
	regex if i ever do anything with computers
	creating algorithms
	I thought for loops and if statements were most applicable to every day coding.
	The info about urllib.request

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

All of it....

All of them

The while loop

functions and loops.

Probably data entry and processing

loops (for, while)

regular expressions and for loops/while loops

making functions, loops

The lecture on looping was very useful and wil1 be used in future CS classes for sure

Game design and mostly everything else

general basics of coding

Lists, dicts, re, and file-building: good for mass data management, which is very useful.

Python

lists

lists

Dont plan on continuing CS so none

Using python in general for simple tasks possibly. or learning new languages

I think all of the topics were very useful since they were all applied to homeworks or labs.

Same as previous

Anything that can carry over to other coding languages, such as R

I felt the whole class was useful. I feel that I don't know enough about other computer languages or computer science in general to answer this question.

I think learning about the math code was most useful.

I think probably the regular expressions will be really useful in the future

I think learning basic programming fundamentals such as functions and loops will be extremely helpful to me.

I think just general knowledge about it because I won't be using it a lot in the future for my career.

Probably regular expressions or lists

Likely just Python in general, as knowing how to code is a valuable skill, no matter what field of engineering you will go into.

I think everything will be helpful in the future.

if elif and else

Function building seemed the most useful to me. We used them in everything after learning how to use them, so I figure they are an integral part in any future coding I may do.

I'm already in my third year with two majors and a minor declared so I won't be continuing with the CS curriculum, but the class has definitely sharpened my problem solving skills. Figuring out the most effective and elegant way to write a program challenged me to think in new ways that I really appreciate.

7. What lecture/topic(s) in this class "did not work" or were not seen as useful in the long run?

~

Question Type: Short Answer

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-001, Tychonievich, Luther

Total	Individual Answers
131	*See below for Individual Results*

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	Pygame did not seem to be as useful to me, compared to the other topics.
	pygame did not seem very useful, but I do think it provided an interesting topic to students who needed it.
	N/a
	I feel like I would move the overview of the course to a later date in the class. There was a three day period where we went over everything we would be learning. I feel using this period at a later date for review of the final exam would be more useful to students.
	regular expressions
	The game.py is not very useful. Generally all lectures on the turtle and game.py were completely useless.
	I don't see myself using the file writing part of class
	Everything taught was necessary. I wouldn't have been able to make code without the topics.
	Homeworks were 10x harder than in class work, many times homework was too hard to do.
	The regex and writing files topics did not seem applicable or easy to understand in class.
	The turtle lecture seemed somewhat useless
	regular expressions
	regular expressions
	regular expressions
	regular expressions
	Debugging. I never really got a clear idea of how to apply this effectively to my own code.
	I do not think that the lectures "did not work" or not beneficial
	regex
	regex
	regex
	N/A
	N/A
	can't really think of one
	I don't think I got a full grasp on the the dictionaries and I dont think they were too useful for this course itself but it might be in the long run.
	none, they were all mostly helpful
	turtle... who cares about turtle
	There wasn't really anything that wasn't somehow useful, but the course started feeling overly difficult once the material got more complicated (beginning with loops). I think the main problem was we were sort of taught one thing in class and then asked to apply it in a different way for homework, and help wasn't always readily available. For example, we normally spent a lot of time in class talking about what something is and seeing examples that helped explain that further, but we didn't always talk about how to use and apply those concepts, or if we did it was very general.
	file writing and regex assignments were significantly (and unfairly) much harder than the material covered in class
	The gaming stuff was a lot to handle at the end of the course along with other difficult material.
	I don't think the course is very useful to those not majoring or minoring in CS.
	regexes seemed less useful to Non-CS majors
	The games, I'm sorry but I am really a math guy and that's why I loved and did well on the projects and stuff. But I really don't care about games and I found it very confusing and I didn't really see the point in it.
	the lab section was kind of unnecessary

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	hard to say as the topics from the lectures were all useful in some way when doing assignments and tests
	csv files
	I felt most were helpful
	None of the lectures were really useless, but the one that I would be likely to use least would be the turtle.py part.
	Regular Expressions
	Importing URLs and searching through the webpages
	file writing was taught too fast
	I thought dictionaries did not seem particularly helpful to me.
	Dictionaries in the lab assignment. As a topic dictionaries have been useful, but when they were used in very complex ways in lab it did not work very well.
	I found the lectures about turtle to be slightly confusing and not helpful to my understanding of CS. I would've prefered to jump right into coding.
	None
	None
	None
	None
	None
	None
	File writing
	Didn't really dive into the full practicality of reading writing new files
	File writing lectures seemed a little oversimplified based on the inventory assignment
	file writing
	file writing
	Code readability and elegance
	the first few lectures seem to be largely a waste of time
	None.
	The lecture on style and elegance, while not conceptually unimportant, seemed unnecessary as all the topics covered in it I already knew from previous lectures or from common sense.
	Inventory assignment and salary is too difficult to understand
	turtle graphics, but it was a good intro
	file writing seemed a little odd or a topic that is for a more advanced class. I don't know why we are learning at the end. I don't understand its use.
	Although it was interesting, I didn't think that the gamebox lectures would be that useful in the long run.
	code elegance
	Turtle lectures at the beginning of the semester.
	n/a
	n/a
	n/a
	Loops
	Regex.

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	I feel like everything is somewhat applicable
	the last few lectures on files
	some of the data analysis stuff
	Regular expressions and file writing. I won't discount either as I understand they are very important down the road, however I don't think there was enough time in the class to cover both of these topics and have the majority of students understand how to use them and what there importance is. I would suggest only covering one or the other in the future and going into more detail with it.
	I really hated Tester. I know it's an important concept and you need to be able to test multiple cases to ensure that your code works, but I didn't like the implementation of the assignment at all. I understand that the point of Tester was to get people to be able to test their own code, but there needs to be some feedback to tell people where they messed up or what they didn't take into account. When, after multiple submissions, people still consistently pass, say, around 8/10 tests, they need some sort of feedback that lets them know what they missed. Without that feedback, they are just getting wrong answers and they don't know why, and they don't learn from the experience. The feedback doesn't have to be immediate. I think that releasing the test cases that Archimedes uses once the late submission date for the assignment has passed and allowing students to see what they got right and what they missed would be valuable. I understand that this would mean that you need to alter the assignment for each subsequent semester, but I personally think that the benefits for the students would make it worth it. Tester could be a very useful assignment, but the lack of feedback to the students had the effect of creating a widespread distaste towards testing as a whole, as I'm sure you have, or will, find out when you read the rest of these evaluations. I highly recommend changing Tester to provide feedback for students.
	The last few weeks were just crazy and confusing
	regular expression
	I did not find dictionaries very useful. I still do not understand how to use them or when to use them.
	Turtle
	Turtle
	Turtle
	Personally, I don't find anything to do with pygame or gamebo useful, but I know others do/will
	I was a little confused during the first turtle classes, but looking back I think they functioned as the perfect intro to the class.
	I didn't see the particular utility of making such a rigorous final game project.
	application of dicts we`re discussed, but not thoroughly.
	none
	none
	The second exam material was very, very rushed, and I felt like I had to cram the material in my head.
	I don't think that anything we did with the games was necessary. I understand that for some people who are interested in becoming game programmers this might be exciting, but I found it rather useless and tedious.
	I think alla concepts were useful but had the most trouble with for loops and iterations
	I don't think the stuff we did with turtle was particularly helpful.
	None of them. Every topic has a use that is extremely important in its own way.
	turtle.
	The writing files was my least favorite unit because at times it just seemed over complicated for the simple task it does.
	I thought all lectures and topics were helpful.
	Regular Expressions. Confusing and not applicable to much outside of CS.
	the last few topics were really confusing for me (regex and files) and it was very difficult to get into the queue. as someone who does not want a CS major/minor, i felt it was not useful for me to know
	Regular expressions were extremely confusing
	I think the debugging lecture/lab was not a good lesson to a very useful and important topic.

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	Why in the world did we learn about files right at the end?? incredibly hard topic to work with, squeezed into 2 lectures. I didn't see the point of gamebox, but it was still interesting to learn about. Gamebox I found that the part on Turtle was not very useful due to the fact that it merely draws a picture. I cannot see how this may be very useful in the future. regular expressions because everyone said you don't use them much. The turtle lectures weren't really useful turtle turtle Unless you go into gaming it won't be totally useful but I really enjoyed doing it. testing/debugging was a bit harder to grasp, but I suppose it could/will still be useful in the future For me some of the really basic stuff at the beginning of the semester because I knew some of this prior. Hardware mentions of computers the debugging/dict lab was crazy hard to follow, a couple of the assignments seemed to go from 0 to 100 in terms of difficulty (lou's list, salary) Learning about turtle was not that useful, but it was also fun. Turtle, we never used it again. debugging lab lists Regular expressions and game code I think the regular expression topic was a little bit confusing and I don't see how useful it could be. the regexs was thrown together and I did not have a lot of time to understand it :/ Can't think of one that stands out, maybe file writing. Not having enough practice examples Well, I can see how pseudo code could be very useful but it seemed like after went over it we never really talked about it again and how to use it to our advantage. Maybe there could be a note above the assignments reminding students to write some pseudocode if they have trouble starting a problem, with the goal to be to break the problem down into more manageable chunks all on their own. all of the topics in this course seem like they could be useful in the future. Nothing in particular game None. Turtle did not seem very useful in the long run but it was an interesting intro. N/A all topics seemed of equal importance. None that I can think of game design maybe not necessary I did not find the topic of regular expressions very useful, but it was interesting to learn about. Personally, I found them confusing, and while I can see the utility, I did not enjoy the topic that much. The gaming portion

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

8. How would you rate the availability of TAs?

~

Question Type: Likert

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
177	2.97	0.96	53 (29.94%)	86 (48.59%)	21 (11.86%)	13 (7.34%)	4 (2.26%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
177	2.97	0.96	53 (29.94%)	86 (48.59%)	21 (11.86%)	13 (7.34%)	4 (2.26%)

9. How would you rate the helpfulness of the TAs?

~

Question Type: Likert

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
177	3.11	0.81	62 (35.03%)	77 (43.50%)	34 (19.21%)	3 (1.69%)	1 (0.56%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
177	3.11	0.81	62 (35.03%)	77 (43.50%)	34 (19.21%)	3 (1.69%)	1 (0.56%)

10. How often did you make use of the TA office hours?

~

Question Type: Multiple Choice

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-001, Tychonievich, Luther

Total	Every week (NA)	Every other week (NA)	Once per assignment (NA)	Rarely (NA)	Never (NA)
177	26 (14.69%)	32 (18.08%)	19 (10.73%)	55 (31.07%)	45 (25.42%)

Results for SEAS, 1000-level courses

Total	Every week (NA)	Every other week (NA)	Once per assignment (NA)	Rarely (NA)	Never (NA)
177	26 (14.69%)	32 (18.08%)	19 (10.73%)	55 (31.07%)	45 (25.42%)

11. Any specific comments about the TAs you would like to share?

~

Question Type: Short Answer

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-001, Tychonievich, Luther

Total	Individual Answers
83	*See below for Individual Results*

Sebastian was very helpful

Took three hours to be seen. Amount of TAs working should be increased on days leading up to major assignments/homeworks

All of them were just really good

no

no

no

no

no

They were generally very helpful, but for the last few assignments, I had to wait more than 3 hours to receive help. And when a TA finally came, they could not provide as much help as I needed because the queue was overloaded.

There were TAs who were indeed great with they're advise. They would look at my code and know what's wrong. But there are other TAs with bad attitude, obviously did not want to be there.

If the TA only comes to me for 2 min to "explain" during busy hours, I might as well stay at home and spent hours to figure it out. It was not helpful at all because the TA did not explain in terms of the level of understanding I have so I didnt understand the explanation.

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	the TA availability was ample until about the last three weeks of the semester. Possibly increasing the number of available TA's on certain days may further incentivize students to get the assignments done well ahead of time.
	Good TAs that know what they are doing.
	I really liked all the TAs. They were good both in office hours and in lab. Could you maybe hire more, though??? I need OH for every assignment, and end up having to dedicate my entire day to office hours because the queue lengths are always absurd and I get bumped to the bottom due to my frequency in OH.
	Sometimes the TA's seemed unfamiliar with the right strategies to use in the assignments, which was confusing and a tad frustrating after waiting for help and not always getting clear direction after talking to them. I think TA's should be required to know the assignments well when they are working office hours.
	They were very approachable, but I had a hard time following their explanations sometimes
	They were available a good amount of the time and they were very helpful on some assignments, though some were better than others.
	none
	none
	TAs were great
	TAs were great
	They were very good but sometimes hard to get.
	Available when I needed them. However, I could usually work around these hours.
	Good at answering questions and steering me in the right direction.
	I wish I had started coming to TA office hours earlier because they were pretty helpful.
	More of them! Office hours are really crowded. The queue is frustrating since if you seek help often you are penalized
	They were great
	Did not seem knowledgable
	N/A
	N/A
	N/A
	They are fantastic. Sometimes it would have been nice to have more available when big assignments were coming.
	they were talking in lab and never noticed when I had my hand up :(
	They were so helpful and nice!
	they were helpful and never really directly gave you the answers but rather they gave a thought process that led to the answer
	They are wonderful and so committed to their duties as TA's
	They were very helpful and informative during labs, especially when they reviewed and talked about commands before labs began.
	Very helpful in office hours and in lab.
	They offered good advice when they knew what they were doing--better at advising what to do than fixing your program.
	office hours were not handled well at all. Even when trying to come in early to get help on an assignment, I was not able to receive helpful advice in a timely manner. This greatly hindered my ability to learn and understand the material for the class.
	I believe all the TAs need to give similar advice for the same questions although I know that with the amount of both TAs and students, consistency of opinion and solutions can widely differ.
	They were really helpful and made sure you understood the material and what you were doing.
	The TAs helped a lot during the labs and readily answered questions I had.

large variability when it came to helpfulness. Some TA`s went as far as in-depth explanation, some seemed to want to move through students as fast as possible.

The helpfulness of TA's varied greatly. Some knew what they were talking about, and others had to refer me to other TA's because they were also confused.

Every TA was extremely helpful, however, the Queue's setup prevents them from reaching their full capability as a TA. I strongly think the system needs to reform.

certain TAs were extremely helpful (sebastian, david, anthony, monique), whereas others would barely help.

I found that some of the TAs were very helpful while others were completely inept. I went to office hours on numerous occasions and had TAs that didn't even understand the assignment or who just gave up on trying to help.

TA's weren't very knowledgeable with problems.

They were always very willing to help students out and encouraged them to come to office hours more.

The TAs in OH were very helpful when available. The Queue is not always manageable, but the TAs handled it very professionally

My TAs were very kind and helpful.

They really know what they are doing. Could probably teach their own courses if they wanted to.

I really can't judge the TA's help accurately. I felt like they had quick responses to emails and Piazza.

After a certain point in the course I essentially started going to office hours for every assignment, and it can be really hard to get TA helps sometimes. I really don't like the queue algorithm that shifts you back spots if you got help recently, sometimes I just had a quick followup question

Most of them were understanding and helpful, which was great. There should probably be more TAs though. In the more difficult labs we often had to wait a while for help, and the wait for help in office hours was often frustratingly long. There were a few times I had to wait an hour and a half when I only needed to spend about five minutes with a TA.

Sebastian is super nice and incredibly helpful. The other TAs are also awesome. One time though, I went to office hours with questions on an assignment and the TA wouldn't help me write code she simply just explained the assignment in different words which was very frustrating. I don't want someone to give me the answers, but I definitely need help starting sometimes.

Kamile and Winston were both super helpful. I filled out the TA eval saying as much.

Our lab TA's were the best. The TA's in office hours were mostly helpful, but many times left me thinking my code was working and everything was good, but then I'd fail some of the tests on Archimedes.

TAs were very helpful during lab.

In lab they could explain more about how things fundamentally work

The TAs are an amazing resource for students--and I do not take them for granted for a second. However, I would like to say that as a student who struggled with CS, I found the queue system for office hours to be very unfair. There were times when I would sit waiting for help for 2.5 hours on end--sometimes on the verge of tears--while people around me would come in and be helped after less than 15 minutes of waiting. I understand that the queue system helps promote the idea of pointed and specific questions, but I really don't think it's fair for students to be punished for having sought out help in the past. All students should be on an equal playing field. If you want to help promote a culture of pointed/specific questions, you should cap TA help at five minutes (or some other specific time you find to be the most fitting). The TAs could have an override button if for some reason they believe they should stay with the student longer (in situations like: if they haven't helped the student find the bug in their code). This would promote the same idea of specific questions without harming those who have proactively sought out help in the past. I'm sure there are many possible solutions to the issue of office hours that could help solve the problem, but that is one suggestion. I really hope you take this comment to heart because I know how much the system has affected not only me, but SO many students.

The TAs in my lab and that ran review sessions were extremely helpful.

Some were better at helping than others. They should actually try to help us instead of not really giving any useful information or not directly answering questions. There needs to be more working on days before assignments are due. The queue should also be first come first serve.

I find the queue system to be a bit absurd; it's almost as if you are being punished if you go ask for help, because you can't ask for help as quickly next time. That is a disincentive to go ask for help. Though, I do see the merit in it preventing people from hogging the TA time and only asking important questions. However, I think a first-come-first-serve function should be added to the algorithm of calculating queue status.

The TA's were great.

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

The TA's are super helpful! I think the hardest part about office hours is not being able to get off the queue in time.

Kamille was great - she brought us cookies and was super smart.

William Greyeski was very good

No

No

No

The Tas were okay, but the way office hours were set up were terrible. You shouldn't be penalized for getting help early in the semester when, it's later in the semester and now everyone is trying to get help

I think that many of the TAs were very motivated and tried really hard to solve student problems, but some of them were very determined to figure things out on their own when collaborating with another TA could have made the process a lot shorter (so they can help more people). Don't underestimate the power of TA collaboration or handing off problems to another mind!

n/a

I received extremely conflicting advise from TA's, much of which I later found out was wrong. Vagueness in answering questions after you've waited for 45 minutes is very frustrating.

The weighted queue punishes people who utilize office hours the most - discouraging such use and results in sitting in stacks for 4 hours without receiving any assistance

Some of the TAs were really dismissive, but most of them were very,very kind and helpful. Could not have gotten through this class without TAs. However, the queue system was the most inefficient system I have ever had to deal with. For the last two assignments, I was on the queue from 3-9 pm and only got help from 8:45-9. Also, the TAs who did come by in that time period had no idea why my code wasn't working.

The TAs in my lab section were extremely helpful throughout the semester.

All the TA's are very approachable and help a lot.

I loved the TAs in my lab and they were very useful, but office hours were a nightmare. I would wait over an hour to get two minutes of feedback for a small issue. There were too many students and not enough available TAs.

Loved having my TAs Gabriel (Trad), Claire, and Maria

I had really good TAs so I felt like that was a lot more helpful than lecture itself

12. The course addressed technically rigorous subject matter consistent with the course objectives.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
176	4.29	0.66	66 (37.50%)	96 (54.55%)	11 (6.25%)	1 (0.57%)	1 (0.57%)	1 (0.57%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
1621	4.10	0.80	492 (30.35%)	882 (54.41%)	155 (9.56%)	59 (3.64%)	18 (1.11%)	15 (0.93%)

13. The instructor used methods other than/in addition to traditional lectures (for example, active learning, in-class problems, collaborative learning, in-class discussion) effectively in this course.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
175	4.34	0.78	85 (48.57%)	68 (38.86%)	15 (8.57%)	4 (2.29%)	1 (0.57%)	2 (1.14%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2335	3.75	1.13	591 (25.31%)	884 (37.86%)	361 (15.46%)	195 (8.35%)	126 (5.40%)	178 (7.62%)

~ QUESTIONS AND DETAILS ~ | *~ ANSWER MATRICES ~*

14. There was a reasonable level of effort expected for the credit hours received.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
176	4.14	0.96	71 (40.34%)	78 (44.32%)	11 (6.25%)	12 (6.82%)	4 (2.27%)	0 (0.00%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
1624	4.05	0.89	493 (30.36%)	873 (53.76%)	138 (8.50%)	80 (4.93%)	38 (2.34%)	2 (0.12%)

15. The homework assignments helped me learn the subject matter.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
175	4.27	0.92	87 (49.71%)	62 (35.43%)	15 (8.57%)	8 (4.57%)	3 (1.71%)	0 (0.00%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
1624	3.84	1.09	481 (29.62%)	660 (40.64%)	224 (13.79%)	146 (8.99%)	71 (4.37%)	42 (2.59%)

16. The textbook increased my understanding of the material.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
176	3.47	1.01	25 (14.20%)	57 (32.39%)	54 (30.68%)	19 (10.80%)	6 (3.41%)	15 (8.52%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
1622	3.35	1.13	167 (10.30%)	275 (16.95%)	331 (20.41%)	126 (7.77%)	72 (4.44%)	651 (40.14%)

17. The course material was well organized and developed.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
176	4.24	0.84	75 (42.61%)	80 (45.45%)	12 (6.82%)	7 (3.98%)	2 (1.14%)	0 (0.00%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2334	3.95	0.91	606 (25.96%)	1058 (45.33%)	333 (14.27%)	133 (5.70%)	38 (1.63%)	166 (7.11%)

18. The instructor was knowledgeable about the subject matter.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
174	4.77	0.44	135 (77.59%)	38 (21.84%)	1 (0.57%)	0 (0.00%)	0 (0.00%)	0 (0.00%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2338	4.33	0.73	981 (41.96%)	973 (41.62%)	172 (7.36%)	32 (1.37%)	12 (0.51%)	168 (7.19%)

~ QUESTIONS AND DETAILS ~ *~ ANSWER MATRICES ~*

19. The instructor was well prepared for class.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
174	4.61	0.57	113 (64.94%)	53 (30.46%)	7 (4.02%)	0 (0.00%)	0 (0.00%)	1 (0.57%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2335	4.29	0.73	908 (38.89%)	1037 (44.41%)	179 (7.67%)	35 (1.50%)	11 (0.47%)	165 (7.07%)

20. I received adequate preparation from the prior courses in the curriculum to be successful in this course.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
175	3.60	1.13	25 (14.29%)	24 (13.71%)	27 (15.43%)	13 (7.43%)	3 (1.71%)	83 (47.43%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
1621	3.66	1.09	242 (14.93%)	352 (21.71%)	246 (15.18%)	104 (6.42%)	42 (2.59%)	635 (39.17%)

21. The grading policy was fair.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
176	3.94	0.97	53 (30.11%)	81 (46.02%)	24 (13.64%)	15 (8.52%)	3 (1.70%)	0 (0.00%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2335	3.79	1.03	517 (22.14%)	905 (38.76%)	364 (15.59%)	183 (7.84%)	72 (3.08%)	294 (12.59%)

22. The instructor responded adequately to in-class questions.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
175	4.50	0.63	98 (56.00%)	67 (38.29%)	7 (4.00%)	2 (1.14%)	0 (0.00%)	1 (0.57%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2339	4.14	0.84	766 (32.75%)	1053 (45.02%)	216 (9.23%)	86 (3.68%)	24 (1.03%)	194 (8.29%)

23. The instructor effectively used technology in support of the learning goals for this course.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
173	4.61	0.58	113 (65.32%)	54 (31.21%)	5 (2.89%)	1 (0.58%)	0 (0.00%)	0 (0.00%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2324	3.89	1.00	633 (27.24%)	893 (38.43%)	370 (15.92%)	170 (7.31%)	52 (2.24%)	206 (8.86%)

~ QUESTIONS AND DETAILS ~ *~ ANSWER MATRICES ~*

24. The average number of hours per week I spent outside of class preparing for this course was:

~

Question Type: Multiple Choice

~

contributed by Office of the Provost

Results for CS-1110-001

Total	Less than 1 (NA)	1 - 3 (NA)	4 - 6 (NA)	7 - 9 (NA)	10 or more (NA)
174	1 (0.57%)	43 (24.71%)	88 (50.57%)	32 (18.39%)	10 (5.75%)

Results for SEAS, 1000-level courses

Total	Less than 1 (NA)	1 - 3 (NA)	4 - 6 (NA)	7 - 9 (NA)	10 or more (NA)
1621	176 (10.86%)	765 (47.19%)	520 (32.08%)	122 (7.53%)	38 (2.34%)

25. I learned a great deal in this course.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-1110-001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
176	4.32	0.75	81 (46.02%)	75 (42.61%)	15 (8.52%)	5 (2.84%)	0 (0.00%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
1620	3.87	1.06	492 (30.37%)	688 (42.47%)	242 (14.94%)	135 (8.33%)	63 (3.89%)

26. Overall, this was a worthwhile course.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-1110-001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
175	4.23	1.00	89 (50.86%)	56 (32.00%)	17 (9.71%)	8 (4.57%)	5 (2.86%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
1619	3.71	1.20	497 (30.70%)	560 (34.59%)	274 (16.92%)	177 (10.93%)	111 (6.86%)

27. The course's goals and requirements were defined and adhered to by the instructor.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
175	4.39	0.71	86 (49.14%)	77 (44.00%)	8 (4.57%)	3 (1.71%)	1 (0.57%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
2327	4.04	0.85	678 (29.14%)	1222 (52.51%)	323 (13.88%)	54 (2.32%)	50 (2.15%)

28. The instructor was approachable and made himself/herself available to students outside the classroom.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
172	4.02	0.88	60 (34.88%)	65 (37.79%)	38 (22.09%)	9 (5.23%)	0 (0.00%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
2325	3.83	0.96	612 (26.32%)	943 (40.56%)	589 (25.33%)	122 (5.25%)	59 (2.54%)

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*

29. Overall, the instructor was an effective teacher.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-1110-001, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
174	4.39	0.77	93 (53.45%)	60 (34.48%)	18 (10.34%)	2 (1.15%)	1 (0.57%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
2334	3.87	0.97	647 (27.72%)	1002 (42.93%)	494 (21.17%)	128 (5.48%)	63 (2.70%)

30. Please make any overall comments or observations about this course:

~

Question Type: Short Answer

~

contributed by Office of the Provost

Results for CS-1110-001

Total	Individual Answers
75	*See below for Individual Results*

Although I struggled through the class, it was because I was naturally not good at solving logic problems. However, I found that it provided a good idea of what CS is, which is useful for people like me who took the class to find out if it would be suitable for us as a major. However, for those who do continue in CS, I have heard that the transition from Python to Java takes a heavy toll on students.

Super fun, really cool people involved. Hated the game part though.

I enjoyed learning about Python and all its functions such as regx and user input commands.

I absolutely did not enjoy the last month of this class. The rigor of the assignments was somewhat reasonable, however the system of TA office hours made it much worse. If there will be difficult assignments that obviously require TA assistance, the system of the Queue must change. There is no reason for a student to wait in office hours for nearly 6 hours only to receive help once. Also, the aggregation of difficult assignments in addition to the game project near final exams was not timed well. Please reconsider the latter portion of the organization of this course, and please reform TA office hours.

interesting, but could be frustrating when you do not have interest, as it is a required class for first year SEAS students. Probably more likable for someone intending for CS major/Comp Eng.

Professor Tyconievich was awesome, and his lectures were engaging and informative.

Good professor, but I struggled with understanding the difficult concepts even with going to office hours

I thought that at times the homework assignments were pretty difficult and going to office hours didn't help because so many people had shown up. I think there should be a better system for how office hours work or maybe have more ta's come to thornton the day before these hard assignments are due so that everyone that needs help can get it. I also thought that some of the labs weren't very helpful and were a waste of time. Other than that, i thought this was a good intro course and I feel that I got a good foundation for cs.

This course made me want to continue studying CS

Professor Tychonievich is a great lecturer, his lectures are very structured so you know exactly what to expect and he does a very good job of understanding what people are trying to ask and accurately answering their questions.

learned a lot

Enjoyed it a lot, excited to take more CS classes and see where it takes me

In lecture it seemed as if Luther was showing off to the point it took away from our learning. We would fly through examples and not really discuss them. Office Hours were awful. The fact that so many people had to go to office hours were indicative that we did not learn enough during lecture, and then the queue was so backed up that there were a few times I went in there and was on the queue for over two hours and was never helped. When you would meet with a TA they were helpful, but almost every time one of them helped me through the homework, they would have me do a process that was not mentioned in class. This was particularly frustrating because it seemed like I had to go to office hours to complete the homework. Also the fact the the TA's had to practically spoon-feed students the answers took away from our learning.

Covering the material for the homework assignments two or three lectures ahead of when they are due would facilitate timely completion of the homework.

Make optional projects that teach more advanced topics, not thought questions, but actually topics

Great course!

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	assignments on average were much more difficult than the examples in class, this seemed unfair as it was often so difficult and time consuming to get adequate TA help at office hours. The course started off very easy, but grew very difficult towards the end. If the difficulty could be better distributed throughout the semester, this would be very helpful for students. I had difficulty as I have no prior programming experience. Overall this was a wonderful class and I have made significant improvements in my skills as a programmer. My only problems are that in the middle of the semester, the lab and homeworks assignments got significantly more difficult out of nowhere. At times the problems we would have to solve were extremely vague and required complex code to complete. Even with the assistance of one of my friends who did really well in 1110 last semester and is in 2110 now, some of the assignments could not be solved. The more difficult problems lead to a large influx of people to office hours which prevented the TA's from being able to provide the help that people needed. I took this course as a student with no programming experience. While I tried to take 1112, I am overall happy with ending up in 1110. I'm not looking to pursue any further study in CS, but I'm very happy I got a strong, broad intro to it. My one criticism would be the the first 6 weeks of the course went at a relatively slow pace and were very straightforward, but then the pace picked up exponentially and the assignments and material became a lot to handle. Many students who did not have at least a semester or year of programming experience from high school under them really struggled. Exams were incompatible with how programming challenges would be like in the real world, did not adequately show student's understanding of the material. Practice exams/problems would be very helpful. none none Really interesting and fair course. I like how programming assignments make up a large portion of the grade, rather than exams - assignments reflect a more realistic environment for programming (open note) rather than memorizing details and nuances of code (however, exams were well designed for the topics) Well designed class, Prof. Tychonievich knew a lot about random things as well, and was great at improvising funny examples. Luther is an awesome professor and person in general. The tests did not really test ability to write code- simple memorization would get a passing grade Working through problems/assignments together in class would be helpful Computer science is definitely not my strength and now I know that. However, I did find some of the assignments to be incredibly time consuming and confusing. In addition, the use of Arhcimedes often caused more issues that it was worth in my opinion. Great professor and probably one of the best professors I've had during my time at U.Va. I thought that the Professor provided very good lectures and encouraged students to get help from the TAs and to not procrastinate on their assignments. I do not think any other professor could introduce CS any better than Professor Tychonievich. He started off from square 1 which really allowed me, someone with no prior computing experience, to actively learn computing. He taught in a way that gave everyone the same opportunities. His passion for CS is evident and I appreciate the atmosphere he creates in a big lecture hall. The grading was super unfair because I could understand the concept and get the general idea but then make one mistake and loose 15 points. That angers me so much. Why do you punish kids who need more help by making them wait the longest? This seems counter-intuitive and discourages me from taking other CS classes, even though I think they'd be useful as I will be pursuing computational chemistry in grad school. Hire more people, make the programs easier, or do something else to help alleviate the demand of OH. I think that the homeworks should be due at midnight instead because sometimes people have exams the same day as the homework is due and they can't fix it. I also think that the office hours system needs to be fixed. I don't think that it is fair that you get put lower down the list the more help you get. I feel like this is reprimanding you for getting help. Also some of the assignments required us to do things that we were not fully taught and the TAs could also not figure it out sometimes and I don't think that this is fair as students spent hours trying to figure out the homework and could not get help because of the office hour system even if they started the homework early. The class is excellent with Tychonievich and is very worthwhile for any engineer to take, as it is valuable for any engineering field. the homework assignments and ESPECIALLY the labs were unfairly hard. This was a fun class. N/A N/A

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*

It may be because Computer Science is difficult for me, but I found every homework assignment after for loops extremely difficult and spent hours upon hours on a single assignment just to not finish.

I very much enjoyed this course. I had no prior experience in Computer Science but I now am taking another Computer Science course next semester. The course well designed and I would recommend it to anyone curious about Computer Science.

The office hour queue system is not effective. In addition to other parameters, the queue should include how long you've been waiting. Students do not have equal capabilities in learning CS so it is not fair that someone who got helped more recently can wait for hours while several people come in and get help immediately.

This introduced me to a whole new field that I had no prior experience in and I am now considering pursuing a CS minor.

The class was a very good introduction to python.

I loved the course, Prof Tychonievich was an incredible lecturer. I hope he continues to teach the course in the future.   By providing recordings of lectures, he really offered students an ability to further their knowledge. Highly recommend him and his course to anyone wanting to learn Python.

Some of the homeworks near the end of the year were a bit more challenging than they probably should have been.

Maybe go over code for homework assignments; like the proper code of how one would do it so we can learn from example, maybe, of how one would use a certain function or something.

The end of the course seemed way too hard for the 1110 level. Many of the TA's did not understand what was going on and therefore could not be useful. I found that Tychonievich had 'fun' teaching us stuff we didn't have to know and may have pushed the class too fast in the last few lectures. It felt as though he thought we had a better base than we did at the end.

Seems to be a good introduction to many topics and techniques but I think more detailed talk about data- types and other things like that would help in the long run for understanding

For being at such a high level of knowledge and having such a strong understanding of computer science, Tychonievich did a great job of making the basics of the discipline accessible. I have already recommended him to friends for this coming semester.

Luther did a very good job at teaching the material using examples that made it easy to understand

I believe that the course got too difficult much too quickly for students not planning to major in CS or students with no prior knowledge of CS and Python.   I believe that there should be more consistency with the difficulty of the homework assignments. Some were easy and could be finished in an hour while others were incredibly difficult and took hours to finish. I believe that the students should be alerted when the homework assignments are going to be marginally more difficult.

gave me a good idea about what cs and programming is

Great course with an even greater teacher. I didn't think I would major in anything computer science or computer engineering related, but after taking this course I knew I wanted to do computer engineering. Best class taken so far.

The queue system is unfair. It should be based on first come first served instead of moving people who have never been to office hours before, in front of the line.

This course helped me decide to minor in CS. That wouldn't have happened if I hadn't had a good experience in this course.

Great course for students interested in CS

This course clearly favored those with prior programming experience. My friends and I came in with no experience whatsoever and we all struggled to get assignments done. The handwritten exams are a terrible idea. Code isn't supposed to handwritten and in only 50 minutes, the tests required a ridiculous amount of work. I can't do well on them if I can't test out my code to see if it works.

TAs should not tell you your code works fine when it does. I was very frustrated because this happened to me and I found it discouraging and did not go to office hours anymore.

This class really made me love CS

The class seemed much harder towards the end than a basic intro class should have been. Even TAs who were about to graduated with a degree in computer science were not entirely sure how to help students in the last few weeks of class

Professor Tychonievich is very, very smart professor, and its very evident that he's prepared for every lecture and knowledgeable enough about the material that he's willing and able to answer any questions you may have. However, the end of this class was frankly absurd. During the beginning of finals week, we had a game project and 2 very arduous CS assignments due. We were incredibly overwhelmed towards the end of this class. After the first exam, there was a massive learning curve that was unhelpful. The class material was interesting, but it was so rushed towards the end that I don't know if I would major/minor in CS ever.

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	I absolutely loved this course. It made me want to major in computer science. This course was taught well, although the assignments took too much time after halfway through. Wish the Office Hours system was much better. I would spend 3-4 hours waiting for help that took only 5 minutes. The harder assignments were due too often, which was a pain for me to do thoroughly and sit through office hours for help. Otherwise, content was very interesting. Have recommended this course for everyone because everyone should take a little CS, and Tychonievich does a great job at making CS accessible to beginners. I love Tychonievich, he's awesome and clearly knows what he's talking about. He's engaging and transparent I know that I am not good at computer science and simply cannot grasp the material like other people can. I was doing fine in the course, but the second exam was a disaster. I think the course would be more effective if the exams were replaced with projects like game.py In the working world, we simply wouldn't write code on paper Challenging, fun, and inquiring! I really enjoyed this course, and am planning on taking more CS courses because of it. I really appreciated the lectures being posted online; it made the hugest difference in my time management in a great way. My only complaint is that I submitted a regrade for my first exam, and mentioned the one problem I wanted re-looked at, and instead I was docked points in a different part of my exam, which I do not think was fair. I enjoyed the class. 3 hw a week was hard and I would've coped better if it was 2. I wish the project was worth more since that was the hardest I've tried in the class. My grade sadly doesn't reflect that. The written code part on tests I think aren't very useful. While it definitely tests knowledge, the written part misses out on the testing and decoding that a usual program would have. It's hard to come up with everything on the fly. We should not have to take tests on paper because we don't code on paper. Tests should be on the computer. Work on more examples in class that we may see on the tests (practice problems) Tychonievich may be the most fun lecturer I've ever had. I thoroughly enjoyed this class and I have nothing but praise for the professor! I really enjoyed having Luther as a teacher. He was very friendly and knew the material very well. I have never taken a cs class before but I am now thinking about majoring in it because I enjoyed the class so much. I felt frustrated by office hours, specifically how students would monopolize TAs, thus taking their time away from other students. Additionally, I felt that after spring break, the assignments often contained pieces that we were not prepared to implement on our own. All preferred comments and changes were mentioned above. The office hours setup was absolutely ineffective. I often spent hours in the queue and was never helped, leading me to not understand the material and do poorly on the assignments. This should be addressed first before any concerns as it really made my experience in the class miserable