CS 1110-001 **Introduction to Programming** - Spring 2018

ENGR (17252)

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

Respondents: 120 / Enrollment: 239

Summary: CS 1110-001 Introduction to Programming - Spring 2018 (17252)

Overall Course Rating

CS-1110-001 Mean 4.03
CS-1110-001 Std Dev 1.02
CS-1110-001 Response Count 594

SEAS, 1000-level courses Mean 3.88
SEAS, 1000-level courses Std Dev 1.00
SEAS, 1000-level courses Response Count 8490

Overall Instructor Rating

INSTRUCTOR: Tychonievich, Luther
Mean 4.52
Std Dev 0.70
Response Count 830

SEAS, 1000-level courses Mean 4.14
SEAS, 1000-level courses Std Dev 0.88
SEAS, 1000-level courses Response Count 19434

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

1. What would you suggest we change about this course in the future?

~

Question Type: Short Answer

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-001, Tychonievich, Luther

Total	Individual Answers
103	*See below for Individual Results*

More time on the midterm exams

Provide better

The amount of information at the end of the semester.

Space out homework a bit more and make exams a bit shorter

The programming assignments went from being fairly simple to very challenging. Towards the end of the semester I felt that I was spending a lot of time on PAs and having too many due in a short amount of time. Maybe progress the difficulty a little more gradually.

introduce the debugging strategies earlier in the class, did not need as much intro in the beginning, more time spent on some of the basic ideas to have a strong foundation

The timing with lectures and some of the PAs made it difficult to try and work ahead (get them done on weekends for the week)

It seemed to move very fast at the end of the semester.

PyGame is fun and increases the understanding of programming. However, gamebox was rather dull to use. But it was easy, which I believe is the reason for the dullness.

More practice with coding, maybe optional assignments?

More assignments about the last few topics of the class (they really helped me to tie the class together and see potential for utility for a non-CS major)

The assignments started off really slow and elementary and then increased in difficulty every rapidly, it would be more effective if there was a gradual increase in difficultly. In other words, if the assignments started off a little hard than they did, it would not be as much as a shock to students.

Homework are much difficult than what we learn in class and what is on the exams. Make homework assignments more relatable and easier.

use laptops for test codes so we can test them

Nothing

Nothing

Nothing

Everything in the course seemed to be effective in teaching the basics of programming. The only thing I might have preferred is following the homework more closely in lecture and lab.

Change the times of homework deadlines to be at or around midnight to help with time management.

Sometimes labs are not helpful

N/A

N/A

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	N/A
	N/A
	I have no suggestions for improvements.
	Take away the two hour wait limit on test failure. Making it two hours does not help one improve their program if they do not know what is wrong with it.
	Give students practice exams that are off the same caliber of the tests. The tests were way harder than the practice exams and students therefore did not have a good way to prepare for the exams.
	Nothing. Itâs great.
	More in-class practice problems
	Coding on the computer instead of on paper
	Maybe consider having office hours on Fridays, even if they are shorter (like from 1 - 3). For PA 19: Flappy Bird, a rubric would have been nice. It could explain the expected level of difficulty.
	Provide more in class practice of material, before testing the concepts directly on the exam.
	Talk about how to move a program from python to your desktop or to the play store/app store
	not much, not paper exams though
	Format of the exams
	Possibly the teaching/lecture style of the professor. I found it a little difficult to follow along during lectures.
	More continuity across the sections
	decrease PAs in the second half
	lab time set aside to reteach topics
	More detailed instructions for homework assignments.
	I personally wish that there was a little more structure to the class lectures such as having powerpoints with important definitions on them.
	I dont know why, but this class was easy until the withdrawl deadline and then became much more difficult.
	Very good course, organized, fun. Nothing to change
	smaller class sizes, instead of a larger lecture
	I really liked how Professor Tyconovich went about teaching this course - everything was clear, easy to access, study guides were good, the TAs were awesome, and I really have no complaints about the course in general!
	Can't think of anything I would change!
	Nothing.
	Labs started off fun but kinda got tedious and boring towards the end
	n/a
	n/a
	Make the increment of difficulty of the course more gradual. The course was very easy up until the drop deadline. It seemed that almost overnight the PAs got harder, which was a surprise to many students.
	I wouldn't!
	Honestly nothing this class was a lot of fun and I learned a lot.
	More focus on projects and assignments, less focus on tests.
	nothing
	nothing
	Include iclicker questions for practice!!!

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	the grading system? like i got 0s on assignments that i did?
	Shorter exams for the midterms because the time constraint is very difficult.
	I would pace the course more evenly, as it was too slow in the beginning and too fast in the end.
	I think an improvement would be more sample code to help guide students for assignments or all kinds, giving them more test cases to try where they know what the result should be and to show them how they should be thinking about test cases.
	If there is ppt and structured outline for the lecture on a specific day it would be better.
	Could you go more in-depth on regular expressions, please? This topic confused me the most throughout the entire course
	More instruction towards the end of the course with regard to writing files.
	Nothing much. I loved the setup of this course and would't change a thing.
	Better grading policies. Much of the grading seems arbitrary and sometimes unreasonable. For example, I can spend hours working on an assignment and have more than half of the code correct and receive a grade of zero due to "insufficient evidence of attempt." In this class, every point counts.
	TA's should be more available and better at communication, especially at the beginning of the class. Our TAs left us to our devices immediately after class was supposed to start.
	I suggest there be fewer lectures set aside for introducing all of the concepts. I found it somewhat hard to follow because I didn't know the concepts in-depth. I also found them somewhat tedious.
	Maybe more practice problems with solutions
	I think some of the labs were relatively difficult to solve in the time period allotted. However, since they were graded on participation, it worked out fine.
	The first exam was rather too specific - I feel like it tested things that were rather unnecessary in the grand scheme of things. I suggest learning more topics we could get tested on before the first exam.
	The two hour wait time after submitting a code seems excessive to me. I would change that in the future. I also think 50 minutes is not enough time for the exams. Most people I talk to knew the answers to the questions but because they were rushed, they either never got to them or made mistakes. I think if an exam is meant to test whether we have a solid understanding of the concepts, the time for the exams should be extended.
	I think that a concise list of commands, etc. learned in each class would be helpful when completing homework
	I don't have any suggestions
	I suggest that you make lectures a little more clear and talk more about specific programming examples than abstract concepts.
	Have more engaging opportunities during lecture
	Some of the weekly assignments were just about impossible without TAs and towards the end of the semester we would have like 2 really tough ones in a week which got overwhelming.
	Not sure, the current course seems to work pretty well and I don't see many areas with need for improvement
	Potentially don't spend as long going over wrong things when writing the code, it can sometimes be confusing
	Make lab attendance optional, as long as the assignment gets submitted at the end of the day then it counts.
	Make lectures mandatory?
	I think the structure is good
	The pacing of the course seemed to change abruptly right around the drop deadline. I wish it had built more gradually.
	Lab could slightly more directly apply things taught in class, such as the decode lab, etc., which did not do so as smoothly as others, but there are very few of these instances so its almost negligible.
	Maybe after submission for a homework is no longer possible, posting an example of an efficient solution, that follows the route that was sort of expected. There were a few times where I made code that worked, but I was ensure if it was the best approach to the solution. Seeing other possible approaches could of helped with further understanding beyond my own personal approach.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	There should be more interaction during class time, so students feel the need to show up to class and pay attention. Decreasing the auto feedback time by an hour, so instead of waiting 2 hours we only wait 1 hour. I understand the need for the delayed feedback, but it can be really stressful during busy times when the feedback comes in 2 hours instead of 1 which is more manageable. Also, if the course continues to have a game project it might be a good idea to play with the schedule, so people have more time to create the game. Because right now, its due among finals time and it is really stressful. Have the lectures directly assist the students with completing homework problems, as I found myself having to constantly go to office hours needing help with programming assignments. Better preparation for the first exam. Hold Office hours at later or more varied times. Out of 30 hours, I could attend a maximum of 3. Make the exams like the practice exam difficulty wise Have more informative automatic feedback for the programming assignments - naming each automated test for example. Easier homework assignments, or grade on completeness for the harder ones Maybe have a minimum amount of attendance be mandatory (aside from tests). I am not sure if all of the exams are weighted evenly, but I think that the final should be weighted more. Also, I feel like it is too easy to lose large number of points on the exams and homework for missing very small things, but maybe that is just the nature of the material. This class was far too fast paced, and the assignments were very difficult. Making them easier would help reinforce the basics and give the students more confidence. There were a few times that both TAs and I were unsure why certain code was not working for homework problems, and they could not see the test cases. If test cases for available to TAs, they would likely be able to help students more. I think the best way to change the course is to adjust the assignments given for homework; if some codes were given to test skills like debugging or logic would help on the tests. I feel as though people might take advantage of the 100% participation lab grade. They should definitely mostly participation, but I don't think just submitting a poor attempt should warrant a 100. I really liked the course the way it is Regular expressions have been incredibly difficult. The course may be better if these are left out of the required material. The assignments relating to them seem too difficult for this course. maybe have parts of the class dedicated to working on the PA's due. Maybe giving more general guidance as to how to approach the problems at the beginning of class would be useful. Also extending TA office hours after 5 would be great since people often have class until 5 or are busy until then.
2. 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)
120	3.46	0.71	68 (56.67%)	41 (34.17%)	9 (7.50%)	2 (1.67%)	0 (0.00%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
233	3.45	0.70	129 (55.36%)	82 (35.19%)	19 (8.15%)	3 (1.29%)	0 (0.00%)

3. 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
103	*See below for Individual Results*

For me, mostly all worked, but the gamebox lectures were way too fast and confusing. Was better to just look at the code for these after.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	Although I recognize the important of regular expressions and HTML, the assignments dealing with them were quite difficult. These topics may not be appropriate for this introductory course.
	I didn't like starting with the turtle programming
	they were all ok
	I thought most things in the lecture were useful as it helped people learn how to use Python.
	I didn't find the game particularly useful
	gaming
	There is nothing I found that was not useful.
	In class we covered the differences between list and tuples, however I still struggle with defining the two. I think it would be beneficial to do side by side examples to see why a tuple or list should be used for different cases.
	Turtle
	Turtle
	Turtle
	Gamebox seemed relatively un useful but it was fun
	None, maybe gamebox but it was interesting
	some lab assignments
	I feel that all the topics were relevant.
	The game project is great, but I felt like the pace of the class went too quickly and that I didn't really learn much/understand why we were doing it
	All of the topics were helpful but some of them did not appear in assignments.
	As much fun as could be had with it, Pygame's usefulness is ultimately a bit questionable being that it is used so exclusively.
	I think that regular expressions just sort of generally tripped me up - I could see their usage pretty clearly but their form and usage were just a little weird to get used to for me.
	none
	none
	none
	none
	I found regular expressions and lists/dicts really difficult. I think regular expressions did not work well.
	Programming games
	I dont plan on using code in my future
	we could have spent a little less time on pseudocode
	regex
	Initial topics of conceptual coding
	im not entirely sure how regular expressions are useful for general coding.
	Much of what we learned stacked cumulatively and was used in later homework assignments.
	N/A
	N/A
	N/A
	N/A
	gamebox unit was cool, but I most likely won't use it again past this class or in other classes

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	Turtles Very useful. Pseudocode. Nothing, really. I would have liked a different homework assignment for dicts - the assignment itself did not make sense to me, which made it hard to practically apply the concept of dicts. gambox; regex nothing really - Debugging - Turtle - Pseudocode I did not see the point of learning turtle seeing that it was only used for the first week and we were not tested on it. I do not fell properly equipped to answer this question as I do not know enough about higher level computing to know which topics are and are not useful. Gamebox was interesting but I don't think it was useful in the long run. Because gamebox is not actually used in the real world, I feel like we should not actually have learned it. I think they were all beneficial in some sort of way Turtle? I really liked learning how to use it and I wrote my first complex program to use Turtle but I don't really see the long-term uses of it (it wasn't even on any exams?) I wasn't a huge fan of Lab 8: Cryptography. I didn't feel like turtle was very helpful in the long term GOod Idk The lectures on debugging and test case selection didn't have enough information to be very helpful. every concept seemed to be built upon on Every lecture was quite successful. It is sort of like an obligation to rewatch the lectures online if one wants to fully understand the material. Go to lecture for the main concept and watch the online lectures for details. Turtle did not seem very useful, but the rest was good. Possibly spent too much time doing Turtle, but this was useful in getting students to practice coding and get comfortable with PyCharm. The turtle programs turtle turtle turtle Regular Expressions They all helped me learn cs so I guess they were all useful. I felt that the intro "turtle" lectures were repetitive and could be shortened. The lectures were exceptional. None, in my opinion. Pseudocode and turtles None None None

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	importing urllib.request was explained weirdly - but i am not sure how else we would learn it. I actually thought all of the topics were pretty helpful in understanding CS as a whole. Using Turtle at the begining debugging (introduced too late) I did not like 'gamebox', it isn't applicable after the course ends; it felt useless and I thought that other topics could have been applied during that time. The topic that "did not work" was visualizing the variables as boxes and diagrams. If anything it just confused me more. None. None. They all worked End of the year game. The very early lectures on pseudo-coding I don't think the turtles were useful in the long run but it was fun so I'd keep it. I felt like gamebox wasn't nearly as important as the time speant using it was, like we speant so much time on it and it is a file created by our professor that we will likely not use again Perhaps turtle, but I still thought it was interesting especially for students who are interested in computer graphics like I am. n/a n/a n/a reg exp I felt like most of the topics were relevant. I cannot think of anything that particularly stands out for this question. Turtle Turtle I think the debugging strategies were dragged out a little. Spending 15-20 minutes of giving debugging strategies with a simple example would have been sufficient. nothing Some of the game related topics may not be as useful in the long run, though I think that some of the skills learned could maybe be applicable in some circumstances. It took me a long time to figure out how to use in concert reading from websites and regular expressions. If there was something else that could have helped us practice regular expression that would have been helpful. RegEx seemed shoehorned in and overall was not an interesting topic None really, each have their use in the future importing urls gamebox
4. 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: 109 Individual Answers: *See below for Individual Results*
	reading CSV

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	Loops and regex
	N/a
	loops and str methods
	The ability to read info from the internet and then use this data in another way.
	I believe my ability to effectively utilize for and while loops are the most useful feature
	basic code, using code to find things online, writing loops to perform tasks
	for loops and if statements
	reading CSV files
	The ability to search a webpage and parse it.
	Regular expressions and the fundamentals in terms of understanding how a computer processes information
	importing data from a url or file and looking at specific categories in that data set.
	accumulator functions
	I think debugging was one of the more useful things we've learned. I also think reading files and websites is really useful (and with that probably regular expressions, despite my not liking them)
	Everything in general that is about programming but also Regex.
	N/A
	File writing will be the most useful in the long run as i will be a mechanical engineer. Understanding basic programs and having a machine out put data into a readable format will be important.
	The fundamental topics, such as those dealing with loops, if statements, and lists, are incredibly useful as a basis for Python knowledge.
	Loops, as many tasks rely on them.
	Reading files.
	Lists, dictionaries, gamebox, functions
	I think opening and reading a file from the web was most useful. I could see myself having to obtain data from the web and it would be easier to write a program that reads it for me.
	regex, and how boolean values work, also dictionaries
	RegEx
	File writing and or streaming web files.
	Basic coding relating to the manipulation of strings and ints.
	writing and analyzing web pages
	Yes these concepts are very useful for any sort of work in the future.
	I think file writing/reading will come in handy the most.
	CSV stuff, reading from websites, gamebox (re-creating old arcade games for absolutely no reason has become a new hobby of mine)
	loops
	loops
	loops
	I think searching and reading websites was very useful.
	I loved making the lab where we found the closest arbys
	How to code games

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	The basics (variables, types, loops, etc.)
	Regular expressions
	Regular expressions
	Regular expressions and the logic of coding
	For and While loops
	Pretty much most of the stuff
	Just general understanding of how code works
	the basics of coding
	Loops were integral for nearly everything I did.
	Loops and exception handling.
	The programming assignements and regex helped me thing about logic which was fun and interesting and obviously applicable
	It will most likely be reading data or also maybe regular expressions.
	If statements, for and while loops, lists
	First 10 lectures.
	The topics that taught how to sort a large amount of numbers and manipulate them according to how one needs to (such as finding the median, mean, etc.).
	Regex
	Sorting through large amounts of data
	Regular expressions
	Basic list functions and string functions seemed to be the most important in class homework, so they may be quite important in future CS classes.
	I can't single out any one topic or lecture. The course as a whole, in my opinion, will be extremely useful to me in the years to come.
	loops, files
	The most useful lectures in my opinion were the lectures on collections and reading files.
	understanding and writing psuedo code to prep for hard and soft coding
	learning how to find information from URLs and working with data.
	Lists and other mutable characters.
	probably using regular expressions
	I really enjoyed the logic-based programs we wrote, however I think that the coding we did based off parsing websites will be most useful.
	python general knowledge
	Regexp
	Probably CSV reading and reading from websites. Maybe gamebox because I am interested in game design.
	Loops Booleans
	I dont plan on using code in my future. The problem solving skills will help in the future.
	The basics of python, loops, if/else statements, functions, etc
	for/while loops if statements
	Reading CSV files
	general ideas

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	The dict topic learning to read from websites, and making codes that take difficult tasks and make them simple processes regular expressions and loops Reading csv files/files from the web and using regular expressions I think just the ability to start thinking step by step that is required to complete programs will be the most helpful Using URL's and extracting information from web pages. Gamebox Debugging All of them I think functions were taught very well early on in the course. The adequate understanding of functions, invoking functions, and overall ideas of coding are key for everything else, and all these things were covered very well. It has helped me understand the ideas behind coding and the methods behind it. I think creating Lou's List and the GPA checker were all very useful. regular expressions and knowledge of coding in general Sorting through files and lists The beginning lectures on algorithms and avoiding ambiguity in coding were helpful for giving me the mindset I needed to go forward in the class. Writing functions, CSV/URL, regular expressions, gamebox list, dict, re I'd see gamebox and pygame being really useful to me in the future - I don't know that I intend to be doing much with websites or anything in the future but the opportunity to make games every once and a while with my own skills is really cool for me, and I could see myself doing it now and again. However, almost everything that we learned in the class had a pretty clear application, and I really liked being able to know the various ways that I could utilize my skills in the future. loops importing info from the internet (url lib.request and assignments that incorporated it) and pygame The nit-picky details about strings lists integers and floats at the beginning of the year were vital to understanding the course. Regular expressions or the structure of loops. Regular expressions, loops Learning how to access CSV files. While and for loops Just basic programming skills For loops and the multitude of ways you can store different variables like tuples or dicts to name a few. I will likely find the ability to use Python to read data files and search for data using regular expressions the most useful in the future. I'm not sure, probably just practice coding in general. I really liked the things pertaining to websites, reading them and organizing information from them. Regex and general loops. I believe that every lecture would be very useful in the future. I think knowing the foundation of if/else statements and loops (for and while) can be applied to other coding language. How to create games or other physical products or how to create regular expressions.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	I think the practice on urllib.request was very beneficial, since it's a little hard to practice that without a specific goal in mind. Loops and importing/using files/websites Regular Expressions and loops
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: 109 Individual Answers: *See below for Individual Results*
	game project! or lous list assignment Loops are the most useful, therefore my favorite topic. I liked loops and figuring out how to make loops because it enabled us to write more complex and interesting code and it was fun problem solving how to put things into what kind of loop. N/a game- useful and see the implications of it everyday ways of asking for input and generating responses Once I grew comfortable with it, I absolutely loved using gamebox to work on the final project. I put a great deal of effort into it, working with more enthusiasm than I did on any of the homework assignments. i wish the final project made up a larger portion of our grade. I liked reading data files and using them in code. I thought that was pretty cool. It's also relevant to my major. I also really liked the poi lab. For and while loops because the accumulator pattern interests me how such little code can do so much while looping PyGame! My favorite topic was creating games because I found it interesting. Pygame was my favorite, for it was a very realistic application of cs. I believe this encourages people to further pursue cs classes. My favorite lecture was the one when we went into the HTML and used regexs to find the information needed. Gamebox. I really enjoyed the work we did with it, including the final project. Reegular Expressions are interesting. loops, very key part of cs N/A The video game topic because it was a fun project to work on that combined a lot of the knowledge we received throughout the course. I though turtle was a great way to get momentum and inspire interest in the subject matter. The game was also of that nature where you were saying, here are some cool things you're capable of and here's a way to think about something productive and low stress and fun. Regular expressions, they just made sense and are fun to create Gamebox was by far my favorite. I spent quite a bit of time preparing the first part of the group game simply because I wanted to see what it would all look like. Also, I knew ahead of time that later aspects of the game would be easier to incorporate if I had the base functions already prepared. Designing the game and applying my knowledge was the best. I really enjoyed the topic about strings because the functions used to manipulate strings was very interesting. I especially liked the lab about cyber security. My favorite were the first few because it helped transition me from being scared to code because I had no clue what I was doing to actually thinking how programmers think. I liked making and using lists because it was fun to work through how to solve different problems by moving values

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	using loops The gamebox because it was enjoyable to learn how to make games. Talking about game creation was a lot of fun and I really enjoyed working with it; I felt like it really connected me to what coding could do in a way that was fun and really tangible. I liked the lecture where we just interviewed Luther about his personal life. I really enjoyed all of it, however I think that the game writing was my least favorite. I love every single topic because one cannot work without the other. for and while loops, interesting way of thinking about things I personally liked logic with if/else statements and using global variables. I like both these things because, when coding, you really need to know what the end goal is to succuessfully code. Gamebox, it made me feel like I was actually creating something useful. Gamebox was definitely the most fun and interesting, even though it was hard. I liked learning about gamebox because games are fun to make. gamebox; it was really interesting to understand more about how games work None The game unit because it gave us freedom to be creative and I like games. Pygame was my favorite topic because I got to create my own game from scratch. I loved anything to do with logic; it just felt like a puzzle. the if statement because they are so useful and the game project because it is just really fun to make a game. Pygame. I loved being able to implement my own creativity in a course, as I have not been able to do that much. Debugging. When working on code hundreds of lines long or implements the use of other files, I find it extremely useful to be able to comb through multiple error messages to be able to find quickly mistakes, even very small ones. n/a making games is a good way to involve cumulative information. I thought regular expressions was cool bc it was like how your computer actually works I liked playing with the turtles, but my favorite was probably regex gamebox creating games! I loved making the game gamebox cause its fun psudeo-coding My favorite topic was the one on if/elif/else statements because I found it very intuitive and easy. Pong My favorite topic was learning about loops as they are extremely versatile. The final game project because it was cool and fun Games - because anything can be a game and it's what allows the user to actual input stuff, so you could actually code things that will be useful for you I enjoyed URLs and files because it was easy to see how this would be useful moving forward. Pygame and gamebox were also interesting. Gamebox was my favorite, because I felt great and accomplished turning what I learned into something that was fun and I could use in the future

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	The game itself, a lot of fun to actually code something that does something and work through problems.
	i liked the analogies about files and delivery man because they allowed me to visualize the process of retrieving files
	the game. it was fun to do
	loops because they are applicable in other places
	I really liked having to design our own game. It was fun, and the lack of specific requirements and a rubric allowed us to explore what we could do, and know what we couldn't.
	REs because i understood them
	The lectures on gamebox were my favorite because we can use those skills to make a real game.
	LISTS. i just love lists.
	I really liked gamebox because I enjoy animation and drawing. It also gave us a way to apply our knowledge to a project and create something of our own.
	- Regular expressions, because was easy to understand and use.
	Pygame because making something you can play with afterwards is always more fun than not.
	I enjoyed learning how to use pygame and gamebox. It allowed me to be more creative with my code.
	Working with CSVs. It is empowering to have the knowledge needed to process lots of data.
	I enjoyed learning Pygame, I think I will continue to play with it after the class is over.
	I do not have a particular favorite topic, but I found most of them to be generally appealing to me.
	Games
	dict Fun
	regular expressions was really pleasing to learn because they were efficient to program, fun to figure out, and their utility seemed more obvious than some of the other things we learned
	The game project was my favorite because it was the most enjoyable topic, and I felt like I had the most freedom in creating something.
	Gamebox, I enjoyed making a game.
	I really enjoyed learning about regular expressions. They are like a puzzle you have to solve in order to find what you are specifically looking for.
	Print and return because they were simple
	RegEx, I could see very practical applications of this topic.
	if statements, they are just fun and simple to code, especially when paired with loops
	Regular expressions because it's kind of like a puzzle
	Working with gamebox was my favorite because it was fun to see the result of our programming in the form of a game.
	gamebox was honestly the most fun out of all the topics.
	Accumulator pattern, it just made sense
	The gamebox because it was fun
	I really liked dictionaries, probably because there's so much that can be done with them. I also really liked making the game for the final project.
	Regular Expressions: This made me greatly appreciate why google is the giant it is.
	Games, as I loved seeing my work slowly piece itself together from the bare bones of a game to a complete and working product.
	I liked turtle because it was a fun way to start the course and it got me very interested in the mechanisms behind what we were doing
	gmae project because it felt the most interactive and felt like i was actually building something

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

Gamebox; because it was simple to understand and its application was entertaining.

For loops, I like their repetitive nature and the ability to change things within it each run without heinously typing out hundreds of lines of code, because that sucks.

My favorite topic was loops for the above reason.

loops with user inputs because I found it useful when making my own programs--small programs, but useful and fun since I got to apply my knowledge.

I really enjoyed the lecture about dictionaries. I think they are really cool in general and it was well taught.

reading CSV, most relevant to what i hoped to learn in this class

reading files because it is outrageously useful

try and except, because I was bewildered by the unpracticality of the program crashing if there was a single error

Learning how to use URLs and CSV files to retrieve information from online.

I loved learning game design.

I liked the assignment with finding the closest Arby's because it was really cool and involved the real world, but I do think they could've given us more general help in the beginning.

The early stuff cuz it was easy

I really enjoyed learning about lists and other collections

Pygame was so fun and a joy to learn.

gamebox because I genuinely enjoyed making a game.

dictionaries, the theory is really cool

6. 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)
117	4.66	0.53	80 (68.38%)	34 (29.06%)	3 (2.56%)	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)
229	4.59	0.63	146 (63.76%)	75 (32.75%)	5 (2.18%)	2 (0.87%)	1 (0.44%)

7. 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)
119	4.52	0.71	74 (62.18%)	36 (30.25%)	6 (5.04%)	3 (2.52%)	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)
231	4.41	0.86	135 (58.44%)	70 (30.30%)	15 (6.49%)	8 (3.46%)	3 (1.30%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

8. 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)
119	3.39	1.32	32 (26.89%)	27 (22.69%)	28 (23.53%)	20 (16.81%)	12 (10.08%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
231	3.25	1.36	52 (22.51%)	58 (25.11%)	51 (22.08%)	36 (15.58%)	34 (14.72%)

9. 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)
118	4.41	0.60	55 (46.61%)	56 (47.46%)	7 (5.93%)	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)
1697	4.05	0.86	511 (30.11%)	881 (51.92%)	201 (11.84%)	74 (4.36%)	26 (1.53%)	4 (0.24%)

10. 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)
119	4.20	0.87	51 (42.86%)	48 (40.34%)	14 (11.76%)	5 (4.20%)	1 (0.84%)	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)	Not Applicable (NA)
2781	3.92	1.02	825 (29.67%)	1071 (38.51%)	399 (14.35%)	215 (7.73%)	71 (2.55%)	200 (7.19%)

11. 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)
119	4.22	0.91	53 (44.54%)	48 (40.34%)	12 (10.08%)	3 (2.52%)	3 (2.52%)	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)
1699	4.06	0.89	549 (32.31%)	859 (50.56%)	167 (9.83%)	88 (5.18%)	32 (1.88%)	4 (0.24%)

12. 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)
119	4.40	0.84	68 (57.14%)	37 (31.09%)	9 (7.56%)	4 (3.36%)	1 (0.84%)	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)	Not Applicable (NA)
1701	3.91	1.03	535 (31.45%)	668 (39.27%)	273 (16.05%)	135 (7.94%)	47 (2.76%)	43 (2.53%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

13. 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)
119	3.39	1.10	21 (17.65%)	24 (20.17%)	41 (34.45%)	15 (12.61%)	5 (4.20%)	13 (10.92%)

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)
1699	3.33	1.11	163 (9.59%)	272 (16.01%)	357 (21.01%)	128 (7.53%)	71 (4.18%)	708 (41.67%)

14. 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)
118	4.40	0.69	57 (48.31%)	54 (45.76%)	5 (4.24%)	1 (0.85%)	1 (0.85%)	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)
2774	3.96	0.97	780 (28.12%)	1217 (43.87%)	345 (12.44%)	159 (5.73%)	77 (2.78%)	196 (7.07%)

15. 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)
119	4.83	0.38	99 (83.19%)	20 (16.81%)	0 (0.00%)	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)
2781	4.38	0.73	1274 (45.81%)	1095 (39.37%)	176 (6.33%)	33 (1.19%)	17 (0.61%)	186 (6.69%)

16. 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)
119	4.74	0.46	89 (74.79%)	29 (24.37%)	1 (0.84%)	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)
2779	4.31	0.77	1178 (42.39%)	1124 (40.45%)	207 (7.45%)	41 (1.48%)	26 (0.94%)	203 (7.30%)

17. 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)
119	3.34	1.14	13 (10.92%)	14 (11.76%)	22 (18.49%)	12 (10.08%)	3 (2.52%)	55 (46.22%)

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)
1694	3.76	1.04	291 (17.18%)	417 (24.62%)	267 (15.76%)	83 (4.90%)	39 (2.30%)	597 (35.24%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

18. 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)
119	4.24	0.88	55 (46.22%)	46 (38.66%)	11 (9.24%)	6 (5.04%)	1 (0.84%)	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)	Not Applicable (NA)
2773	4.10	0.88	838 (30.22%)	1141 (41.15%)	270 (9.74%)	101 (3.64%)	40 (1.44%)	383 (13.81%)

19. 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)
117	4.51	0.71	70 (59.83%)	41 (35.04%)	3 (2.56%)	2 (1.71%)	1 (0.85%)	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)
2775	4.27	0.76	1062 (38.27%)	1193 (42.99%)	230 (8.29%)	41 (1.48%)	22 (0.79%)	227 (8.18%)

20. 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)
119	4.69	0.50	84 (70.59%)	33 (27.73%)	2 (1.68%)	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)
2771	4.07	0.89	886 (31.97%)	1134 (40.92%)	375 (13.53%)	117 (4.22%)	31 (1.12%)	228 (8.23%)

21. 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)
119	1 (0.84%)	36 (30.25%)	57 (47.90%)	17 (14.29%)	8 (6.72%)

Results for SEAS, 1000-level courses

Total	Less than 1 (NA)	1 - 3 (NA)	4 - 6 (NA)	7 - 9 (NA)	10 or more (NA)
1703	268 (15.74%)	827 (48.56%)	464 (27.25%)	106 (6.22%)	38 (2.23%)

22. 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)
119	4.60	0.54	74 (62.18%)	42 (35.29%)	3 (2.52%)	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)
1700	3.96	1.01	570 (33.53%)	705 (41.47%)	258 (15.18%)	121 (7.12%)	46 (2.71%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

23. 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)
118	4.40	0.83	67 (56.78%)	37 (31.36%)	8 (6.78%)	6 (5.08%)	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)
1697	3.83	1.14	567 (33.41%)	597 (35.18%)	282 (16.62%)	174 (10.25%)	77 (4.54%)

24. 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)
118	4.59	0.56	74 (62.71%)	40 (33.90%)	4 (3.39%)	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)
2772	4.13	0.85	991 (35.75%)	1313 (47.37%)	362 (13.06%)	50 (1.80%)	56 (2.02%)

25. 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)
119	4.24	0.80	54 (45.38%)	42 (35.29%)	21 (17.65%)	2 (1.68%)	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)
2771	4.06	0.88	938 (33.85%)	1227 (44.28%)	471 (17.00%)	93 (3.36%)	42 (1.52%)

26. 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)
119	4.50	0.66	68 (57.14%)	44 (36.97%)	5 (4.20%)	2 (1.68%)	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)
2779	4.03	0.95	959 (34.51%)	1190 (42.82%)	438 (15.76%)	126 (4.53%)	66 (2.37%)

27. 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
64	*See below for Individual Results*

Good course

The TAs are not always helpful or willing to meet the student halfway on assignments. As a result, during the second half of the semester, I hired a tutor who has a bachelors degree in computer science and even he was struggling with understanding some of the assignments. It's a 3 credit introductory course. I'm in danger of failing the class which could result in academic probation.

This was hard because i did not expect to the amount of work needed

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	N/a
	This was my first CS course, and it was difficult but enjoyable. The level of difficulty was a good challenge for my skill level.
	I only had brief experience with coding before this class. However, Prof. Tych and all the TAs have been wonderful, to the point where I am now considering a minor in CS. By far my favorite class this semester.
	Really awesome class with a great professor and great TAS
	I think this was a good and very challenging course. I have one critique though. I think that a main reason that people have trouble following Prof. T in lectures is because he is an incredibly fluid and fast typer in Python. Moreover, he would type the expressions and code so much faster than everyone else. I think that you have to realize that we are not nearly as fast as you on the computer. Please slow down for the students in the future, it makes it very difficult to follow along if you can type a line of code in 30 seconds that takes me 2 minutes.
	Great course!
	Prof Tychonievich is a great professor and it's very evident on how knowledgeable he is with the material. I wish there a little more structure to the lectures because sometimes it could be easy to zone out and when you zone back in it would be hard to figure out what was happening since it was always examples of code.
	Thank you for a very interesting and informative CS introduction!
	..
	This course was AWESOME. Professor Tyconovich is great at explaining things and also really, really engaging, and I always enjoyed learning from him. He does a great job of teaching coding and Python. The TAs were all excellent and I had a lot of fun in this course :D
	I have no additional comments.
	Professor Tychonievich is an excellent instructor. However, I feel as if the course started too slow at first and then quickly sped up at the end, especially with dicts, games and regular expressions.
	A difficult course at times but a good professor helped.
	I think this course is a worthwhile course and was enjoyable. My only complaint was that it seemed like the exams were designed to trick you and test the one case you never considered to check. The TA's were very helpful and there was plenty of opportunities to get help on assignments.
	I really enjoyed this course! Before, I was considering the idea of minoring in Computer Science, but now I'm actually going to try and minor.
	I really enjoyed this course, and it is because of Professor Tychonievich. He made the class entertaining and engaging, and his methods for teaching (his explanations through drawings and analogies) made it easier to understand.
	I liked learning how to do basic coding, however, some of the homework assignments seemed to require skills well beyond what we used in class.
	This was the worst class I've ever taken solely because I've learned I am really detail oriented enough to be natural at programming. However I have no issues with how the course is structured and run and think the Professor and TA's do a really good job. Though I'd never take a CS course again, I found this helpful in the long run.
	A phenomenal teacher. The class was very interesting and definitely would recommend to others.
	Tychoneivich often inadvertently made students feel dumb for asking questions he didn't understand, in office hours he did not seem happy to answer questions regarding test grading
	Please get rid of gamebox; it does nothing and only confuses students!
	For the most part, thoroughly enjoyed this introductory course.
	Please make the exams more similar to the practice exams
	I felt this class was very beneficial to my understanding of computer science and now I will be able to use the skills I've learned to code programs in the science world.
	Interesting course, I like the flexibility of lectures
	His lectures and teaching were very clear! Computer science is not a walk in the park, but he tried to explain it as best he could. He posts all his lectures online, so you could easily go back and rewatch a topic you don't understand. That being said, the exams were very long and difficult for the amount of time given (50 minutes). While do-able, they were still difficult and it was even harder under the time constraint.
	Great teacher.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	Tychonievich is an amazing professor... give him a raise!
	Prof. Tychonievich's passion for the subject material is conveyed through his way of teaching. This course was extremely well organized (I am especially fond of the Archimedes assignment submission page), lectures were recored, making it very easy to catch up and review. There was a nice balance of work and fun. Overall, extremely worthwhile.
	was a good course, just done care for CS
	Good one
	This class should be advertised as a class only for students with prior CS knowledge; or if you have unreasonable amounts of time to dedicate to it.
	great course great professor. Hopefully exams are easier since I don't think they reflect Computer Science skill as much as test-taking skills
	The study resources should be clearly advertised to students. I would have done significantly better on the first exam if i was aware of which study websites to use and how they worked.
	Loved it
	thanks for putting the lectures online
	I thoroughly enjoyed this course and regret not taking it sooner in my college career - I probably would have majored or minored in CS, but no longer have the time to do so (rising 3rd year, already declared 2 majors).
	My only complaint really was the pace of the course. At first it started very slow and then out of nowhere after the drop deadline it significantly picked up. Some weeks id have 3 difficult assignments due and I'd spend so much time in office hours. Other than that I feel like I learned so much and Tychonievich was an outstanding professor. He is so knowledgeable in the subject matter and is very good at teaching CS.
	The professionalism points are kind of dumb. Asking for a regrade that, in the TAs opinion, is "frivolous" does not warrant losing points on your total grade. Especially since TAs are students too. They should want to give us points based on our understanding of the material, not give you sass when you ask for your code to be looked at again. TAs literally signed up for this and are being paid to grade and look at code. Sometimes you have to deal with looking at things again, no matter how many idiotic or seemingly frivolous requests there are. Although, in life, we canât filter these out. It seems like TAs have been trained to try to do that. Anyone who says they wouldnât try to get as many points as possible (within good reason of course), including the TAs, is lying. I could see their frustration if they werenât getting paid, but that is not the case. TAs are literally there to help you for less than a minute and then race off during office hours. Some TAs in labs didn't know what they were doing and looked at the rubric code when explaining things (especially regular expressions) and labs, for most, were never actually completed. My problems this year were, clearly, with the TAs not Professor Tychonievich, who was a great and enthusiastic professor. Also, this is coming from someone who lost a negligible amount of points to my grade due to these professionalism points but was annoyed with the intimidations made by the TAs to dock points.
	Great course. I learned a lot about python and about the logic behind programming in general.
	Very good class overall, has convinced me to take the next level of CS
	I thought Tychonievich was a great professor. It was clear that he cared about the subject matter and had a through understanding of what he was talking about but was also expert at conveying these ideas to people who really have no idea how to do anything computer related (me). He also had an impressive amount of patience when dealing with some really idiotic questions.
	None
	Tychonievich is a wonderful professor, great class
	I think I accidentally put my review of the lecture in the course evaluations for the lab so my review of the lab was that the TAs were good but could've talked to us more at the beginning about how to approach the problems.
	I took this class because I noticed that many jobs/internships list Python as one of the skills they look for. It's also required for my major. This class gave me a good introduction to Python. The TAs were great. They explained concepts well.
	I really enjoyed this course, my first CS course.
	Tychonievich was amazingly prepared for each class and is extremely good at answering questions.
	The exam difficulty definitely increased from past semesters judging from the practice exams provided.
	n/a
	n/a

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	Great course. Taking another CS next semester. It was fun and taught me huge amounts. As I stated in the other evaluation, this course was definitely helpful and I really enjoyed it. The practice tests were way easier than the exams-- students should be provided with practice tests that are of the same caliber as the exams. It is not fair to have students take an exam that is way harder than the previous assignments/ practice material given to them. Exams should also be in python--they should allow students to write a program in python, because this resembles cs in the real world-- in the real world you do not code on paper. Someone should figure out a way to monitor if people are opening up other tabs and such -- that way coding can be done in python or in an online version of it-- i know this is possible. This would allow students to change their code based off of the feedback python gives them(error messages, etc produced after running it). This is a way to make this class a more accurate real world model. Taking this class was an absolute joy. I personally never showed up to lecture, but the videos were great! Tychonievich is refreshingly enthusiastic about his job and it carries over to the students. The labs and assignments were fun and challenging for the most part except for maybe regex and some others. luther rox This definitely was an important, albeit demanding, course for me. I know its an important subject. I do not plan to take any further CS classes because it is just not my cup of tea, but have complete and total respect for CS majors. The lecture would've been more effective if he responded directly to questions even though we did not particularly know what we were talking about since this was an entry-level class. The review sessions were not that helpful because he does not base current tests on previous tests. I really enjoyed Introduction to Programming. The content was well-organized and clear and challenged us just enough. It helped solidify my basic knowledge of computer science and made me declare it as my major. Loved this course, even though it was challenging was really rewarding when you got it down Amazing course! After taking this class I decided to major in Computer Science.