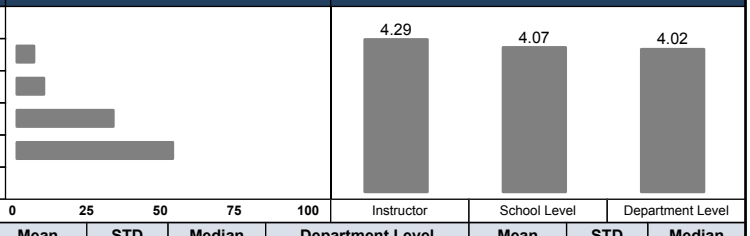
JHU - Krieger School of Arts & Sciences / Whiting School of Engineering
ASEN.2017.Fall

Course: EN.601.220.04.FA17: Intermediate Programming

Instructor: Rohit Bhattacharya *

1 - The overall quality of this course is:

Response Option	Weight	Frequency	Percent
Poor	(1)	0	0%
Weak	(2)	2	6.45%
Satisfactory	(3)	3	9.68%
Good	(4)	10	32.26%
Excellent	(5)	16	51.61%
N/A	(0)	0	0%

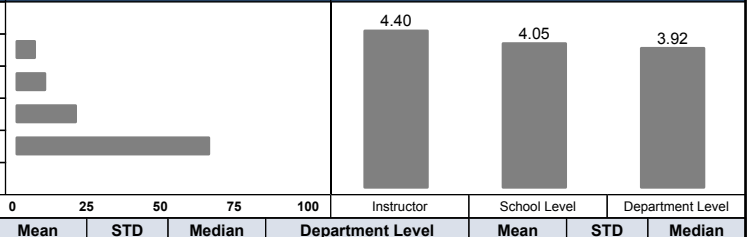


Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
31/34 (91.18%)	4.29	0.90	5.00	10,238	4.07	0.98	4.00	1,807	4.02	1.02	4.00

2 - The instructor's teaching effectiveness is:

Rohit Bhattacharya

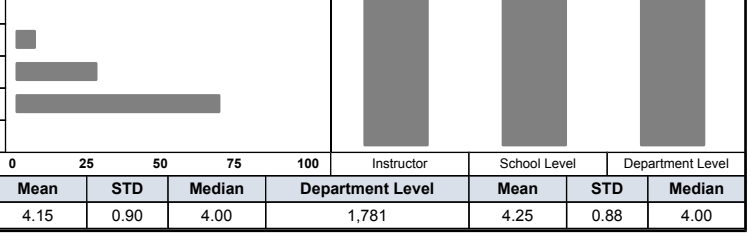
Response Option	Weight	Frequency	Percent
Poor	(1)	0	0%
Weak	(2)	2	6.67%
Satisfactory	(3)	3	10%
Good	(4)	6	20%
Excellent	(5)	19	63.33%
N/A	(0)	0	0%



Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
30/34 (88.24%)	4.40	0.93	5.00	10,146	4.05	1.05	4.00	1,785	3.92	1.14	4.00

3 - The intellectual challenge of this course is:

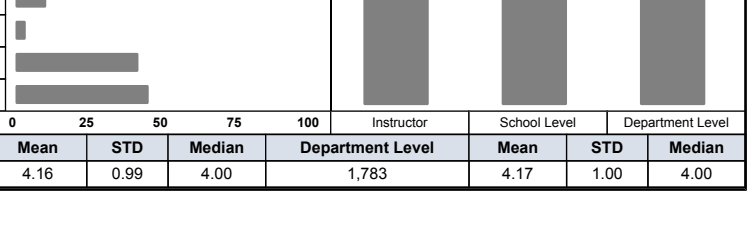
Response Option	Weight	Frequency	Percent
Poor	(1)	0	0%
Weak	(2)	0	0%
Satisfactory	(3)	2	6.67%
Good	(4)	8	26.67%
Excellent	(5)	20	66.67%
N/A	(0)	0	0%



Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
30/34 (88.24%)	4.60	0.62	5.00	10,136	4.15	0.90	4.00	1,781	4.25	0.88	4.00

4 - The teaching assistant for this course is:

Response Option	Weight	Frequency	Percent
Poor	(1)	0	0%
Weak	(2)	1	3.33%
Satisfactory	(3)	3	10%
Good	(4)	1	3.33%
Excellent	(5)	12	40%
N/A	(0)	13	43.33%



Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
30/34 (88.24%)	4.41	1.00	5.00	10,133	4.16	0.99	4.00	1,783	4.17	1.00	4.00

JHU - Krieger School of Arts & Sciences / Whiting School of Engineering
ASEN.2017.Fall

Course: EN.601.220.04.FA17: Intermediate Programming

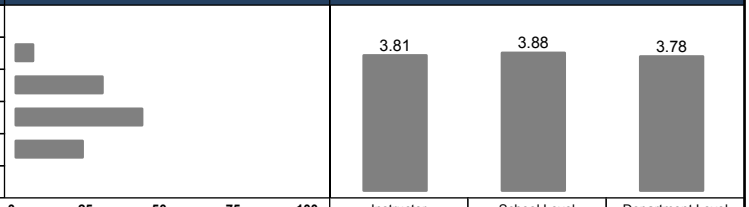
Instructor: Rohit Bhattacharya *

5 - Please enter the name of the TA you evaluated in question 4:

- Adriana Donis
- Adriana, Reena, Francis
- Adrianna and Frances and Reena and Taha
- Shijia Liu
- Taha Baig, Reena Sarkar
- Shijia
- Don't know their names...
- All the CAs in my section: Taha and Francis especially
- Don't even know any of their names really
- N/A
- I just went to office hours so just in general
- Taha Baig
- Dont know

6 - Feedback on my work for this course is useful:

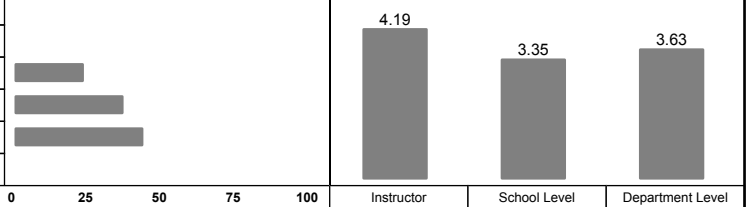
Response Option	Weight	Frequency	Percent	Percent Responses	Means
Disagree strongly	(1)	0	0%		
Disagree somewhat	(2)	2	6.45%		
Neither agree nor disagree	(3)	9	29.03%		
Agree somewhat	(4)	13	41.94%		
Agree strongly	(5)	7	22.58%		
N/A	(0)	0	0%		



Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
31/34 (91.18%)	3.81	0.87	4.00	10,091	3.88	1.08	4.00	1,767	3.78	1.14	4.00

7 - Compared to other Hopkins courses at this level, the workload for this course is:

Response Option	Weight	Frequency	Percent	Percent Responses	Means
Much lighter	(1)	0	0%		
Somewhat lighter	(2)	0	0%		
Typical	(3)	7	22.58%		
Somewhat heavier	(4)	11	35.48%		
Much heavier	(5)	13	41.94%		
N/A	(0)	0	0%		



Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
31/34 (91.18%)	4.19	0.79	4.00	10,120	3.35	1.03	3.00	1,773	3.63	1.00	4.00

JHU - Krieger School of Arts & Sciences / Whiting School of Engineering
ASEN.2017.Fall

Course: EN.601.220.04.FA17: Intermediate Programming

Instructor: Rohit Bhattacharya *

8 - What are the best aspects of this course?

• the programming assignments were a lot of fun and tbh the exams were also fun just because they were logic puzzles

• Great professor, fun assignments, lots of practice using C/C++, very project and homework-based which was helpful for learning and practice

• Assignment based, really manageable if you have good coding principles. Love the in-class demos and the sense of community created from the class. Course Assistants are great and love helping people. Code doesn't have to be perfect, it just has to be good enough and handle the most important edge cases, which is lenient. Definitely try to take this class before Data structures if you can. They really hold your hand on what to do.

• Learn c/c++. Usefully

• The professor is incredible. He engages with students, goes above and beyond to help them, and most of all exhibits a positive attitude that helps make the class quite an enjoyable experience.

• Provides a very solid foundation for programming C and C++. Teaches information that is slightly out of the scope, but interesting nontheless

• Some of the projects were fun.

• Learn practical tools for software engineering. The projects are really awesome, especially the Kmeans one.

• The homework were interesting an reflect practical applications.

• The professor is very caring and does his best to make sure that we know what we are learning. He's a really nice guy too!

• I got many projects to practice my coding skills with.

• I think Rohit explained all the concepts really well, and all the programming assignments were really good applications of our knowledge, along with building some practical skills. I also liked the written homeworks, since they were a bit of a break from the longer coding assignments, and helped focus on some specific syntax knowledge and the like.

• Experience working with large-scale programming projects.

• Useful to learn about how to program in C / C++

• Great course in programming. I learned way more than I expected.

• Learned C and C++

• The professor and the material.

• I believe it was intellectually challenging, but it was also very interesting. It was also taught incredibly well; I retained a lot and I was always excited to go to class. For his first semester teaching, I think Rohit did an incredible job, and he genuinely cared about everybody's performance which was really great.

• This course really helped cement my understanding of programming fundamentals, including style and memory allocation. It was really great to be introduced to two languages that are applicable in industry and in life.

• The fun and interesting programming tasks and design ideas. The ability to learn new programming languages and to think like a programmers.

• Learn C and C++ Much more manageable than data structures HW assignments are very reasonable

• Projects are meaningful

• good intro to C/C++

• I learned about not only the computer languages but also about how a computer works. Material was tough but fair.

• The homework problems are really interesting and engaging.

• I liked how we got to learn a language that has many practical applications outside of school.

JHU - Krieger School of Arts & Sciences / Whiting School of Engineering
ASEN.2017.Fall

Course: EN.601.220.04.FA17: Intermediate Programming

Instructor: Rohit Bhattacharya *

9 - What are the worst aspects of this course?

- sometimes the work load seemed heavy but that was mostly when i was procrastinating
- Sometimes insufficient was time given to complete homework assignments. The course content moved very quickly and sometimes the slides posted did not cover all the content. The midterms were difficult and we did not have much previous practice writing code in a timed environment.
- If you aren't looking to develop software engineering skills, don't take this class.
- unit test
- Lots of homework, but that's not really a bad thing; it's just the nature of the beast I guess.
- Grading is slightly picky.
- The timelines given for some projects didnt make sense. Homework grades were not returned in a timely manner meaning you would have to submit the next homework without feedback from the first.
- Can have a heavy workload sometimes.
- There should be solutions to homework after submission used for studying.
- I wish there were more problems similar to what's on the exam for practice. I don't think we get sufficient practice for that.
- The workload is heavy, and the fact that this course doesn't have a curve really gets me worried.
- Nothing really.
- Projects are extremely time-consuming.
- No complaints to speak of.
- N/A
- The late grading and not knowing what they are looking for.
- Honestly, the meeting time. 4:30 to 5:45 isn't the best but it could definitely be worse!
- The homework programming assignments were quite long and took up a lot of my time.
- Homework submission can be very annoying. Sometimes, there is technical issues and you can potentially lose marks that you deserve.
- Teacher is kind of petty TAs are not very helpful Class is not that useful to attend
- Pace is too fast; no time to consolidate knowledge
- challenging
- Sometimes I felt that the classwork was too guided to the point where I wouldn't learn much when I was doing the work. Perhaps the classwork could be restructured so that it assesses student's knowledge better.
- The difficulty is a lot higher than any previous peogramming courses.
- Some concepts took longer to grasp and wish I had more time to fully understand it.

10 - What would most improve this class?

- quicker feedback on assignments
- More practice writing timed code, more time to do the homework / late day policy for homework.
- This class is great, don't change it. Maybe make the midterm slightly easier!
- lower workload
- Better CA's would be good. There are one or two helpful CA's, but the majority seem to not really know how to help me. Having the professor going around and helping people usually alleviates this problem, but still it would be nice to have more helpful CA's.
- A more structured note system. Lecture slides are often jumbled and disorganized, making studying difficult.
- Getting homework back before having to submit the next homework.
- More answer key releases
- See above
- We should get more time in class to complete our projects/assignments; we should get a chance to ask questions on our assignments in class.
- Not much that I can think of.
- Shorter lectures, longer timeframes to complete homework after they are assigned (i.e. 1.5-2 weeks instead of 1). More use of algorithms in projects.
- I think the course is great as is, I enjoyed it a lot.
- N/A
- More information about edge cases and solutions posted online for every exercise and homework.
- I can't think of anything really :)
- Perhaps if programming assignments were due every 7-8 days instead of the 5-6.
- More interactive lectures during class time instead of just using slides almost every single class. Answers to every in class exercises should be also released as soon as possible so students can study from them.
- A teacher (not a PhD candidate) More in-class exercises
- Probably shouldn’t learn two languages in one semester
- workload adjustment
- More practice problems for the exams would help a lot.
- Certain assignments were very difficult and would take more time than the given week to complete.
- Personally, I felt like more example codes would be helpful. The in class exercises were usually good but sometimes the lecture didn't feel enough.

JHU - Krieger School of Arts & Sciences / Whiting School of Engineering
ASEN.2017.Fall

Course: EN.601.220.04.FA17: Intermediate Programming

Instructor: Rohit Bhattacharya *

11 - What should prospective students know about this course before enrolling? (You may comment on any aspect of this course such as assumed background, readings, grading systems, and so on.)
• the programs take some time so start early but overall it was an incredibly useful and enjoyable class
• It was a very good course, students should be prepared to put a lot of time into assignments. It will be challenging but manageable if you practice on your own outside of class and go to office hours.
• Having taken java helps, especially with learning about C++'s OOP. Would recommend.
• have enough time
• Be very familiar with preferably multiple languages prior so you are able to learn new ones quickly. Also, don't get complacent during the first few weeks when you are just learning C basics; it gets MUCH harder.
• Good class!
• Make sure to keep up with the material and go to class.
• Have good background in basic coding.
• They should know java and how to program well in java. They should also know what's expected in the class.
• This course assigns you a project every single week so you should be prepared to really devote a decent amount of time into this course.
• This course expected and required more coding background than I expected. I was already an experienced programmer, but coming in after only taking Intro Java/AP CS would be tough I think.
• Set aside enough time to complete projects.
• could be a little bit challenging if you haven't learned anything in C before
• You should be very comfortable with programming logic coming into this course. It is an intense course, but is completely do able if you put enough time into it. Prepare thoroughly for the exams and start the homework early.
• N/A
• Lots of work but manageable if do on time.
• This is a difficult class, but if you put the work in it's also one of the most rewarding ones. If you're interested at all in programming, I definitely recommend taking this!
• Be prepared to learn a lot over the course of the semester and the course workload. Programming assignments are harder than they initially appear, so make sure to start them early. Pay attention in class; exam material reflected what we learned in class the most.
• Students should have a moderate background about coding and programming. Students should expect a lot of time spend on completing the homework, the exams are not very difficult, though.
• Don't sign up for the staff section. The staff section isn't harder, it's just less useful. More is definitely the superior educator.
• Its workload is really heavy; don’t take more than two courses at this level at the same time
• heavy workload
• This class is fairly work heavy, but you end up learning a lot. If you did java, you should be fine with the background.
• It is a lot more difficlt than Introduction to Programming and it is a huge leap in diffuclty.
• meh
• Expect to spend a considerable amount of time for this course. It's okay to feel behind and even be a little behind, but just make sure to start homeworks early.