CS 1110-002 **Introduction to Programming** - Spring 2014

ENGR (17342)

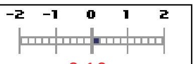
INSTRUCTORS: **Tychonievich, Luther (lat7h)**

Respondents: 189 / Enrollment: 313

Summary: CS 1110-002 Introduction to Programming - Spring 2014 (17342)

Overall Course Rating

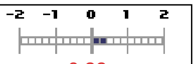
CS-1110-002 Mean 3.96
CS-1110-002 Std Dev 1.00
CS-1110-002 Response Count 927

Difference from Category Mean, Expressed in Category Standard Deviations	 0.16

SEAS, 1000-level courses Mean 3.79
SEAS, 1000-level courses Std Dev 1.12
SEAS, 1000-level courses Response Count 6388

Overall Instructor Rating

INSTRUCTOR: Tychonievich, Luther
Mean 4.24
Std Dev 0.82
Response Count 1302

Difference from Category Mean, Expressed in Category Standard Deviations	 0.32

SEAS, 1000-level courses Mean 3.89
SEAS, 1000-level courses Std Dev 1.10
SEAS, 1000-level courses Response Count 12130

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

1. 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-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
189	4.25	0.88	84 (44.44%)	83 (43.92%)	10 (5.29%)	9 (4.76%)	3 (1.59%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
290	4.34	0.84	147 (50.69%)	114 (39.31%)	15 (5.17%)	10 (3.45%)	4 (1.38%)

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-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
188	4.32	0.75	85 (45.21%)	84 (44.68%)	14 (7.45%)	4 (2.13%)	1 (0.53%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
288	4.41	0.72	148 (51.39%)	118 (40.97%)	16 (5.56%)	4 (1.39%)	2 (0.69%)

3. 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-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
189	3.04	1.40	39 (20.63%)	37 (19.58%)	40 (21.16%)	38 (20.11%)	35 (18.52%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
290	3.17	1.41	70 (24.14%)	58 (20.00%)	63 (21.72%)	50 (17.24%)	49 (16.90%)

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

4. How accurate is this statement for you: Pair Programming helped me learn the material better.

~

Question Type: Likert

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
189	3.43	1.18	33 (17.46%)	74 (39.15%)	40 (21.16%)	26 (13.76%)	16 (8.47%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
290	3.45	1.18	54 (18.62%)	109 (37.59%)	64 (22.07%)	39 (13.45%)	24 (8.28%)

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

~

Question Type: Short Answer

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-002, Tychonievich, Luther

Total	Individual Answers
166	*See below for Individual Results*

Making the Joust game because I was proud of the result.

Methods, because that is when it all made sense to me.

Writing methods was my favorite topic because they are so useful.

The turtle because it was the easiest.

Creating classes and methods that are not built in to Java. It is cool to see how programmers can turn code that looks like gibberish to something that you can be proud of, like for example the Joust Screen

Classes and Methods. Most useful in the future

favorite topic was loops. I thought they were the most interesting form of logic

I really enjoyed learning about loops and recursion.

Proffessor Sherrif's visiting lecture because he explained the material so well

loops were interesting to learn and allowed us to do so much more.

Learning to create and implement classes. This seems widely applicable in CS.

The beginning. It was easy

I really enjoyed making the bird video game. It was a great way to bring all the topics of the class together in a fun project.

My favorite topic was getting data from URLs/Excel spreadsheets because it is extremely useful.

I really liked the scavenger hunt activity with the caesar cipher. It was creative and fun.

I liked loops because they allowed you to do more complicated things and come up with results that a person couldn't do efficiently by hand.

if statements and for and while loops because I understood them and he spent a lot of time going over them step by step, making sure everyone was keeping up.

Making the joust game

learning about for loops because they are widely applicable in an array of situations

I loved the recursion topic the most because it was difficult to wrap my brain around at first so it was challenging

designing the game for our last homework

Turtle cause it allowed the creative freedom of what to draw

I liked learning about methods once I understood how they worked which took me a week or so. Once I realized how useful they could be and how to call them I enjoyed working with them.

methods and classes. it demonstrated the variability and the creativity that programmers need to have in order to write efficient programs that do what they need to do

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	For loops
	I liked the intro material because it was easier to understand.
	N/A
	The birds project was my favorite because it showed how you can actually do something real with the material we learned.
	Loops, it made me think differently
	recursion, looks the coolest
	I was not fond of any topics, but Recursions was the least bad for me because it was the easiest to understand.
	examples
	I enjoyed the Caesar cipher scavenger hunt the most, since it combined coding ability with an interactive activity.
	The method portion of the class was my favortie becasue that is when we were actually able to make stuff happen in lab and homework.
	I liked some of the earlier stuff we did, like the calendar program.
	I enjoyed writing methods because it consisted of how to write tasks for computers
	i liked learning methods because it allowed the code to be so much less repetitive and messy
	Arrays; most useful
	video game-interesting
	Basic turtle drawing.
	I enjoyed the Kitten lecture quite a bit. Overall, the lectures were entertaining and always provided useful information, so no one lecture stands out in my mind.
	The lecture on methods was my favorite because we can create our own programs and make it do what we want it to do.
	ciphers
	TURTLE DRAW CUZ TURTLES ARE TIGHT
	classes and methods
	loops! Once I got it, it made sense!
	the first one.
	loops
	loops
	loops
	Loops because they did cool stuff but weren't too hard to understand.
	Turtles was my favorite because we could be very creative with our pictures!
	Turtle drawing
	Fractals
	I enjoyed learning about methods and classes. Programming was interesting before, but seeing that it is possible to create basically anything that you want or need in a program was fascinating. Although it was confusing and there was a steeper learning curve, I like the way writing methods makes me have to really think.
	I loved learning about methods because it was especially interesting for some reason
	Loops, they were easy to understand.
	turtle, pictures
	methods because they allowed the creation of complex programs

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*

recursion when we actively participated (i.e., stood up and verbally demonstrated what was going on in a certain recursive program). It was fun and informative.

Classes - While not overly interesting classes are a key part of any coding and allow people to create far easier to read and to use programs.

loops because they can complete so many tasks

Loops, because I liked being able to run iterations of code with only having to write a few lines.

My favorite topic was the turtle pictures because it was simple and was a good starting off point of coding.

Early on, turtles was a real cool and got me interested in the class. We brought them back later in the course and I felt like they made more sense.

Loops

Loops

Using turtle was very fun and I also enjoyed learning about recursion because it was interesting to incoporate.

Learning to write methods and classes

The simple methods in the beginning because it was easier to grasp.

The drawing of shapes using mathematical knowledge or recursion because I like to draw and enjoy using math in other subjects.

I like turtles.

Recursion, because it looked cool

Manipulating arrays and arraylists using loops was my favorite because I understood it instantly.

Recursion was pretty simple for me to grasp

Writing methods because it is very adaptable to the programmer's needs.

Computer Game design!

Can't say I have a favorite, I really enjoy CS. In general, I most enjoyed the more challenging homework problems that made use of applied loops and reading CLV files.

practicing writing code because that's where i got to practice what i'd learned and that's when it really gets engrained

I enjoyed most of the topics, especially the early ones.

The assignment involving making a game, it allowed a lot of chances to figure out how I wanted to do things.

The game topic, it was the most interesting.

The lessons in the first two months because those were easiest to understand.

I liked making the computer game; it was a lot of fun and tied together many concepts well.

Recursion, making fractals was something that interested me before this course.

Methods, was actually useful, not just doing stuff that other people have done a thousand times.

classes and methods, made programming a lot more interesting

I really enjoyed learning about recursion. The demonstrations during lecture were helpful and enjoyable and the things that the code written could do was extremely informative and entertaining.

Recursion

The initial topics on logic structures.

Topics covered in the first four homework assignments.

I liked learning the basics because I needed to learn general coding to build upon that.

All.

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	I very much enjoyed all the class writing we got to do, because there is an immense amount of satisfaction that comes from starting with nothing and building a program from the ground up.
	I enjoyed learning about loops and arrays the most. After finishing the homework that scanned various webpages and gave back information based on the user input, I felt like I really understood the material.
	I did not have one.
	moving turtle. simple and easy.
	I like the turtle lectures because I was fascinated that I could program a computer to draw pictures using a turtle.
	While loops because it was taught the best.
	I liked the loops because they made executing actions much less work.
	Reading files, made the most sense to me.
	loops...it was fun to implement and very useful
	Turtle, it was fun
	The first couple lectures about Turtle, becuase turtles are always fun.
	Creating the game for homework 6
	recursion
	recursion
	recursion
	Classes show the power of cs
	HW06 was the most interesting assignment because I gained insight on how video games are made.
	Turtles because it was fun
	Loops and decision structures because they were easy to understand.
	Recursion: It was the most challenging to understand but the easiest to code
	Recursion, it made sense to me although it was very difficult to put into code.
	I enjoyed reading Array Lists. I felt like I grasped that concept better than the others.
	I enjoyed programming the game for homework 6
	Turtles
	I enjoyed the creating new classes section most because I understood more of how Java works instead of just calling classes that Java already had and just expecting them to work.
	I really enjoyed recursion because it was something I had not learned before and I thought the applications of it were very interesting.
	My favorite part was learning how to write and use methods as that was the point I felt like we really learned how to manipulate code and make it work for us. Neat stuff.
	I liked when we coded with just one main method because I understood that a lot better.
	Bird because it was the most interesting and beneficial.
	None of them. I was miserable throughout
	Loops because they were fun
	Turtle demos
	I enjoyed for loops because I could very clearly see the practical application.
	I enjoyed writing and being able to personalize code for a game (HW 6)
	I liked learning about loops. It made me realize how a lot of programs work in a more simple manner than many may have expected.
	last homework

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	I enjoyed using Turtle. turtle Classes/Methods because they allowed us to apply a more practical knowledge to our skills. Turtle work My favorite topic was the arrays and array lists. I'm not really sure why, I just liked how they functioned. The first ones that weren't hard because I actually understood them Arrays and array lists because it was interesting to see how the information was sorted and how it could be utilized to do things such as sorting classes for scheduling. I can't point to any particular topics/lectures, but I found the ones on loops, recursion, and scanners to be some of the most interesting and useful. The beginning parts because that's when I actually understood the stuff. My favorite lecture was the first lecture where we drew pictures with turtles because I honestly enjoy drawing. Loops, because it saves a lot of time programming with loops. Homework 5 and CSV readers because I felt that I was comfortable with the many functions of Java and could see the real world applications of this topic. I also liked the lecture where we made paper airplanes because it was fun. methods because learning these allowed us to really begin to understand basic code My favorite topic was on making methods because it was easy to understand and made coding much easier. the game bc it was very useful Learning about methods was pretty cool, because it increased the range of commands we could tell the computer to do. making classes because that I feel that it is the most commonly used skill in programming. Turtle Drawing I enjoyed learning about recursion and the power of it. It is pretty cool how it does everything on its own and can be incredibly powerful. I liked loops, they were complicated enough for me, but not too difficult where they just confused me. I did not have a favorite specific topic or lecture Methods were my favorite items in the course because they made the coding much easier. Loops and recursion, because it was interesting to learn ways to make the program repeat for a certain amount of times. For and While loops were very useful, and can be applied many ways in writing code I liked the last homework problem the best because it incorporated everything. Recursion and loops. It's interesting how you can condense code like that. Turtle because it was fun to play with Recursion - definitely the most challenging concept My favorite topic was working with for loops and if statements. For loops because there were many different components of them that needed to be figured out and it was easier to understand what was going on The coding of the game, because it seemed fun and useful in real life. Algorithm. It is intriguing. The easy stuff at the beginning was fun. I really liked learning how to write classes. It opened up a whole new world of possibilities that I hope to explore!

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*

I enjoyed loops the most because I understood them and they made sense; they are also very applicable.

Loops - seemed like a difficult topic at the beginning, but after a while were an essential part of programming.

Map activity, cipher

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-1110-002, Tychonievich, Luther

Total	Individual Answers
159	*See below for Individual Results*

I'm not sure that I'll use any of this in the future.

basic programming

method/loops

Understanding classes, because it seems like a fundamental topic in order to have multiple systems work with each other.

Most useful would be creating different classes and methods.

Methods/Classes

Arrays

The most useful lecture was probably the recursion lectures.

N/A because I don't plan on continuing in CS.

I think that understanding the basic way Java works will be the most useful in the future, as it will allow students to have a better understanding of other languages.

The lectures improving our understanding of if statements and different types of loops will prove to be most valuable, in my opinion.

Any class that prepared me for future CS classes.

I think using recursion methods will be the most helpful in the future.

writing methods

The overall process of troubleshooting I think will help me the most in the future because it's nonspecific to Java- I am less afraid of using my computer overall.

the basics of declarations and printing out things

concept of recursion

Writing code

Learning Java in general

Nothing.

the introductory knowledge to programming

N/A

N/A

All of them.

The lecture on classes is probably the most important. I didn't pay attention and had a lot to catch up on.

I think any code that you write that organizes information systematically will be the most useful for jobs in the future.

See above.

actually working with java. besides that i don't actually know how much CS I'll need in the future

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	learning how to program a basic bird computer game
	Logic (true, false, etc) and loops (for, if, else, etc.)
	Probably just thinking out problems in a logical and unambiguous manner.
	Class and method writing are the most useful topics because they make code a lot shorter and faster.
	I also think loops and decision structures will be the most useful
	The basics of coding logic
	methods and games
	Methods
	Methods
	I think coding as a whole will be helpful in the future.
	if statements, loops, recursion
	Loops and recursion.
	I think all of the basic skills such as writing loops and calling methods and all of that will be helpful in the future if I need basic Java skills for my job.
	loops
	loops
	I think just the basic stuff learned in the beginning will be useful.
	The basic more versatile classes we learned to make such as linear and binary search.
	I would say the class in general, to give a basic knowledge of programming, is the most useful
	basic coding
	The paper airplanes lecture on ambiguity.
	All of them!
	The lecture on methods and class.
	None
	None
	The first few lectures about just understanding how code works will be helpful in the future.
	Basic programming
	Prolly how to use classes properly.
	classes
	loops and if statements
	just the general better understanding I now have of computers
	Problem solving in general would help me as I am not planning on continuing with CS
	unknown
	Definitely the if statements and for loops
	Also methods.
	Everything. Code writing!
	Loops
	Loops
	Loops
	The thought process / logic skills developed

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	Probably writing classes and recursion
	Probably just the basic programming of a method.
	Probably method and class devlopment
	Loops and if statements.
	I think the logic of if statements and loops will be most useful for me since they are pretty basic and I don't plan to go too far into CS.
	cipher
	the ability to write basic code/understand the language; because i'd like to make or edit some computers to do DIY projects
	Learning about classes and methods is the most useful, because those are concepts that are harder to learn on your own. All the little details I can easily look up online.
	methods
	methods
	methods
	methods
	All the basic coding skills I learned will be useful in future classes and maybe even after college
	the general lecture about what makes a good code
	I think just the process of learning a oomputer language will be the most useful thing moving on, since it will translate to so many other skills.
	Everything
	They all are because you need every piece to do the work required.
	I found using arrays and array lists were useful.
	Using methods will probably be the most useful in the future.
	Binary, linear searches and recursion
	Writing methods
	the fundamental principles behind programming
	General knowledge to use Eclipse
	I believe that all the lectures built upon each other and each was important to my understanding of cs.
	All of it; this class made me want to be a CS major.
	All.
	The topic involving loops and if statements
	none
	none
	Probably the 'nuts and bolts' of coding lectures, where we actually learned about loops and objects etc.
	recursion
	recursion
	recursion
	Writing Methods.
	I believe the lectures on methods will be the most useful in the future.
	Objects and methods
	Java introduction...moving on to HTML and javascript for web design

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	Reading code will help me the most in the future.
	I think writing classes and methods will be the most useful.
	In general knowing how Java works and how to program.
	making more than one methods in one class.
	Probably the ability to trace code. I feel as if that's really important for engineers to be able to do when they are designing processes.
	Most of the material covered later in the course.
	File reading
	Recursion?
	I thought that the discussion of methods was most useful for the future, since it made coding much, much easier, as well as broadening the capabilities of programming.
	I think the use of loops, if-then statements, or class creation may be the most useful topics. I think recursion is also very important.
	Turtle demos
	Classes - see above
	Very little of it
	Overall, everything in this course is cumulative. Any topic learned at the beginning of course was still reinforced by the end and as such, I think it's all useful.
	Writing classes
	I think the UML diagrams will be the most useful. They help organize information to make a well run program.
	Being able to use "if" "while" and "for" statements/loops are somethings that seem to appear quite a bit and will probably be useful.
	Basics
	Creation and implementation of classes.
	building the game for homework
	All of them
	Methods again
	Simple loops used for future hw assignments.
	Generally how to communicate with computers (early on in the class).
	methods and classes.
	all of it
	Arrays and array lists
	I think that the lectures from the early part of the semester will be the most useful in the future because they taught me how to be more deliberate, clear, and concise when dealing with computers.
	working with arrays and array lists
	I really did not like this class and hope to avoid computer science related material in my future.
	I think the Birds program will be very useful. Although it was a very simple game, I realized the mechanics of it were quite difficult. I think it will help me with CS 2110 if I choose to take it next semester.
	if statements
	Learning the basics of Java.
	classes/methods

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

Honestly, the whole class is really useful. With the basics now understood, I feel as though I could teach myself applications beyond the class to where I would be able to complete simple tasks using JAVA in the real world.

All of it

Ripping the phonebook apart was memorable and will help me remember the different kind of searches

The most useful topic learned will definitely be when we learned about looping.

The more basic topics will probably be most useful since I'm much more likely to use these simple aspects than the more advanced things

the lecture about ArrayList

I think loops are one of the most useful/important topics that we did. Overall, I found that the homework was the best way for me to practice my skills and find my weak/strong suits.

Getting info from URLs or Excel docs.

"Classes" lecture

None. I thoroughly do not enjoy CS and do not plan on using it.

The URL scanning, splitting lines, and finding certain pieces of texts seems the most useful for my future.

All of it... Becoming a CS major

I think the overall idea of the logic behind coding will be most useful.

Data analysis

None. I will not every use these skills in my life.

In general, just all of it.

Loop code and the ability of the computer to perform repeated processes with precision and efficiency.

Honestly probably none since I don't see programming as a part of my future.

Learning to create methods

Methods.

Recursion will probably be the most useful to me in the future.

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-1110-002, Tychonievich, Luther	
Total	Individual Answers
152	*See below for Individual Results*

The turtle lectures at the beginning of the year I do not really see why we did those.

Creating the Joust Bird Game.

I started getting confused once we got into objects. After that things were a little foggy. Recursion was the hardest concept I think.

no

no

csvreader

All topics were very useful.

The lecture on what makes "good code"

The background information in the first few classes about computers and the basics aren't something that I see myself remembering or being very useful in the future.

when things were drawn was confusing.

None really. Maybe making video games, except creating Joust Screen definitely gives you a better understanding of hold to write your own classes and methods.

comparing different coding languages

Recursion is still confusing to me

The programs we wrote before loops could have been written much easier with the knowledge of loops.

I cannot think of any.

There were some lectures/topics that I had a lot of trouble grasping and that I still struggle with, but I do not think any was not useful in the long run. The topics build on each other or create a foundation for future courses.

peer programming. I'm not sure why the huge emphasis on group programming: it just proved difficult to coordinate and slowed down our progress. Also, I'm not going to major or minor in CS or engineering or anything like that. I took this class to learn basic programming skills, so i guarantee i'll never peer program again

Making code better

some stuff was rushed/didn't capture theme well in class

Can't think of any.

Nothing.

I think the recursion demo was pretty rough... but mostly because half the class wasn't paying attention or didn't care enough to try.

N/A

N/A

N/A

none.

the numbers patterns

recursion. I was lost basically during all the lectures

I mean I don't really see myself pursuing anything with CS so technically all of them but I think all the topics were very helpful and interesting to learn over the course of the semester.

I had a very hard time understanding recursion and don't see it as being useful in the long run.

we did some stuff that was not necessary in the coding we had to do. we should do less fancy stuff and more basics

The encrypted stuff was cool, but seemed kind of useless.

Turtle was not useful to me. I do not understand the graphics part of this course as well.

The last few did not really seem to apply to what was happening in lab/hw.

MANY of the lectures were completely pointless. i often sat in class wishing i could be back in my room simply reading the textbook.

?

none, they all seem useful

All were good.

I was not able to do well in this course overall, so I do not see a use for any topics.

Turtle earlier in the semester was too much too quickly and almost encouraged me to drop the class.

I feel like most of the classes after working on "bird" project were useless other than the class spent talking about recursion. I feel like I didn't really learn a lot from all of the conceptual based classes. Also, I never got a firm grip on static vs. nonstatic.

Array Lists were annoying, but I did see the value in the end.

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	I didn't like how we did not
	The more conceptual classes weren't introduced well. The Movies/Theaters classes were gone through too rapidly and quickly and should've been introduced earlier and broken down further.
	They were all equally valuable
	field tables
	I feel as if the types of searching were pretty useless and semi-common sense.
	The whole course is not very useful for someone who is not planning on being an engineer but in my case I really was interested
	forcing us to be partners with someone in our lab
	topics that we did not to use in homework at all.
	None
	None
	None
	None
	the Turtle/World segment at the beginning of the year was more confusing than helpful in my opinion. It seemed more as a way of using every topic we were going to learn without teaching them. Every topic after was helpful though.
	Recursion was the hardest because i just wanted to do everything with loops.
	The complex topics... basically everything after spring break. Methods and objects were very difficult. Recursion is too confusing
	Most lectures did not help me learn the material at all.
	None.
	None.
	Learning about the background of coding was too incomplete to be useful
	The lectures on recursion were difficult to really learn the concept... but I think that is just because recursion in general is confusing. I don't think this is the instructors fault... and I still think it is a useful topic.
	n/a
	n/a
	Most
	I thought the material quickly became too challenging which was a hindrance to my learning.
	First few lectures were too advanced. We didn't start with "Hello world" like most intro to CS classes
	Not sure.
	The lecture in which students were required to ask each other for a certain number, which required another student to do the same to other students (in order to teach us recursion) was not helpful and was just tedious overall.
	all topics seemed to fit, except maybe the garbage collector topic
	UML diagrams
	I did not like the first two or so weeks using turtle. I felt like I learned little to nothing. I also did not like classes where there was expected group work and participation from the class, all of that seemed very useless. Lectures where Luther showed us how to program were decent though.
	turtle drawing
	the sundays assignment was difficult and tedious and I still do not understand how to successfully do it.
	Thought all lectures were valuable.
	Recursion

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	Recursion
	Recursion
	Recursion
	Recursion
	After homework 6 I feel like the lectures were more conceptual so I lost interest.
	After the second exam we seemed to have basically a filler lecture on nothing of particular import. Also, from the time we took the second exam until we started talking about recursion the lectures seemed unimportant.
	No thoughts.
	I thought that all of the lectures increased our coding abilities or broadened our knowledge of Java in some way.
	I don't recall any
	none
	none
	none
	The fractal business
	paper airplane
	fractals
	recursion
	recursion
	recursion
	recursion
	Going on a scavenger hunt type thing during lecture.
	Our introduction to recursion using turtles was very brief and rushed.
	turtle.
	Recursion seemed pointless to me because it just seemed like an unnecessarily more complex way to do loops (although it likely has some major advantages in more advanced CS classes)
	Learning methods or the different parts of method headers - understanding what static/non static ect.
	Useful coding lectures didn't seem quite that useful, mostly because it seemed like common sense
	None, it was all very useful.
	I think dynamic typing was not as useful as the other classes.
	Turtles
	When Mr. T went to fast during lectures, the material got hazy.
	I did not think the algorithm material was particularly useful or added to my understanding of computer science that much.
	the video game
	In general, just giving us complicated code and telling us to add the simple parts to it did not help us learn how to code.
	I think the recursion topics at the end were not useful because we did not have time to practice or fully understand them before the course ended and it confused me on some past subjects.
	I think at this level of CS, it is not as useful to learn about encryption.
	most of it
	UML diagrams, seemed kind of unnecessary

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	Some labs were not very worthwhile.
	I didn't really see a huge point to the "good code" section because as long as it works, it shouldn't matter in the small projects we're doing "how fast it runs."
	Recursion is not very useful.
	Encryption.
	Some of the recursion lectures.
	I found that a lot of topics in this class did not work. The style of teaching was very difficult, especially for someone like me with absolutely no computer programming experience. I felt the professor needs to slow down his pace to ensure that nobody is left behind.
	Some of the lectures towards the end of the course seemed discombobulated. For example, the lectures on how to search for things did not really seem to relate to anything that we had covered so far.
	The encrypted lecture. It was a nice break from traditional lectures but I think we could have gotten much more from a normal lecture.
	I never really understood how recursion would work in a practical application rather than just in pictures or simple problems. I might have understood it's uses better if we had completed HW 7.
	Some of the drawing that we did.
	The lectures about making code more efficient and "faster" was very vague and not very productive
	I thought that forcing us to stay within one format was bad and allowed for no creativity
	Some of the lecture about storage types
	drawing with turtle
	The Colored Maps homework confused me personally; I'm still not very solid on reading files and such so I'm not sure how useful that will be.
	Quick typing
	The turtle lesson
	The topics that I don't think will be as useful in the long run is the lectures on random small topics
	none that I can think of
	in class talking to people next to you
	I found the paper airplane lecture to be fun, but a bit pointless. I didn't learn anything from that lecture.
	recursion, more simple code helps with understanding
	Recursion was useful but it was just very hard to understand.
	There were no lectures that "did not work". However, I think that the recursion lectures needed more class time; it's a really tough topic to digest. (Although, to be fair, the reason why the topic was scrunched into a short time was a result of having so many snow days.)
	Some of the things not necessarily related to code, such as the Garbage Collector, dynamic and static programming languages, and efficiency/design of code didn't seem useful in the long run. For someone not continuing in programming, these topics don't really hold much relevance.
	They all seemed useful for different reasons.
	Fractals
	The ones where we had to draw the boxes with the methods.
	do while loops
	do while loops
	recursion
	binary codes
	Recursions really confuse me
	None of them.

The stuff for Test II seemed tedious and hard to get

Some of the lectures where we just wrote an example of code were not that helpful.

many of the lectures were confusing and WAY TOO FAST. these lectures not only did not work, they made me fall behind.

I didn't really see the point of recursion when that stuff can be easily done with loops.

getters and setters

The game programming is a very specific type of programming and I did not personally see it beneficial to my future.

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

~

Question Type: Multiple Choice

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-002, Tychonievich, Luther

Total	Every week (NA)	Every other week (NA)	Once per assignment (NA)	Rarely (NA)	Never (NA)
188	27 (14.36%)	30 (15.96%)	40 (21.28%)	59 (31.38%)	32 (17.02%)

Results for SEAS, 1000-level courses

Total	Every week (NA)	Every other week (NA)	Once per assignment (NA)	Rarely (NA)	Never (NA)
289	44 (15.22%)	47 (16.26%)	63 (21.80%)	86 (29.76%)	49 (16.96%)

9. How would you rate the availability of TAs?

~

Question Type: Likert

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
187	3.09	0.90	68 (36.36%)	82 (43.85%)	25 (13.37%)	10 (5.35%)	2 (1.07%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
287	3.13	0.88	112 (39.02%)	119 (41.46%)	40 (13.94%)	14 (4.88%)	2 (0.70%)

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

~

Question Type: Likert

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
186	3.04	0.84	57 (30.65%)	91 (48.92%)	29 (15.59%)	7 (3.76%)	2 (1.08%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
287	3.04	0.85	91 (31.71%)	133 (46.34%)	50 (17.42%)	10 (3.48%)	3 (1.05%)

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

~

Question Type: Short Answer

~

contributed by Tychonievich, Luther (lat7h)

Results for CS-1110-002, Tychonievich, Luther

Total	Individual Answers
93	*See below for Individual Results*

William and Kevin were the most helpful TA's!

sometimes they can't solve the issue

I really appreciated the large number of office hours that the TAs all had between them, and for the most part, they were very, very helpful. There were a couple of times where the wait list was excessively long; perhaps when there is a very hard assignment (such as maps), there should be more office hours.

They were extremely helpful

no

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	no It is hard because I feel like everyone has different ideas about the best way to code and so sometimes TAs wouldn't try and help you with your existing code but instead would try and get you to re-write your code to match their ideas of how the program should be written. Nah Most of them were solid. Some were incredibly unhelpful and even condescending, the rest were very helpful. Most were very helpful. Made themselves very available during office hours. Overall, the TAs were very helpful with the homework assignments. They are wonderful! They were good at explaining how to do things on a conceptual level. didn't It was frustrating how some people were very helpful and others weren't. Avoided asking them for help because it took to long to get help in stacks. Were very helpful in lab though. The TAs were very knowledgeable and helpful. When I went to office hours it was really hit or miss on whether or not I found the TAs helpful. The TAs were always very helpful and always did their best to help My lab TAs were very good. The older TA in our lab (3rd year CS major) was really really helpful. The younger TA (1st year) was pretty good but just not as knowledgeable because he had less experience. They did a nice job explaining things. Pretty helpful. They're very approachable and helpful Some TAs were very good at explaining, but others tended to be more impatient. There were a couple times when I had multiple questions but the TA would leave as soon as they answered one of my questions, or the TA would express impatience when I did not understand something the first time. very knowledgeable office hours works to a point, but every time I went, I had to wait over an hour to get my questions answered, which by that point, I had already come up with a solution. it might be helpful to have more TA's have office hours at the same time I really liked the efficient way that office hours were set up. However, some TAs were definitely much more helpful than others. Matt Pearson-Beck is a god among men. He brought us doughnuts to show his love Some TAs definitely knew the assignments better than others so the experience I had during the TA office hours varied depending on the TA that helped me. As a whole though, the TAs were a great resource for homework help. Very good! they were helpful but the wait list at times was too long There needs to be more TAs since so many students go to office hours N/A I felt like when explaining sometimes they wouldn't make it as obvious and would kind of beat around the bush Most of the office hours had too many students for the 2 TA's that were there - difficult to receive help. I went to office hours once and I had to wait a while for help on the HW and it wasn't even the day before it was due so having more than one TA available at office hours would be helpful.

~ QUESTIONS AND DETAILS ~ *~ ANSWER MATRICES ~*

none.

They were helpful in lab.

The TAs were fantastic, helpful, and patient, even though they were vey busy. i really liked my lab TAs too. This is confusing but interesting stuff, and I am thankful for their help.

I would like to ask that you not employ any TAs who are sexist. i had one TA who spent at least 15 minutes with a group of male students then when he can up to me he asked if I needed help or just wanted help. In addition this same TA treated me like a child when I did not understand the concept, I needed help with. no more arrogant TAs

perhaps having more TAs would be good because sometimes the line was too long to get help in the allotted time

I liked them in lab! But not very accessible office hours

my lab TA's were AMAZING

My TA's Sam and Monika were very helpful and patient, and really allowed me to better enjoy computer science, even though it was a very difficult course for me.

Helpful during lab, not during office hours

No.

Sometimes they were very unhelpful. Most TAs were fine, but you could wait for three hours to speak to someone and get very little help.

Every TA I interacted with was helpful, enthusiastic, and knew how to give hints without just doing it for us (ie they made sure we actually understood the material).

I really like the TAs of my lab section.

Some TA's wern't helpful at all and didn't seem to know material. Others were great and seem interested in helping.

Very helpful people.

I had one TA, a man with curly brown hair and glasses, who was EXTREMELY rude to me every time I attempted to get help from him. He made me feel stupid and incompetent, and almost made me never want to come to office hours again. He was condescending, unhelpful, and judgmental. It was the worst experience I have ever had with a teacher at this university.

Office hours are too packed

Sometimes the TAs were not as far as the students on the projects and did not understand all of it.

They were very knowledgeable in answering my questions

The TAs always seemed to know what was going on and how to help me.

Some TAs were super super helpful, whereas others just told you what you were supposed to do (as in restating the assigment), which wasn't really helpful because I understood what I was supposed to do, but I had trouble coding it.

some TAs would ignore the queue by helping those who had not waited

nope

nope

The TAs in CS 1110 and specifically my 8am lab time were phenomenal. I have nothing but good things to say. The TAs were very knowledgeable about every specific assignment and had excellent skills to convey their knowledge to confused students like myself. It was clear that they really knew their stuff. Additionally, I never encountered an impatient TA, even if I had a "dumb" question. I was thoroughly impressed with the TAs, their teaching skills and knowledge about CS.

I LOVED some of the TAs. My favorites were Scott, Casey, Jackie, Jim, Stephanie, and Matt.

The TAs in my lab were very helpful and knowledgeable, so I never needed office hours

The TAs are hit or miss. Some are excellent and some were very confusing. All were very knowledgeable, but some were better at assisting than others.

Thank you.

Some of the TAs were incredibly knowledgeable and were willing to help whereas others seemed a bit clueless.

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	None. None. None. most of them were very helpful and passionate about what they do. You have so many students seeing TAs at the same time that the TAs often rush to give whatever five words of advice they can think of first so they can run off to the next student. Use more of the TAs at once so the answers are better. After going to office hours once and waiting for help for two hours without avail, I gave up. In labs, however, ta s we're very helpful They are very helpful in lab. The overwhelming majority of the TAs were very helpful. However, there were a few that were incredibly rude to myself and other students. For example, one berated another student for having a slow computer, and told her to "call up mommy and daddy and ask them to send you a new computer". He is tall, has curly hair and glasses. I do not know his name, but he was the rudest TA I have ever encountered. No No No Some of them were very helpful, while others were not. for our final assignment we had 6 TA's look at our code, which was not working, and none of them could fix it. One last TA came and told us to make one simple change, which fixed everything. The TA's also will each give you their own way of doing things, so it can be very confusing when they are all telling you to do something different. The TAs were great. The TA office hours were extremely helpful because they were almost any time I needed help. The TAs were always available to help and most gave sufficient help for finishing larger assignments that I was having trouble on. Some TAs did not seem very helpful or really couldn't help much at all, but for the most part, the majority of the TAs were very helpful. The TAs were all wonderful! Extremely helpful and smart! Jim and Joe were very helpful during lab very helpful very helpful Many times, especially close to assignment deadlines, there were way too many people who needed help at once so the TA's couldn't give as good help because they couldn't stay and help for more than a minute or two because of the amount of people who needed help. I would suggest putting more TA office hours in the day or two before assignments are due to make up for this influx of people TA office hours are inefficient. I went only once because I waited for two hours for a TA to come tell me that he didn't understand our code and therefor could not help us. The TAs in my lab are great. They tried their best to help students. I liked most of the TA's but I wish they were all on the same page. I was already often confused about the material and when they said contradicting things it made it much worse. Just the ratio of students to TA's was large in Thornton, but the system for entering a help queue was fair and very effective! It was hard sometimes because the Queue would shut down because there were so many people needing help on some homeworks. Really depends on who you get. Some of them were fantastic, others...not so helpful.

~ QUESTIONS AND DETAILS ~ *~ ANSWER MATRICES ~*

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-1110-002

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
187	4.21	0.63	58 (31.02%)	111 (59.36%)	15 (8.02%)	2 (1.07%)	0 (0.00%)	1 (0.53%)

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)
1282	3.95	1.04	409 (31.90%)	589 (45.94%)	154 (12.01%)	62 (4.84%)	63 (4.91%)	5 (0.39%)

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-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
187	3.98	0.95	58 (31.02%)	84 (44.92%)	31 (16.58%)	8 (4.28%)	5 (2.67%)	1 (0.53%)

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)
1746	3.74	1.15	468 (26.80%)	599 (34.31%)	300 (17.18%)	144 (8.25%)	99 (5.67%)	136 (7.79%)

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-1110-002

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
187	4.11	0.96	69 (36.90%)	91 (48.66%)	10 (5.35%)	12 (6.42%)	5 (2.67%)	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)
1279	4.02	0.93	404 (31.59%)	620 (48.48%)	148 (11.57%)	74 (5.79%)	29 (2.27%)	4 (0.31%)

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-1110-002

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
184	4.39	0.82	100 (54.35%)	66 (35.87%)	10 (5.43%)	6 (3.26%)	2 (1.09%)	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)
1274	3.92	1.17	473 (37.13%)	449 (35.24%)	149 (11.70%)	98 (7.69%)	77 (6.04%)	28 (2.20%)

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-1110-002

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
186	3.49	1.06	28 (15.05%)	73 (39.25%)	48 (25.81%)	21 (11.29%)	10 (5.38%)	6 (3.23%)

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)
1277	3.30	1.17	158 (12.37%)	339 (26.55%)	299 (23.41%)	145 (11.35%)	100 (7.83%)	236 (18.48%)

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

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-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
184	3.98	0.86	49 (26.63%)	97 (52.72%)	27 (14.67%)	8 (4.35%)	3 (1.63%)	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)
1726	3.74	1.10	402 (23.29%)	676 (39.17%)	294 (17.03%)	113 (6.55%)	99 (5.74%)	142 (8.23%)

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-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
186	4.65	0.58	128 (68.82%)	53 (28.49%)	4 (2.15%)	0 (0.00%)	1 (0.54%)	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)
1729	4.16	1.00	714 (41.30%)	586 (33.89%)	175 (10.12%)	52 (3.01%)	60 (3.47%)	142 (8.21%)

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-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
185	4.44	0.68	96 (51.89%)	78 (42.16%)	8 (4.32%)	2 (1.08%)	1 (0.54%)	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)
1729	4.09	0.96	603 (34.88%)	677 (39.16%)	202 (11.68%)	56 (3.24%)	50 (2.89%)	141 (8.16%)

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-1110-002

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
183	3.37	1.19	22 (12.02%)	32 (17.49%)	35 (19.13%)	14 (7.65%)	10 (5.46%)	70 (38.25%)

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)
1276	3.58	1.14	195 (15.28%)	268 (21.00%)	212 (16.61%)	88 (6.90%)	50 (3.92%)	463 (36.29%)

21. The grading policy was fair.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
187	4.06	0.81	51 (27.27%)	109 (58.29%)	17 (9.09%)	7 (3.74%)	3 (1.60%)	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)
1735	3.59	1.25	403 (23.23%)	601 (34.64%)	207 (11.93%)	206 (11.87%)	141 (8.13%)	177 (10.20%)

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

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-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
187	4.22	0.81	73 (39.04%)	89 (47.59%)	18 (9.63%)	3 (1.60%)	3 (1.60%)	1 (0.53%)

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)
1727	4.04	0.96	556 (32.19%)	691 (40.01%)	210 (12.16%)	74 (4.28%)	43 (2.49%)	153 (8.86%)

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-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
186	4.39	0.73	92 (49.46%)	78 (41.94%)	12 (6.45%)	1 (0.54%)	2 (1.08%)	1 (0.54%)

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)
1738	3.87	1.14	536 (30.84%)	584 (33.60%)	218 (12.54%)	137 (7.88%)	81 (4.66%)	182 (10.47%)

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-1110-002

Total	Less than 1 (NA)	1 - 3 (NA)	4 - 6 (NA)	7 - 9 (NA)	10 or more (NA)
187	2 (1.07%)	51 (27.27%)	94 (50.27%)	32 (17.11%)	8 (4.28%)

Results for SEAS, 1000-level courses

Total	Less than 1 (NA)	1 - 3 (NA)	4 - 6 (NA)	7 - 9 (NA)	10 or more (NA)
1283	107 (8.34%)	559 (43.57%)	457 (35.62%)	127 (9.90%)	33 (2.57%)

25. I learned a great deal in this course.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-1110-002

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
186	4.18	0.88	75 (40.32%)	83 (44.62%)	18 (9.68%)	7 (3.76%)	3 (1.61%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
1274	3.77	1.26	442 (34.69%)	440 (34.54%)	162 (12.72%)	117 (9.18%)	113 (8.87%)

26. Overall, this was a worthwhile course.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-1110-002

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
185	4.20	0.95	84 (45.41%)	71 (38.38%)	17 (9.19%)	9 (4.86%)	4 (2.16%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
1279	3.74	1.33	476 (37.22%)	395 (30.88%)	150 (11.73%)	119 (9.30%)	139 (10.87%)

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

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-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
185	4.30	0.69	74 (40.00%)	96 (51.89%)	12 (6.49%)	2 (1.08%)	1 (0.54%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
1724	3.89	1.03	515 (29.87%)	733 (42.52%)	329 (19.08%)	65 (3.77%)	82 (4.76%)

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-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
184	3.65	0.92	34 (18.48%)	73 (39.67%)	58 (31.52%)	17 (9.24%)	2 (1.09%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
1727	3.82	1.00	475 (27.50%)	670 (38.80%)	434 (25.13%)	95 (5.50%)	53 (3.07%)

29. Overall, the instructor was an effective teacher.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-1110-002, Tychonievich, Luther

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
186	4.01	0.91	59 (31.72%)	84 (45.16%)	31 (16.67%)	9 (4.84%)	3 (1.61%)

Results for SEAS, 1000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
1738	3.71	1.14	487 (28.02%)	608 (34.98%)	408 (23.48%)	124 (7.13%)	111 (6.39%)

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

~

Question Type: Short Answer

~

contributed by Office of the Provost

Results for CS-1110-002

Total	Individual Answers
75	*See below for Individual Results*

Very good teacher, I had never taken a CS course (or ever programmed) and the course made me much more interested in CS

Luther is an OK teacher, the main reason I succeeded in this course was due to homework and learning from classmates, not his lectures.

Great course! Fun, informative, and useful.

Really enjoyed the course overall. I think the labs and homework assignments were especially effective for me in learning the concepts.

Giving us a large amount of code that we don't understand and telling us to add to it is not an effective way to have us learn the material. Start from the basics and build up.

Very difficult but worthwhile course.

Terrible teacher, but important class.

Tychonievich is very intelligent, but almost to the point where it is impossible to understand his explanations because he doesn't know how to "dumb things down" to people who have no programming experience. Homework assignments were doable but the last 2 are IMPOSSIBLE.

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*

I believe that the homeworks were too difficult compared to lectures and the tests. I always felt lost and confused during the homework because I felt like we did not learn the material needed. I think that more lab periods instead of lecture would be helpful in this course. Some of the lectures could be taught through a lab setting. That way the students can think and figure out problems for themselves with guides and TAs to help right then and there. I feel like I always learned a lot more trying to figure out how to solve the lab problems rather than passively watching someone else type code. Also, integrating homework topics into the lab would help too. I found it very beneficial when we did that for the last two partnered homework assignments.

The teacher went way too fast in class and was hard to follow.

Needs to really that he is teach a MASSIVE lecture hall and not a classroom of 30 students.

Fun course

Thank you for the course and all of your help

I am not a computer person, and I had a very hard time understanding and keeping up with the material. In class, Professor Tychonievich went to fast and it was hard to type and keep up with what he was doing in the lecture.

none

Homework #5 and #6 (especially #6) took A LOT of time. It would be helpful to have more than two TAs in lab for the last couple sessions when we were talking about recursion.

Due to the work required and the number of hours spent in actual class time, I'm not entirely sure why this isn't a 4 credit class. Otherwise, a very interesting class. Tychonievich was a very enthusiastic and knowledgeable teacher.

I truly enjoyed this class this semester. Even though I had already previously taken programming, this course helped reinforced what I already knew and as a result, I have improved greatly. The homework assignments were extremely beneficial to understanding the material and even though the last two assignments were much more difficult than the previous 4, they taught much more about code design and efficiency. The lectures were enjoyable and the availability for help was astounding.

You are a very smart professor who understands computer science more than I ever will. However, I feel your skills are not suited best for teaching the material to students. You move too fast at times through the material that you may find fairly simple, leaving students who haven't been exposed to computer science, like myself, confused and in need of more help. Also, when students raise their hands in class when you ask if anyone still doesn't understand what's been said or done, you still move on anyway frequently. I greatly respect your knowledge of computer science and I believe you should be working more so in the field for a company where I believe your impact in would be greater.

I felt like the grading on the last exam was unfair and the some of the more complicated material was covered as quickly as the easy material.

Tychonievich is one of the best (if not the best) instructor I have ever had. He is so knowledgable, enthusiastic, and well prepared. There was not a single question raised in class that he was not able to immediately answer. His lectures were always entertaining. I would recommend anyone to take his course.

hard to follow in class

Worthwhile course

This was a very well run course (especially for the subject matter). I always felt like I could get help and had access to the resources I needed to do well.

I didn't learn much from the class. In class everything would be simple, but the homeworks would be annoying and it was very difficult to figure out what to do with just the simple stuff we did in class since the questions were just like, write a program that calculates this and does this , and in class, we would have discussed only ints, loops, etc. I know you have to think to get how to do it, but still. Some people aren't that smart

Tychonievich has been one of the best teachers I have had. His course was always engaging, the extra activities supplemented the course really well (the encryption hunt was excellent!), and during office hours he was extremely helpful. When a couple friends decided to take CS 1110 next semester, I recommended that they take his class.

Professor Tychonievich did not put in enough effort on organizing this course. He is extremely knowledgeable in the topic of Computer Science and assumes his students know way too much. For those with CS experience from high school, this class should be fairly straightforward. However, for those that don't have such experiences, this class' difficulty level skyrockets after several weeks in the class. This class made me very upset that I was not accepted into CS 1112. Nevertheless, this class would be fairly useful if I understood what was going on.

Although I didn't think much of CS before coming into the course, I feel like I learned a lot of useful information throughout the process.

Great class overall

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	For me, intro to programming was a very challenging course. Tychonievich was a decent lecturer, but I thought the material was too difficult to be covered just in lecture and the lecture did not really help on the homework assignments.
	This course was fantastic!! During my time at UVA thus far, this was probably my favorite course. First, the instructor, Luther Tychonievich, was exceptional - very well prepared for lecture, very knowledgeable in the subject, passionate for the subject material, and very very adept at explaining that material to students. His homework submission and feedback system also made for a great experience. The textbook was also amazing (my favorite course text thus far at UVA). The tests were also a great test of the program knowledge we had covered; their administration with a large time clock displayed was also great. Hands down an amazing academic experience with an amazing instructor. If I was given the option to sign on now to take all my future programming courses with professor Tychonievich, I would have a hard time turning that offer down.
	I came in knowing absolutely nothing about computer science and finished the year very confident about my programming skills. This was my most fun class and it was the only class I looked forward to doing the homework in. It really showed me how much fun problem solving can be. Tychonievich was a funny and effectively clear teacher. He couldn't have done a better job and was a big reason for my new-found love of CS. It's so great to see the rewards of your hard work immediately. I may be transferring majors to CS after taking this course.
	I felt like the tests didn't accurately gauge my knowledge. Also, some of it seemed unnecessary, because eclipse catches small errors. I didn't like the way that part of the class was handled--I think partial credit needs to be more available because otherwise it's unfair. However, I learned a lot from the homeworks mainly. I wish that we would cover more in the course.
	I appreciate that the homework was worth a large percentage of the grade. Professor Tychonievich is very engaging and entertaining. However, he coded way to fast for me to keep up, and it was difficult to follow many of his coding procedures in lecture.
	I think Tychonievich was great, but during lectures he talked way too fast and it was hard to follow a lot of times. He wouldn't stress on essential stuff either so it was hard to know what material he was emphasizing as important.
	10/10 would take again
	I enjoyed most of the homework assignments. Some of them were very challenging, but after completing the assignment, I felt that I had a much better understanding of the concept.
	This is a great course to take as Computer Science is a field that is growing exponentially. Although the concepts are difficult, they are so interesting and the instructor as well as the TAs are great to go to for help or just to talk to. Thank you! I did find that some concepts were a bit rushed, but that was most likely due to the snow days.
	the curve on the last test seemed to unfairly reward those who did poorly over those who did well by curving the lower grades up much more
	Good, worthwhile class. Confusing at times. The individual homework really helped me understand the topics that I may not have understood completely in class because it forced me to work it out and understand it on my own. HOWEVER, THE PAIRED HOMEWORKS DID NOT MAKE ME UNDERSTAND THE MATERIAL, AND I WOULD NOT RECOMMEND CONTINUING THEM. It's not that I meant to be lazy or do less work than my partner, but because I was slow to understand the concepts in class, I tended not to contribute as much in the paired homeworks. I thought Prof. Tychonievich did a good job for the most part.
	Homework 5 was very difficult to understand, specifically the instructions were poorly organized. Homework 6 was much better.
	sometimes the professor would go to fast and try to speed through difficult topics
	Loved this class and loved the professor.
	I thought Professor Tychonievich was a good instructor, though he sometimes coded too fast on his computer that it was hard to keep up with the code and still understand what was going on (I was often just mindlessly copying his code to keep up). For the most part he was good at answering questions during class but sometimes he didn't fully explain his answers. Also, he occasionally sped through material too quickly or just talked over a noisy lecture hall which made it difficult to pay attention. It would have been helpful if he had tried to wait for everyone to stop talking before beginning.
	Homework assignments took way too long. I think 3 per week would have been sufficient to get the points across yet not completely overwhelm us with CS work.
	I believe this course was way too difficult as a intro to CS course and the expectations for this course were too high. The homeworks were consistently too difficult and the only students who did well in this course were those who had had prior knowledge of CS which isn't fair since this is an intro course. I am incredibly disappointed in the way this course was taught and I hope it is revised in later years.

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	I enjoyed this class a lot, and it definitely piqued my interest in computer science. Tychonievich clearly knew what he was talking about, and for the majority of the time his lectures were very helpful. Furthermore, his website was well laid out and easy to access. Sometimes the homework assignments were excessive; my classmates and I would have to essentially block of an entire day for some of the weeks because we were unsure how long the assignments would take us. Furthermore, it seemed as though we were putting in a lot of effort for a small amount of points toward our grades. In addition, it was sometimes difficult to follow the code in class, and when I got lost, I could not catch up.
	The course website was the best study and learning tool that I had in any of my classes this semester, and the homework was also extremely helpful to my learning
	good course, good course
	I learned absolutely nothing in class. Tych. moved quickly and typed away his code. Sometimes he was jumping around...which for CS, I feel makes everything 10X more difficult. The slides in which he wrote himself were not helpful, jumbled, and could not be referred back to for help later
	It was really hard
	The class was a challenge because of the different ways of thinking and the new concepts involved, but it was rewarding to understand a topic and actually get the homework assignments to work. Some of the homework assignments were very hard, but after completing them I had a better understanding of the topics. It would have been beneficial to know where points were taken off on homework assignments instead of just receiving the grade.
	For Tychonievich: Please try to explain recursion better. It is still hardly understood by most of the students.
	None
	I really really did not like computer science, mostly due to my part, but I really struggled with this class and really wish I did not have to take it as it has nothing to do with what I want to do in my future. It is going to hurt my GPA, and was a subject matter that my brain could not wrap itself around.
	Tychonievich is the reason I put CS as my second choice major, before this class I had completely written it off. I'm very happy that I took it
	This is a good course for learning programming.
	Professor Tychonievich knows a lot about computer science and really understand everything he is talking about during class. However, he usually speaks too fast, making it hard to follow his logic all the time.
	Even though I had to wake up early for my 8am lab, it was worth coming in every week. Thanks for a great lab!
	None.
	None.
	None.
	I think focusing on the specifics of some code, or having explanation on them would have proved helpful.
	I felt actual lecture wasn't that useful. The teacher would move way to fast on difficult topics. Most of my learning was self-taught out of the book or through the lab
	Prof Tychonievich was an overall great professor and I enjoyed his class
	n/a
	This is a very useful course.
	Professor explains the materials too fast so I missed some of the things that I should've known after each lecture. Also group-work is a good way to learn how to work with other people but it did not work for some people.
	I had never taken a computer science course before this one and I thought I would hate it. I soon realized how much I liked the subject. The homework assignments were well structured and I learned most of the material through the homework. I am taking the next level of CS because of this course and am seriously considering minoring in CS.
	Great professor. Great course. Very informational. Had a great time as well
	all these questions were asked in the other eval
	This course was a lot of fun as we learned to programmed. I really like the end of the year game as the culmination of our work because it really incorporated a lot of what we have learned

~ QUESTIONS AND DETAILS ~	*~ ANSWER MATRICES ~*
	His lectures were so scattered and he moved through the material way to fast for students who have no background in CS I love how Luther is as a teacher. I just cannot take an interest in this course, so I still feel as if I learned nothing. It was hard for me, but it was extraordinarily taught.