# JHU - Krieger School of Arts & Sciences / Whiting School of Engineering
# ASEN.2013.Fall

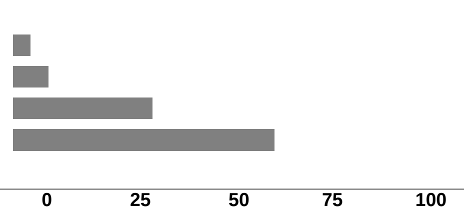
**Course:** EN.600.120.01.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

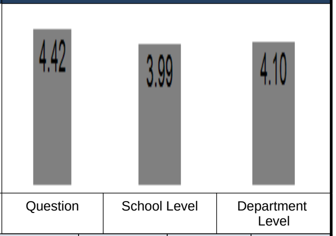
**1 - The overall quality of this course is:**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Poor | (1) | 0 | 0% |
| Weak | (2) | 1 | 3.85% |
| Satisfactory | (3) | 2 | 7.69% |
| Good | (4) | 8 | 30.77% |
| Excellent | (5) | 15 | 57.69% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means

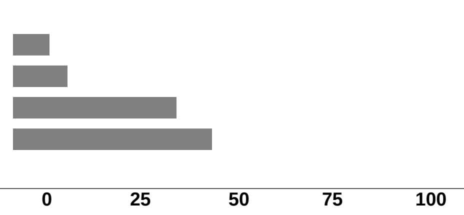


| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 26/27 (96.3%) | 4.42 | 0.81 | 5.00 | 7,734 | 3.99 | 0.96 | 4.00 | 994 | 4.10 | 0.90 | 4.00 |

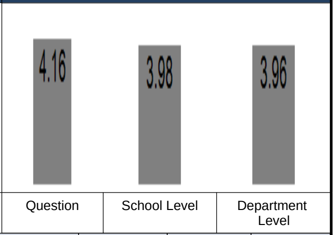
**2 - The instructor's teaching effectiveness is:**
**Yair Amir**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Poor | (1) | 0 | 0% |
| Weak | (2) | 2 | 8% |
| Satisfactory | (3) | 3 | 12% |
| Good | (4) | 9 | 36% |
| Excellent | (5) | 11 | 44% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means

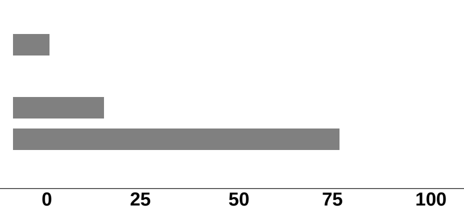


| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 25/27 (92.59%) | 4.16 | 0.94 | 4.00 | 8,088 | 3.98 | 1.03 | 4.00 | 996 | 3.96 | 1.04 | 4.00 |

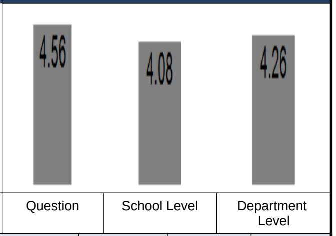
**3 - The intellectual challenge of this course is:**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Poor | (1) | 0 | 0% |
| Weak | (2) | 2 | 8% |
| Satisfactory | (3) | 0 | 0% |
| Good | (4) | 5 | 20% |
| Excellent | (5) | 18 | 72% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means



| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 25/27 (92.59%) | 4.56 | 0.87 | 5.00 | 7,697 | 4.08 | 0.90 | 4.00 | 987 | 4.26 | 0.82 | 4.00 |

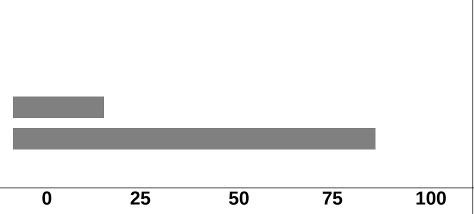
**Course:** EN.600.120.01.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

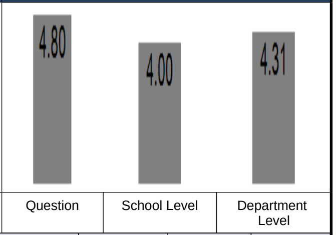
**4 - The teaching assistant for this course is:**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Poor | (1) | 0 | 0% |
| Weak | (2) | 0 | 0% |
| Satisfactory | (3) | 0 | 0% |
| Good | (4) | 5 | 20% |
| Excellent | (5) | 20 | 80% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means



| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 25/27 (92.59%) | 4.80 | 0.41 | 5.00 | 7,705 | 4.00 | 1.07 | 4.00 | 990 | 4.31 | 0.92 | 5.00 |

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

| Return Rate | 20/27 (74.07%) |
|---|---|

- Everyone was great -- Dano and Graham were the people I received help from most frequently.
- Jeffrey DalaTezza
- Jeffrey DallaTezza
- There were many of them, all were excellent.
- Amy Babay, Jeff Dallatezza
- Jeff
- All
- Amy, Jeff and the CA's
- Amy Babay
- All of them
- Malvi Hemani
- Amy Babay/ Jeff DallaTezza
- Jeff
- Jeffrey Dallatezza, Jennifer Lu, Graham Smith
- All of them
- Jeffrey
- Amy and Jeff
- Amy, Jeff, Jen Lu
- Jeff / Amy/ All of them
- Amy/David

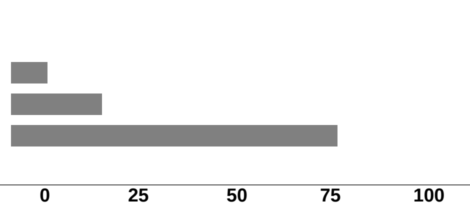
**Course:** EN.600.120.01.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

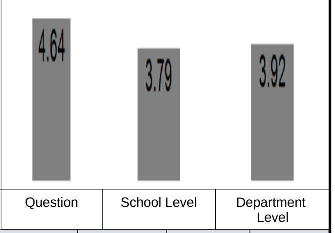
### 6 - Feedback on my work for this course is useful:

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Disagree strongly | (1) | 0 | 0% |
| Disagree somewhat | (2) | 0 | 0% |
| Neither agree nor disagree | (3) | 2 | 8% |
| Agree somewhat | (4) | 5 | 20% |
| Agree strongly | (5) | 18 | 72% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means

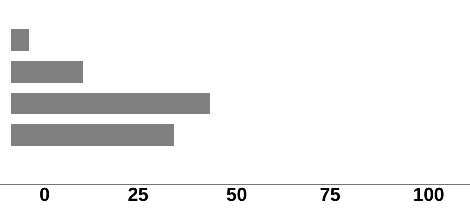


| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 25/27 (92.59%) | 4.64 | 0.64 | 5.00 | 7,688 | 3.79 | 1.07 | 4.00 | 985 | 3.92 | 1.07 | 4.00 |

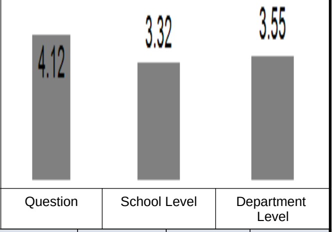
### 7 - Compared to other Hopkins courses at this level, the workload for this course is:

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Much lighter | (1) | 0 | 0% |
| Somewhat lighter | (2) | 1 | 4% |
| Typical | (3) | 4 | 16% |
| Somewhat heavier | (4) | 11 | 44% |
| Much heavier | (5) | 9 | 36% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means



| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 25/27 (92.59%) | 4.12 | 0.83 | 4.00 | 7,697 | 3.32 | 1.01 | 3.00 | 989 | 3.55 | 1.05 | 4.00 |

### 8 - What are the best aspects of this course?

| | |
|---|---|
| Return Rate | 22/27 (81.48%) |

**Course:** EN.600.120.01.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

- The projects are a challenge and frustrating, but very rewarding once completed. The number of course assistants allows for you to always get a quick response with any problems you are having. The small class size seemed very appropriate for the course, as it requires more personal interaction with professors and the TA/CA's than most other courses.

- The interaction with the TA's always looking at your source code and telling you what to do better.  Being able to pull up your code on a computer in class and ask a TA for help on the latest project.  I liked the lengthier projects which really let you get into the code, as opposed to weekly assignments for JAVA which were over in maybe 10 minutes.

- It was extraordinarily helpful to be able to work on our assignments in class. This allowed us to get useful feedback from the professor and ta's that was specific to our individual issues. The help email address was also useful. The printouts of the code were also very helpful as reference material for assignments.

- The assignments are designed so that you actually learn how to program. Yair and all of the TAs are amazing. They care so much about how much we learn and they offer so much of their time to help the students.

- The multitude of TA's allowed this course to feel very personalized, even though there were 90 kids enrolled over 3 sections.

- You WILL leave this course knowing how to program, Yair is an extremely passionate Professor that really cares about his students.

- The intellectual challenge was the best part of the course. All of the projects can be accomplished by applying the material taught in the lecture at a much higher level. The instructor had a team of course assistants and TA's who provided meaningful assistance and feedback and helped students grow in understanding the material and applications of designs.

- Great assignments. I feel that I have a good handle on both C and C++ because of the beneficial projects. Also, the amount of people available to assist us in our work was amazing. The email service for questions was great for submitting questions because you knew someone would be able to get back you with great help. The sample code on the servers was always a great reference.

- Interesting topics, I really think that I became a better programmer. Also, all of the TA's were very helpful and nice.

- Great feedback. Great professor. TAs and CAs are very helpful and available. The assignments are fun if you're into that kind of thing.

- I think this course definitely challenges you to understand the programming languages thoroughly and build structures from their roots.  I feel I have a much better grasp of the languages than I might have taking a similar course elsewhere because the projects were challenging and the professor made smaller lecture sections instead of a much larger one.  TAs and the instructor himself were able to make their rounds during class to help each individual.

- Professors/TAs genuinely care that you learn the material. They were always accessible if I needed help.

- The feedback on the projects is detailed enough to allow you to improve your code quality and design logic over the course of the semester. I really enjoyed the fact that we had a handful of large projects rather than weekly assignments.  I felt that a huge part of what I took away from this class came from tackling large design problems.  Having sections of 30 allowed us to work on projects periodically during class. A lot of general design questions were best addressed in this environment (ex. technicalities of how the data structure should function). Having the TAs/CAs present during these meetings was extremely helpful, especially if I was too self-conscious to email the help email.

- The amount of feedback and the face-time with TAs and the professor

- Projects are fun

- I think programming is just really fun. I really like doing the class assignments because it gives me a chance to solve problems and get practical experience. Also, it was really nice that this course was taught hands on in a computer lab, instead of in a lecture hall. Also, the professor and TA's are very accessible so help is easy to get when your program isn't working (through the email help hotline).

- The TAs for this class are excellent.  There are many that are available at many times throughout the day, and they are all very helpful.  The class help email is also very useful.  The TAs respond extremely quickly to any email you send them.

- The content was great and I got a lot better at programming.  Feedback from everyone was very detailed and allowed me to make changes that improved the quality of the work I submitted.

- The teacher and TAs genuinely cared about your work and improvement and someone was always willing to invest a lot of time and effort into helping you. The class setting was very helpful in that only about 30 people were in your section at one time and that everyone had computers in front of them that allowed them to program along with the teacher and have interactive class sessions. It allowed for a more personal learning experience that lead to a better understanding of the material. In addition, the way the projects were designed allowed me to learn the material through execution as opposed to memorization. It was a much more interesting and effective approach to teaching.

- Spends a lot of time allowing us to write code directly and to ask questions and get feedback during class. The only way to learn to write code is to actually do it, and this class gives many opportunities to do so.

- The teaching lab was definitely the best aspect. Because it only allowed for 30 or so students at a time, I was able to realistically speak with at least 2 maybe 3 TA's during every workshop session and get feedback on how I was approaching the programs. The class size is the second best part as it obviously facilitated the increased interaction.

- The challenge of putting things together. Seeing yourself grow. The small programming environment kept me focused which was very good.

**Course:** EN.600.120.01.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

**9 - What are the worst aspects of this course?**

| Return Rate | 22/27 (81.48%) |
|---|---|

- Yair is clearly very passionate about his work and the class was awesome but he can be abrasive and short-tempered.

Obviously the improvements I made in programming required a lot of time committed to this class -- (disproportionality high, upwards of 200 hours of time spent on homework for the course) but I would not recommend a change in workload because I felt it was time well spent.

- The course load is very heavy and the grading is very harsh. On the other hand, the harsh grading improved my skills as a programmer tremendously.

- Reading through code can be a little dry although there are not many alternatives to this.

- Lecture days where we would sit for an hour or so just going over code examples. Though I was not at all disappointed by these lectures, they were the most monotonous part of the course, yet I understand that they are completely necessary for learning. Going over swaths of code is always going to be a little boring in any class or context, but Yair makes it very easy to follow along and is really good at connecting older concepts to newer ones.

- Some of the projects take a lot of time to do and can be very frustrating.

- It was difficult to flag a TA's attention during class because so many people needed help. The C++ lectures went really fast and a lot of the subtleties of the syntax were lost on me.

- We learned new code by being shown pre-written examples in class. This was often quite dry and boring. It would have also been useful to learn a bit more about make files.

- Projects took a lot of time. This course had the biggest workload of all my courses despite the fact that it was not my most difficult course.

- Bugs, errors, lectures can be a bit dull.

- I can't think of anything.

- The professor for this class is awful. He acts rude to students who ask questions. He is very intimidating, almost to the point where there were some days that I would just rather not come to class than deal with this. His teaching style does not work in the slightest. Instead of teaching what specific code is used for, he just quickly run through example code, only slightly mentioning how it can best be used. Also, many classes are spent just sitting in the room working on projects, which seems like a waste. This time could have been used to better teach the concepts. The professor also greatly favors C as opposed to C++, which is very clear in his teaching. While he comes up with his own code and does a moderately okay job at teaching the C portion of the class, C++ is a mess. He simply rips code straight from a book, going through it extremely quickly and tries to cover so many topics in a short amount of time that it very difficult to sink in all the info. Additionally, for some reason the professor must also have some strange desire to teach the data structures course, because that's what half of this course is. There is a whole class designed to teach this, so it is not very important to learn this in Intermediate Programming. The projects are also ridiculously long. The workload is unparalleled by any other 100 level class I know of. Be prepared to spend all of your free time working on these projects instead of having fun or relaxing. The midterm was also a poor way of testing programming knowledge.

- It's sometimes to figure out what efficiency/design components the graders are looking for. Sometimes, I felt like I could make my projects more efficient if I would have learned different ways how to do so.

- I know some people that had trouble keeping up with the assignments, but if you plan your time right they're very doable.

- I think that learning a brand new language requires practice. Non-graded practice. I felt as if we watched lectures for a couple days then were thrown into a big, tough project right away.

- The assignments sometimes took too much time to do little things that the assignment required, but were not necessarily helping to learn anything (i.e. The extensive class application in the final project). I almost can't say anything was a worst aspect though...

- The professor presented all the material to be learned without really going through it, leading to a lot of confusion.

- The intense workload

- A lot of work

- Only having four, but very intensive, projects was stressful. I feel that I would have learned more with shorter, but more frequent assignments. Also, I think that more physical office hours would be more beneficial than emailing for help. Also, the prompts for projects are unclear sometimes, and a seeing a grading rubric ahead of time would help. Also, making up a data structure for the final project was a little much. I spent much of my time figuring out how the fake structure worked.

- N/A

- You will learn to hate segmentation faults. It was hard on the professor and the TAs to manage the size of the class. They definitely deserve better resources to teach this course.

- At times it was difficult to follow how we went over the code in class. I believe some of the syntax could have been learned more efficiently through exercises like the packages for inheritance and complex numbers for overloading operators, etc.

**Course:** EN.600.120.01.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

**10 - What would most improve this class?**

| | |
|---|---|
| Return Rate | 20/27 (74.07%) |

- This class is excellent.

- Utilizing Piazza, in my opinion, would improve the quality of this course many times over. Although the cs120 help email could still be utilized for specific coding issues, Piazza could then be a central location for all of the general questions and announcements for the course. I found myself spending too much time searching through my email for a specific answer to a question that had been asked. I'm sure many other people did that a few times throughout the semester so having a central place where we could all ask general questions (and answer each other's questions, possibly) would be extremely advantageous to the learning in this class.

- The midterm was useful in learning to think and apply concepts quickly, but I would suggest scheduling time to do a 2 hour midterm instead of 1. Also - making the assessment strategy clearer (maybe with a print rubric) for the midterm would be helpful.

- The professor should teach material instead of just reading source code.

- Possibly more little exercises in class like the package inheritance exercise, in the beginning of the semester to help us get familiar with c and with dynamic memory.

- Focusing a bit more on certain aspects of programming. Really beat down a few concepts as opposed to covering a wider range.

- I'm aware that the number of people interested in taking this course has grown too large to fit in the same profile as previous years. Still, I hope that in the future this course will continue to be taught the same way: hands on, in the lab, with lots of feedback from the Professor and TA's. Up until now, all of my courses were taught in big lecture halls, so having a smaller class size and one on one time with the Professor + TAs in this course was really refreshing. I would be sad to see this course become another big lecture hall class, especially since programming is such a hands on subject.

- More established office hours to go in and discuss problems.

- Physical office hours, and more, but shorter assignments.

- More clarification on what is expected of the students (a clearer rubric for grading) and a clearer explanation of some concepts (especially within inheritance).

- I think having smaller practice assignments to acclimate everyone to the different concepts and intricacies of the languages would have been really helpful going into projects. It certainly helped me in high school and I think that is something that is doable here. Also, the professor started to have us do little tasks in class toward the end of the semester, which was nice but could have been done earlier on.

- More TAs, making it easier to get help, more required checks on your code because a lot the time your code compiles and runs and you think its working when really you have some logic flaw and your results are wrong. More talk on what is a good design document, what is good style etc.

- I think it would have been useful to have a few more small exercises to introduce new concepts. In the class, it seemed like to only programming we did was on our projects, where all of the concepts we had learned were supposed to work together. If we had had several smaller programs to demonstrate some of the concepts, it may have helped some student do better on the larger assignments.

- The sections should be split up into smaller ones. I do not know how professors taught this course with a lecture style. We really benefited from one-on-one help.

- I would have like to have face to face feedback after one/some of the projects to expose some of my mistakes that I made. I thought the meeting at the end of the year would have been even more useful if it occured earlier in the semester.

- More in class assignments would have been nice (see 9).

- Actually teaching material, rather than just handing it out to students via paper handouts.

- I may get crap from my friends, but I would love to see more smaller assignments/projects in addition to the ones we did. My problem with CS is that I sometimes forget design and syntax if I'm not constantly working on code during the week. My first programming course did a great job of having two weekly assignments that made learning how to program awesome.

- I think the midterm carried too much weight. On top of that, we didn't get the grade break down of how the midterm would be graded, so it was difficult to know exactly what the professor was looking for. Even if you knew what you were doing, you could get a bad grade if you didn't do what he was looking for.

- less work

**Course:** EN.600.120.01.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

**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.)**

| | |
|---|---|
| Return Rate | 23/27 (85.19%) |

- Only four projects throughout the semester!

- This course is a lot of work. It is structured very well and is run by people that really want to see you succeed.

- It is very fast paced and time intensive.  In order to do well, one must have the time and patience to put into projects.

- If you want to learn programming in C and C++ as well as programming mentality--take this course. Don't take it if you're not willing to put in the hours and hours necessary to put together a presentable, high quality program.

- Although a strong programming background is not required (I only took Intro to Java at that point), patience and willingness to troubleshoot for a few hours in a row is the most important quality to have when going into this class. The first programs will not take too long, but the deadlines will soon be longer on later projects, meaning that the time spent on each program will have to go up. Be organized and always seek out help if you're stuck. If there were ever a class where a second pair of eyes were a god-send, it is this one.

- This class is very time-consuming, but extremely rewarding.

- Use the TAs as much as possible because they can help you get a good grade.

- Be prepared to spend all of your time on these assignments.

- course load and grading system

- Be ready to solve problems. In this course, you'll learn to program some basic (and not so basic) data structures, which can be pretty tricky. Plan your program ahead or it will be very hard to tell if your program is doing what you think it's doing. The email help-line is a great, but if you've made some mistake in the logic of your program, the TAs won't be able to guess the mistake. So when something isn't working, experiment. Break your code down and find the specific part that isn't working.

- Don't procrastinate, as the projects are always more complex than they may seem at first.

- I wasn't at all expecting so much work from the class.  So just be prepared for that.

- A lot of work, but you will learn a lot and become a better programmer. Definitely helps to know data structure before hand.

- Basic programming knowledge and concepts would be helpful as pace is very quick in terms of going through code. Make sure to start on assignments early and work through them by spreading the workload. Try to keep up with the suggested deadlines and finishing should not be an issue.

- It is a lot of work and don't be afraid to ask for help from the TAs.

- This class expects you to be comfortable the basics of programming already. There was not much time spent reviewing simple syntax or style.

- Take this course with Yair and Amy if you can! It is far from impossible to do well in the course. The textbook is helpful if you are not entirely comfortable with programming yet, but Google can be just as helpful. Get started on projects early, so that you can get as much help from the TAs as possible.

- I wish I had taken advantage of the TAs more often. Don't be shy to ask questions or approach them with bugs. Don't wait until the final presentation to figure out that they honestly want to help you :)  Time management is huge. After the first 2 projects, my grades became a direct reflection of the amount of time I put in relative to how much time I should have.

- This course requires a lot of your time and a LOT of individual learning.  Failing to master concepts simply given to you in handouts can lead to problems later on in the course.

- Lot of work. Fun. Utilize the TAs and seek out help from other CS students.

- The book is useless.  Any information you want to know about programming can be found with a simple Google search.  Start projects as soon as you get them.  They take an insanely long amount of time.

- Great class if you love programming.

- You will be working within a unix shell so it would be good to familiarize yourself with that a little before entering the course if you have never worked with it before. Grading is harsh but in a very helpful way.

# JHU - Krieger School of Arts & Sciences / Whiting School of Engineering
# ASEN.2013.Fall

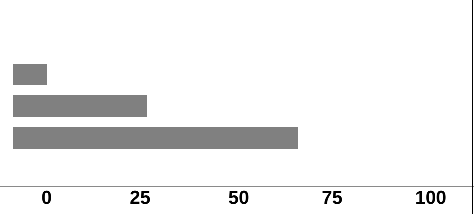
**Course:** EN.600.120.02.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

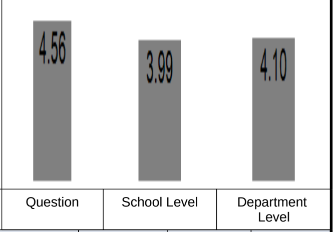
**1 - The overall quality of this course is:**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Poor | (1) | 0 | 0% |
| Weak | (2) | 0 | 0% |
| Satisfactory | (3) | 2 | 7.41% |
| Good | (4) | 8 | 29.63% |
| Excellent | (5) | 17 | 62.96% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means

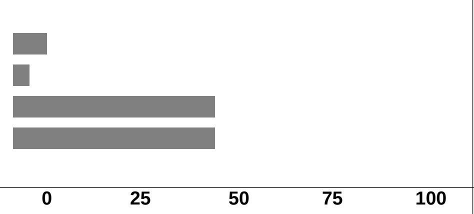


| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 27/31 (87.1%) | 4.56 | 0.64 | 5.00 | 7,734 | 3.99 | 0.96 | 4.00 | 994 | 4.10 | 0.90 | 4.00 |

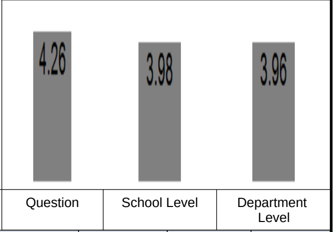
**2 - The instructor's teaching effectiveness is:**
**Yair Amir**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Poor | (1) | 0 | 0% |
| Weak | (2) | 2 | 7.41% |
| Satisfactory | (3) | 1 | 3.7% |
| Good | (4) | 12 | 44.44% |
| Excellent | (5) | 12 | 44.44% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means

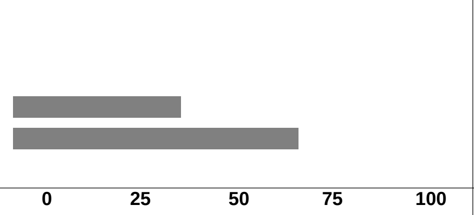


| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 27/31 (87.1%) | 4.26 | 0.86 | 4.00 | 8,088 | 3.98 | 1.03 | 4.00 | 996 | 3.96 | 1.04 | 4.00 |

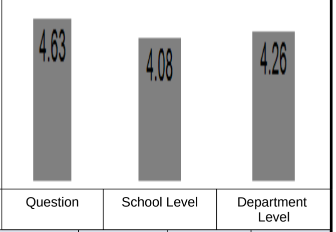
**3 - The intellectual challenge of this course is:**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Poor | (1) | 0 | 0% |
| Weak | (2) | 0 | 0% |
| Satisfactory | (3) | 0 | 0% |
| Good | (4) | 10 | 37.04% |
| Excellent | (5) | 17 | 62.96% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means



| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 27/31 (87.1%) | 4.63 | 0.49 | 5.00 | 7,697 | 4.08 | 0.90 | 4.00 | 987 | 4.26 | 0.82 | 4.00 |

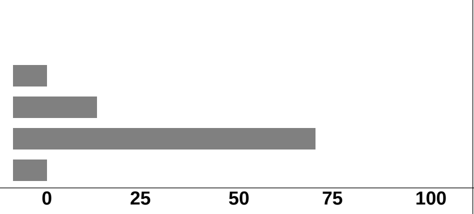
**Course:** EN.600.120.02.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

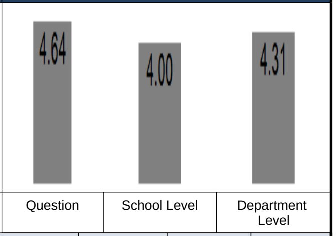
**4 - The teaching assistant for this course is:**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Poor | (1) | 0 | 0% |
| Weak | (2) | 0 | 0% |
| Satisfactory | (3) | 2 | 7.41% |
| Good | (4) | 5 | 18.52% |
| Excellent | (5) | 18 | 66.67% |
| N/A | (0) | 2 | 7.41% |

Percent Responses



Means



| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 27/31 (87.1%) | 4.64 | 0.64 | 5.00 | 7,705 | 4.00 | 1.07 | 4.00 | 990 | 4.31 | 0.92 | 5.00 |

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

| Return Rate | 22/31 (70.97%) |
|---|---|

- Amy Babay, Jeff Dallatezza, Tyler Cloutier, Jen Lu
- Jeffrey Dallatezza
- Jeff, Amy
- Jeff
- Amy Babay
- Jeff
- Jefferey DallaTezza
- Jeff Dallatezza, primarily, but definitely the rest of the staff as well
- JeffreyDallaTezza
- Jeff DallaTezza
- Jeff & Amy
- Amy, Jeff, Tom, Dano, All the CAs
- Amy Babay
- Jeffrey DallaTezza
- Jeff
- Amy Babay, Jeffrey DallaTezza
- Jeffrey DallaTezza
- Jeff
- Amy Babay
- Jeff
- ?
- Jeffrey Dallatezza

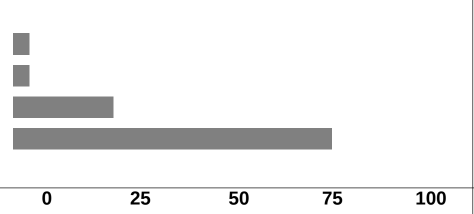
**Course:** EN.600.120.02.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

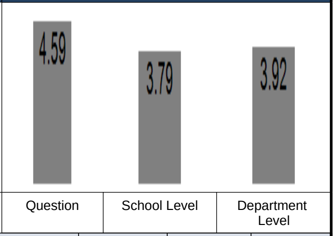
**6 - Feedback on my work for this course is useful:**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Disagree strongly | (1) | 0 | 0% |
| Disagree somewhat | (2) | 1 | 3.7% |
| Neither agree nor disagree | (3) | 1 | 3.7% |
| Agree somewhat | (4) | 6 | 22.22% |
| Agree strongly | (5) | 19 | 70.37% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means

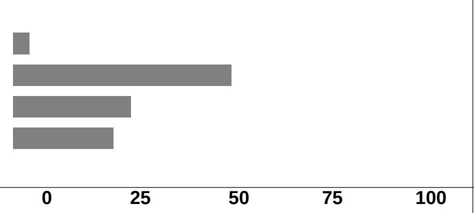


| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 27/31 (87.1%) | 4.59 | 0.75 | 5.00 | 7,688 | 3.79 | 1.07 | 4.00 | 985 | 3.92 | 1.07 | 4.00 |

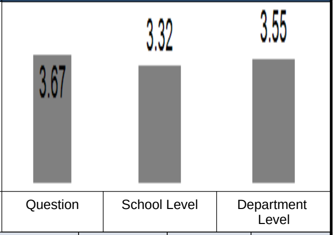
**7 - Compared to other Hopkins courses at this level, the workload for this course is:**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Much lighter | (1) | 0 | 0% |
| Somewhat lighter | (2) | 1 | 3.7% |
| Typical | (3) | 13 | 48.15% |
| Somewhat heavier | (4) | 7 | 25.93% |
| Much heavier | (5) | 6 | 22.22% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means



| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 27/31 (87.1%) | 3.67 | 0.88 | 3.00 | 7,697 | 3.32 | 1.01 | 3.00 | 989 | 3.55 | 1.05 | 4.00 |

**8 - What are the best aspects of this course?**

| Return Rate | 24/31 (77.42%) |
|---|---|

**Course:** EN.600.120.02.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

- TAs are really nice and helpful.

- Challenging projects.
Good feedback on work.

- great feedback on projects. lots of personal attention from the professor and TAs.

- Teaches how to program effectively and cleanly. Lots of attention from teachers and teaching assistants.

- I liked that the instructor cared about each student and gave personal feedback on each of the projects. I liked that the class is project based and each concentrated on the specific learning tasks. I liked that there was a help e-mail and extensions were often given when deemed appropriate.

- The TAs, instructors, and CAs are incredibly helpful and willing to answer questions 24/7. They provide a wealth of resources to help you succeed in this very challenging class. Help line is AMAZING. Projects are interesting; I never felt like a minute of my time was wasted in class.

I would try to keep the class as small and hands-on as possible. I understand that this requires a lot of resources, but based on my own experience, the students leave with an incredible set of programming skills.

Lastly, I am very impressed by how hands-on the instructors were. This was my first semester in college and I never expected them to be so helpful and to care so much about each individual student. I really appreciated the time they spent to help me.

- This class really makes you feel that if you put work and motivation you are going to be a better programmer by the end. I liked that the course was divided into three sections so that we could get one to one assistance from the TA's and Dr. Amir. This model should continue on for other CS courses.

- Very interactive learning.

- The well spec'd projects early in the semester.

- You spend an equal amount of time on C and C++ so you learn both pretty well. Also the TAs were AMAZING. Easily more helpful than Yair. I often went out of my way to ask the TAs for help rather than Yair.

- I enjoyed learning C & C++ and how to handle memory. I also like that the TAs are very easy to approach even when I think my questions are really dumb.

- Interesting course material, professor is engaging and friendly and really wants to help you and make sure you do well in the course and come out a better programmer.

- In-class coding with TAs available for help was very useful to me as a student.

- Professor is knowledgeable. He makes the knowledge very clear.

- I learned a lot of concepts in C/C++ programming while gaining experience in creating somewhat complex data structures; It was nice to learn more than one thing at once.

- Hands-on projects to teach, and the TA's and professor really care about your progress in the class. They come around during lecture and ask how your progress is going.

- Lots of interesting challenges. If you manage to make it through then you really learn something.

- Yair, Amy, and the rest of the course staff were amazing, and definitely cared about our success.

Dividing the class into small groups was excellent, and I completely agree with Yair that large classes are not as good as small classes. I think it was the right decision, even if he found himself stretched a little thin by 3 classes. (2 groups a semester might be better; or perhaps 2 small groups taught by Yair and 1 larger group taught by someone else?)

I liked how frequent 1-on-1 interactions were, even if I didn't always take advantage of them (I sometimes felt like I couldn't explain what I was doing to other people and wanted to just smile, nod, say I was doing fine, and get on with fixing bugs). But having the option to ask a TA or even Yair himself for help was definitely great.

- The work was really engaging.  Because it was project based, there was a lot learning by dong which I think is more effective than learning through lecture only.  There was a lot of one-on-one guidance.  At least three TAs were available each class to help with any problems.  Class was very hand-on and the professor was very approachable and easy to listen to.

- The course goes through many important aspects of programming with both the languages and the concepts of logic. The teacher works with students to make sure that they understand what they are doing during lectures, and during the workshop sessions, CAs work one on one with students to help them with any concepts they still do not understand. It is very comprehensive and extremely rewarding.

- The best aspect of the course was that it teaches C and C++. Also the TA's that were there to answer questions were very helpful.

- The projects. I thought they where very fun and allowed me to learn the languages rather quickly

- Given the subject matter, the smaller class size certainly helps in making lessons easier to understand.

- pointers.. haha jk

**Course:** EN.600.120.02.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

**9 - What are the worst aspects of this course?**

| Return Rate | 22/31 (70.97%) |
|---|---|

- The projects do take some time to do.

- The way that they teach the course is purely with example code. This method of teaching in my opinion is not engaging and does not fully explain many aspects of the language. The professor recognized that looking at pages of code was boring. Later on when one wishes to write a program they have this example code, however, the examples may not always help with problems.

- Some time class lecture could be a bit boring, just going over code

- Some subjects are brief compared to others, but minor.

- instead of4 big hw assignments having a little every week would be better.

- The course is scary...{(-_-)}

- the lectures fairly dry and boring. we were never really told how a design document should look, so getting points off for them was somewhat frusturating.

- Classtime is not spent well.

Even when class time was not explicitly just "we're here to help you with our projects", I often found myself working on projects anyway- either because I already understood what we were talking about or because I simply found Yair's lectures (going over ~10 example programs line by line to explain what they did) boring and I knew I could teach myself the topic when working on a project.

Having everyone always logged into computers does not promote paying attention in class, too.

I was also... this only applies to me, but I feel the need to note it anyway- I didn't like the "if you can't solve a bug in 15 minutes, talk to us about it" rule that Yair imposed at the beginning of the semester. First of all, I ignored it- but secondly, you don't learn by having someone else fix your own problems! You learn by debugging them yourself and learning what they were.

Perhaps it works if the TAs explain how they found the issue- looking at line numbers, running debuggers, printing out information, and so on- but still. I think it would be more useful to teach us how to debug than to solve our problems for us.

- Some of the projects - like the Deli - lack specifications. Also, this project felt strange - what exactly were we making, a simulation? A deli management backend? Improving these projects would help.

- Yair doesn't explain things particularly well, however asking a TA to explain things later clears up confusion but it makes the class itself feel like a waste of time. Yair is quicker to criticize or just seem generally disappointed than he is to provide helpful responses. In class a student asked for help with his constructor method and Yair responded "write it so it does what you want." Hopefully that kid knew to ask a TA next.

- The professor is not clear on the grading rubric. I didn't know I could lose points for it until I lost points for it.

- Slightly ineffective teaching style.

- Skipping lecture will bite you in the butt, don't do it.

- Project instructions could be more clear.

- A lot of the lecture portions of the course consisted of the professor or TA reading code to the class. I thought this was an ineffective way of teaching. It was difficult to follow along with what they were saying when they would review 10-12 programs a class.

- I did not like that the midterm accounted for such a large part of the class yet only a few days were spent on the project.

- Intense workload. Some classes are really light with the learning material while in other classes we cover a lot of material really quickly, perhaps evening it out would be better/easier to follow.

- Projects usually only give opportunities to have good scores or poor scores (no middle ground..)

- Heavy workload.

- Some additional time complexity analysis would be helpful. It was covered, but only very briefly.

- Even though I did relatively well on the midterm, it was a lot to do in the time allotted.

- Nothing!

**Course:** EN.600.120.02.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

**10 - What would most improve this class?**

| Return Rate | 24/31 (77.42%) |
|---|---|

- See 9.

- While I see the benefit of having fewer but more involved assignments, I think it would have been better to have more assignments that cover the same material but in smaller chunks. For instance, our first C++ assignment involved writing/using data structures and multiple classes. None of use had ever programmed in C++ so we had to learn EVERYTHING while doing one assignment. I think it would've been better to stretch into 3 assignments: one where you write a data structure, one where you write multiple classes, and one that puts it all together in a final result. I found it very stressful and intimidating to go into an assignment where I didn't know the language and still had to do so much. Starting off with that attitude and fear resulted in a product that I knew wasn't my best work.

- More structured teaching.

- Some smaller possibly handwritten assignments between larger projects to understand why certain data structures are more efficient in both time and space.

- I believe that this class would be improved if they taught the programming languages in a different way that is more in depth and engaging.

- More in-class work and smaller class sizes

- Recitation section. Also, not teaching programs by just presenting the code. Should do a hands on way of learning.

- Large projects expected from reviewing smaller example code. Not a terrible aspect but it's hard to go from looking at small samples of code to creating one massive data structure. Regular Course Assistant office hours would also be very helpful.

- It would have been nice to have a little more specificity in the feedback on my projects. I often lost points for having an unclear design file, but I was never sure what I should do to make it more clear.

- More hands-on learning exercises even if it was not a full-on project. Less code being read at the class. Group projects would have also been appreciated.

- A few extra days to complete the assignments would be good, although a lot of time is given in the first place.

- No suggestions.

- teaching it the way joanne teaches intro java.

- Hope the professor will redesign the grading scale to allow students to make mistakes. After all, it's only an introduction level course, or simply, a course that aims to get the students learn.

- I agree that design is a big part of coding but I think that there's a lot of stuff you can't learn just through the design process. There should still be emphasis on taking the time to write out the code and learn through errors that come up so that in the future when similar errors do pop-up, you will have the experience to draw upon and fix that error. Although this is a minor suggestion -- I thought that the course was extremely instructive and I enjoyed it a lot.

- more programming excercises, as opposed to just the projects

- N/A?

- Projects details made more clear (maybe posted online). Groups for projects would be fun.

- I would like more office hours and more coding to understand the lectures, less reading other people's code.

- A bit more variation in project topic. Also, maybe have a bit of time--a day or two--between the submission of one project and the assignment of a new one. I often felt mentally exhausted.

- Better use of class time- perhaps we could write programs in class instead of talking about ones Yair had already written? He did this a few times, but I think it should be done more often.

Or, like Joanne, he could write the programs in front of us, and we could collaborate on them as a class.

Otherwise, I think it's generally a fantastic course.

- n/a

- Keep it divided!

- Make a clear rubric which states exactly what kind of things will lead to points deduction.

**Course:** EN.600.120.02.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

**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.)**

| | |
|---|---|
| Return Rate | 22/31 (70.97%) |

- Go to class.

- Start the assignments early. It helps a little if you treat them like multiple assignments. DO NOT go to Yair for help, the TAs are MUCH better at explaining things and won't be visibly disappointed if you ask a question.

- The grading is harsh. But you do learn a lot in this course.

- Should be willing to think.

- A background in Data Structures would be helpful. Be prepared to spend several hours on each assignment.

- Should be comfortable with programming because the course moves fairly quickly and assumes some familiarity with a lot of the topics covered.

- They should know that they should have a basic understanding of some of the more basic data structures since data structures were frequently used as examples.

- LEARN TO USE VI/VIM.

- Heavy workload from occasional projects -- make sure you start early!

- 1.Being scared by your grades is a part of the course, but you learn from your own mistakes. Talk to TAs as often as you can, just to be sure you are on the right track.
2.It's good to struggle with your bugs yourself, but make sure to balance your our courses.

- design is very important

- This course is pretty fair, but the difficulty ramps up towards the end. Late work is 0-tolerance, and you're allowed to submit as much as you want. Definitely a good idea to take this one with Yair, he's one of the best professors I've had at Hopkins (and I'm a senior)

- The grades are based on five projects and one midterm. The professor has everything planned out. Most of the time the classes start with 30-45 minutes of lecture (through looking at code) and then the rest of the class to work on projects and to ask for help. I liked the course a lot.

- Make sure you have a solid understanding of basic programming before taking this course, because you will be forced to learn languages quickly

- This class is Baptism by fire. It is very challenging, but also very rewarding. Unlike other courses I took this semester, I felt like I was constantly learning something new. Be sure to start the projects early, since they do take a lot of time to complete and are mentally trying. And as cliche as it sounds, don't give up, because it's really tempting to do so. Just remember that you will leave this class and have the skills to program anything. The CAs, TAs, and instructors are also great resources and are very encouraging.

- Students should be willing to ask for help if anything is not understood, as each lesson builds upon the previous.

- It very much depends if Yair or Peter (or someone else?) teaches it, obviously. I can't speak to Peter teaching it...

If you have taken Data Structures first, be prepared for a pretty easy class (assuming you did well in data structures). Four out of the five projects, including the final project, revolve around data structures- at least two of which were explicitly

But don't underestimate the class because of this.

If you don't have Data Structures, be prepared for things to be a bit more difficult. You'll not do well on the first project, probably- your code needs to have a good style too.

Definitely ask for help if you need it, though. Otherwise you're wasting all the resources offered to you here.

- Work hard, and you will get a lot out of the course

- It is a lot of work. Especially out of class. The projects take time. Going in to see TAs can be incredibly helpful.

- Students should have a solid understanding of a programming language, preferably Java. It is also helpful to understand data structures because they are relied on heavily in this course.

- I came into this class knowing it was going to be a challenge for me since I wasn't the most comfortable with all of the basics, but I would strongly recommend knowing at least the basics. I know some other people that took several more classes than I had before taking this one and had a much easier time.

- a lot of work. dont skip out at ALL on classes.

# JHU - Krieger School of Arts & Sciences / Whiting School of Engineering
# ASEN.2013.Fall

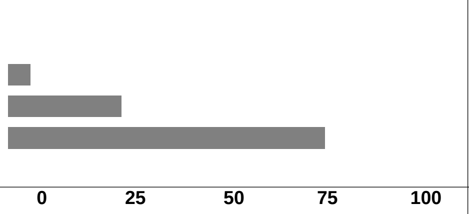
## Course: EN.600.120.03.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

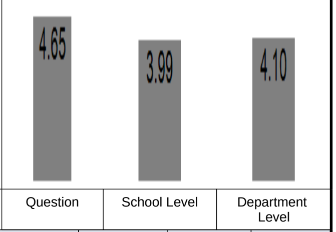
### 1 - The overall quality of this course is:

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Poor | (1) | 0 | 0% |
| Weak | (2) | 0 | 0% |
| Satisfactory | (3) | 1 | 5% |
| Good | (4) | 5 | 25% |
| Excellent | (5) | 14 | 70% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means

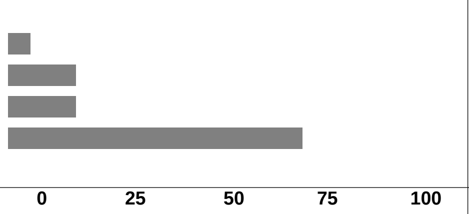


| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 20/25 (80%) | 4.65 | 0.59 | 5.00 | 7,734 | 3.99 | 0.96 | 4.00 | 994 | 4.10 | 0.90 | 4.00 |

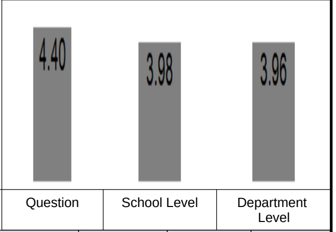
### 2 - The instructor's teaching effectiveness is:
**Yair Amir**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Poor | (1) | 0 | 0% |
| Weak | (2) | 1 | 5% |
| Satisfactory | (3) | 3 | 15% |
| Good | (4) | 3 | 15% |
| Excellent | (5) | 13 | 65% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means

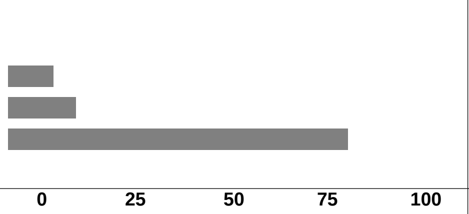


| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 20/25 (80%) | 4.40 | 0.94 | 5.00 | 8,088 | 3.98 | 1.03 | 4.00 | 996 | 3.96 | 1.04 | 4.00 |

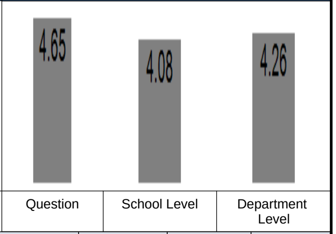
### 3 - The intellectual challenge of this course is:

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Poor | (1) | 0 | 0% |
| Weak | (2) | 0 | 0% |
| Satisfactory | (3) | 2 | 10% |
| Good | (4) | 3 | 15% |
| Excellent | (5) | 15 | 75% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means

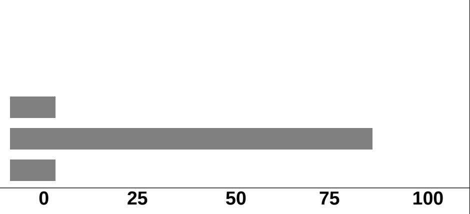
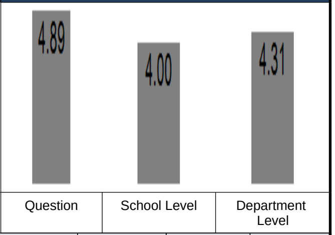


| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 20/25 (80%) | 4.65 | 0.67 | 5.00 | 7,697 | 4.08 | 0.90 | 4.00 | 987 | 4.26 | 0.82 | 4.00 |

**Course:** EN.600.120.03.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

**4 - The teaching assistant for this course is:**

| Response Option | Weight | Frequency | Percentage | Percent Responses | Means |
|---|---|---|---|---|---|
| Poor | (1) | 0 | 0% |  |  |
| Weak | (2) | 0 | 0% | | |
| Satisfactory | (3) | 0 | 0% | | |
| Good | (4) | 2 | 10% | | |
| Excellent | (5) | 16 | 80% | | |
| N/A | (0) | 2 | 10% | | |

| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 20/25 (80%) | 4.89 | 0.32 | 5.00 | 7,705 | 4.00 | 1.07 | 4.00 | 990 | 4.31 | 0.92 | 5.00 |

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

| Return Rate | 17/25 (68%) |
|---|---|

- Jeffery DellaTezza
- Amy Babay and Jeffrey Dallatezza
- Amy Babay
- Jeffrey Dallatezza
- N/A
- Amy, Jeff, Dano, Jen etc.
- all of them
- Jeffrey Dallatezza
- Jeffrey Dallateaza
- Jeffrey DallaTezza
- Jeff DallaTezza, Tom (not sure of last name), Ben Cohen, Malvi Hemani
- Jeff and Amy
- Jeff DallaTezza and all other TAs
- All of the TA's were extremely helpful.
- All of them
- Jeff Dallatezza
- Jeff.  Also Tyler, although not the official ta was Very good.

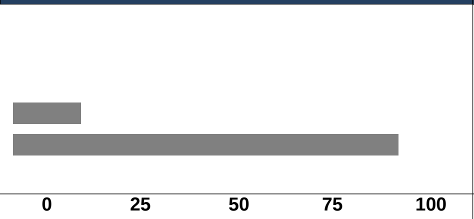
**Course:** EN.600.120.03.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

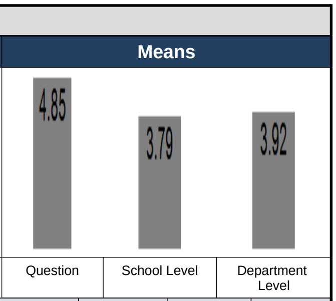
**6 - Feedback on my work for this course is useful:**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Disagree strongly | (1) | 0 | 0% |
| Disagree somewhat | (2) | 0 | 0% |
| Neither agree nor disagree | (3) | 0 | 0% |
| Agree somewhat | (4) | 3 | 15% |
| Agree strongly | (5) | 17 | 85% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means

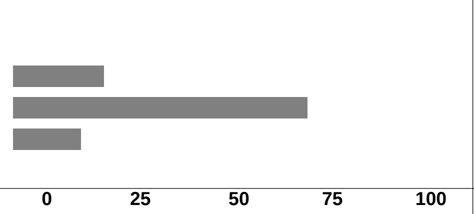


| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 20/25 (80%) | 4.85 | 0.37 | 5.00 | 7,688 | 3.79 | 1.07 | 4.00 | 985 | 3.92 | 1.07 | 4.00 |

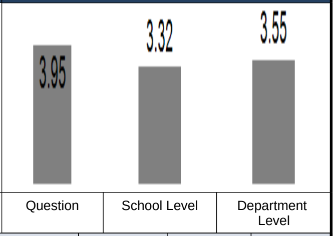
**7 - Compared to other Hopkins courses at this level, the workload for this course is:**

| Response Option | Weight | Frequency | Percentage |
|---|---|---|---|
| Much lighter | (1) | 0 | 0% |
| Somewhat lighter | (2) | 0 | 0% |
| Typical | (3) | 4 | 20% |
| Somewhat heavier | (4) | 13 | 65% |
| Much heavier | (5) | 3 | 15% |
| N/A | (0) | 0 | 0% |

Percent Responses



Means



| Return Rate | Mean | STD | Median | School Level | Mean | STD | Median | Department Level | Mean | STD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 20/25 (80%) | 3.95 | 0.60 | 4.00 | 7,697 | 3.32 | 1.01 | 3.00 | 989 | 3.55 | 1.05 | 4.00 |

**8 - What are the best aspects of this course?**

| Return Rate | 20/25 (80%) |
|---|---|

**- The course really does teach you how to program effectively, regardless of how you do on the assignments.**

**- This course gave a very nice overview to concepts in C and C++ through a lot of examples and projects. Going through and doing all of the projects helped me learn a lot.**

**- The instructor presents the material very clearly and manages to spend some individual time with each student during classes. We are also given ample time to work in class with the help of the professor and TAs which is quite useful on larger projects.**

**- The one to one TA's in every class.**

**- Being able to interact with the professor on a regular basis was amazing.**

**- This course introduces you to how data structures work and how to build them.**

**- Almost everything. The professor, TAs, and CAs were all very very helpful (and at times, inspiring). Everything was clear and they made sure we learned everything we needed to know.**

**- It's very thinking heavy, most of the work for the course is design**

**- This course is phenomenal. From start to finish, this class is something special.**

**Professor Amir has great vision for who we are as students, programmers, and innovators. He has no reservations with sharing that vision, and it is refreshingly inspirational.**

**There is a tremendous support network throughout the course. Almost every class meeting offers time to talk one-on-one with a TA, and they are available 24/7 through the course-help email. The longest I've ever waited for a reply was a few hours, and that was because I emailed in the middle of the night.**

**The course will teach you how to program. The first big project gets you into C pretty quickly, and from there, everything builds wonderfully in complexity. If you keep**

**Course:** EN.600.120.03.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

up, you can see the power of what is being taught and learn to apply it effectively.

There are tons of examples. Every week is met with new sample code, and all the code is stored online for viewing. If you have any syntax questions, in all likelihood, there is a piece of sample code that will tell you everything you need to know.

Grading is fair and transparent. Every project is returned with a detailed assessment and reasoning for why points were or were not awarded. A practice midterm is given to help prepare for the real midterm. This course is not meant to trick you, it is not meant to give out so many A's and B's, and it certainly is not meant to "weed out" anyone from programming/CS. It is meant to teach you how to code, and it is very successful at doing so.

- The best thing about this course is the personalized attention each student receives from both Professor Yair and the TAs/CAs. The course staff as a whole actively works with each student on a weekly basis, going over their project designs and code (often line-by-line) and helping them to understand how they're progressing and where they can improve. The TAs/CAs were absolutely phenomenal. I honestly can't say enough good things about them, but I think it's obvious that they care a lot about the course and about the students. Every TA I've spoken with has been happy to help and answer my questions, extremely knowledgeable, and very patient. They refuse to leave a student's side until they have cleared up all of the student's confusion, and they never seem burdened by tons of questions and tedious debugging sessions with our code. One TA in particular, Jeff DallaTezza, stuck out to me as an all-around awesome TA, and I really benefited from his help and advice on the projects, and enjoyed chatting with him about coding in general from time to time. This is a great group of course staff all around, and I can honestly say that they're the best course staff I've had so far in my three semesters at Hopkins.

Of course, I can't end this evaluation without speaking about Professor Yair and how inspirational he was as a professor and a mentor to the class. It's clear that Professor Yair loves working with students and cares about every individual student and their success in the course. His decision to meet with each of us individually to go over our final projects was extremely helpful, and I appreciate his time in doing so. He always puts in extra effort that is not normally given by professors, and this is what makes the difference in this class. He wants every student to be an excellent programmer after they finish the course, and I believe that he does everything he can to achieve this goal, and I think his goal has been successfully achieved this semester. I walked out of this class with so much more knowledge about how to write efficient, readable, and clean code, along with picking up two new useful languages and some Linux terminal experience, which I've always wanted. Finally, Professor Yair's last class meeting where he brought all three sections together to summarize the course was a great experience. Hearing Professor Yair talk is very inspirational, and he has encouraged me to look inside myself for what path in life will best serve my particular interests, while having a meaningful impact in the world. Professor Yair meets the criteria of a great mentor in every way, and I am happy to have taken his course.

With regard to having three separate sections of the course as opposed to placing all the students in a single time slot, I personally thought having separate sections was integral to the individualized attention that we received, which made the course so worthwhile. However, if this individualized attention could still be provided in a section with three times the students in the same hour-and-a-quarter period of time, then I think grouping all of the students into a single large section would suffice. To me, whatever can improve the individualized attention in this course is what's best for the students.

- Nice teachers and TA's are always willing to help.

- Projects push you to be a better programmer and to think about what you're doing. Yair, Amy, Jeff & TAs were all very helpful and encouraging. Definitely feel like I have a good conceptual and practical understanding of what was discussed in the course.

- Feed back on unfinished code was extremely helpful, and I learned the most from a TA explaining a problem in my code that I didn't understand. If this class is changed to be 1 big giant lecture than the friday sections were you could ask why about your code needs to stay.

- The 30 person sections were extremely helpful for my understanding of the material and getting to speak 1 on 1 with the TA's was invaluable.

- This course has small sections so that class periods can be used for both lecture and one-on-one help and guidance on assignments. The course is fast-paced but comprehensive, and the assignments cover programs that are difficult conceptually but simple to construct using what we've learned in class.

- Getting lots of feedback in class

- 1. Feedback for projects was very useful.

2. Professor, Amy and the TA really took in the effort to manage the course. They were very helpful whenever you had questions with your projects.

3. The projects were very intense, but they made you learn how to think in programming terms. The class overall was very stressful but I learned a lot.

- Learning how to program more effectively.

- -Excellent focus on programming syntax and efficiency
-Feedback for programs is great

- You are really challenged to think.

**Course:** EN.600.120.03.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

**9 - What are the worst aspects of this course?**

| | |
|---|---|
| Return Rate | 17/25 (68%) |

- It's a bit work-heavy and it's in C/C++. The first half of the class is very easy and then the second half is difficult.

- The first two weeks are a steep learning curve, especially if you don't already know C/C++

- I understand that some topics are inherently boring in a programming class, but on a few days it was hard to stay focused and pay attention to the lectures. This wasn't a big problem, however, and it was alleviated by print-outs of the code being provided to us at the start of each class to help us follow along.

- Debugging and checking for memory leaks.

- -Since there are four projects in this course, a bit of time needs to be put into each one.
-The midterm felt out of place (I'm not sure if speed coding is a skill that's needed for this course)
-C++ memory errors are difficult to fix

- none that I can list.

- Sometimes lecture material was too fast-paced and used terminology that was not at 100 level, but this seems a simple problem to fix.

- 1. I did not enjoy the lecture formats. Prof. Amir went over the programs too fast. He assumed you know a lot about programming already. The lectures themselves weren't very useful for me.

2. For the first project, a lot of students including myself weren't very sure on how he graded the projects. I lost a lot of points for not meeting the standards. However, I think I could have done much better if I had a more clear set of guidelines.

3. I did not like the midterm portion of the course. I don't think the midterm really tests you on how good a programmer you are. You would lose lots of points for very simple mistakes that caused the program to crash.

4. Professor seemed somewhat impatient when you ask him questions that he has already talked about in class.

- It's a lot of work

- None.

- The workload in this course is a bit unbalanced. Some projects (1, 3, final) take seemingly endless amounts of time. Other projects (2, 4) do not take nearly as long. Between projects, there is little work to be done other than keep up with what is going on in class. The result is that you go through periods where you spend every waking hour writing code, and you go through other periods where intermediate programming is not even on your to-do list. I wish the workload could be more balancec throughout the semester. Really, though, this is a small complaint and does not detract from the quality of this course.

- Sometimes just diving right into example code was confusing without a more general overview of certain concepts first.

- Grading criteria are not very clear. While I always know what my project is supposed to do, more subtle things such as memory management, degree of commenting, and breakability are ambiguous.

- It takes a few weeks to get going, as the groundwork has to be laid in lecture. These lectures are a bit dry, but they start making sense when you're actually applying them with projects the projects. Also vim is still kind of confusing.

- Sometimes, the information for the course went by too fast and made it difficult to integrate into the programs we were writing.

- The workload and the fact that you have to learn more than what is taught during class to complete the projects.

- The midterm was very hard.

**Course:** EN.600.120.03.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

**10 - What would most improve this class?**

| Return Rate | 19/25 (76%) |
|---|---|

- Nothing

- I honestly can't think of anything

- More variation in the type of assignments.

- The only thing I would suggest is providing a little more guidance on the expectations for design documents for the projects. I found myself writing very verbose designs, and I think it was because I wasn't sure what exactly to include, so I just included everything I could think of in order to be on the safe side (pseduocode, detailed explanations on even minute things, etc.). I think if a bit more guidance could be provided on how to write a concise yet effective design doc, this would clear up the ambiguity.

- Give Yair a break!

- -NA (I felt that the course was great and I have no suggestions on how to improve it)

- Maybe if the professor didn't do 3 classes in a row since he was grumpy by the third (my) section. Although I understand that it's really stressful to do 3 classes in a row.

- Having a brief introduction to concepts (even just providing definitions or a quick blurb about how it is used) before going right into example code could be useful.

- Making the project descriptions a bit more clear would help quite a lot for the reasons above.

- If they get to keep it.

- More short in-class ungraded assignments was help correct misunderstanding I had with new material before I started the next project.

- Keep the 30 person lecture sizes.  Doing it as a 90 person lecture without the active programming during lecture would make the quality of the course drop immensely.

- More small assignments throughout the semester rather than only a few big assignments would help keep students on track of learning new material.

- Having more in-depth instructions with the specific parts of the languages C and C++

- 1. Improve the lecture format in this class by going slower and in more detail.

2. Make a clearer set of grading instructions and project guidelines on the class webpage instead of sending us tons of email all the time.

3. Cancel the midterm for this exam.

- I think this class is fine as it is.

- A very small first project, where people can get familiar with the tools (vim and the compiler) and the basics of C.  Maybe not even a graded, but just a set of directions to follow to learn how to use vim functions (visual mode, split screen, copy/paste, searching) while you put together a very simple program. Explaining vim functions anc basic C concepts is not nearly as effective as using them, so I think something interactive would help.  Otherwise the class was very effective.

- A bit more in-depth explanation of certain programs covered in lecture, with a little less use of programming terms that students just out of Intro to Programming would not understand (i.e, a friend in the course was still not used to the meaning of 'instantiation', but again this is a simple issue to fix).

- Each program should be divided up into smaller parts and checked more frequently because otherwise its too overwhelming.

**Course:** EN.600.120.03.FA13 : Intermediate Programming

**Instructor:** Yair Amir *

**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.)**

| | |
|---|---|
| Return Rate | 18/25 (72%) |

- Know basic programming and expect to think about problems in ways you haven't before

- They have to want to work on the projects to learn.

- The grading for this course is rough, but its worth it

- You should be familiar with programming before coming into this course.  The topics are similar to that of Intro, but in languages you may not have worked with before and with a deeper level of complexity.  While the professor stops a few times to make sure everyone is OK with the content being covered, if you aren't at least somewhat familiar with the content ahead of time, you will get lost.  The course moves at a quick pace and two languages are covered in a single semester, so you have to be aware of what's going on.  Also, be prepared to spend a lot of time writing code, probably at least 10 hours for each project.

- Focused entirely on projects, which should be started ahead of time.  Use class time well to ask questions. Don't be afraid to bother the TAs, they're super helpful.

- This course is very hands on and quick paced, but extremely rewarding in terms of the results. Professor Amir states that the goal of this course is for students to never worry or question their programming skills again, and that should they see a program design, they would know how to build it, and by the end of the course I felt that this was true.

- It can be a Lot of work.  However grading is very fair and you learn a lot.

- Don't take it with other heave-workload classes

- You need to have a good grasp on some sort of programming language. This is not an introductory course to programming. Also, the projects take a long time to complete around the end of the semester. This is also a bit harder than it's 100-level suggests.

- This course requires a lot of time for projects, but the grading is fair and the projects are very useful. You will learn a lot in this class.

- Make the most of your time in class with the professor and TAs to work on your code. If you get stuck, just email the hotline and you will get a quick response. Besides that just avoid procrastinating and you will be fine.

- Definitely have a background in basic programming before coming to this class. A lot of the basics will be rehashed in the first week (variable declarations, loops, functions, etc.), but be sure to review some intro programming concepts before the first class. Most importantly: Be prepared to put A LOT of time into this course. Because I loved this course, I didn't mind the time I had to spend on the projects, but you should understand that the projects will require a lot of thinking and time spent on design, writing code, and debugging. The course staff are readily available to help you with debugging and getting over humps via email, but the projects are time-consuming. Do not leave any project to the last few days. Also, you may first think that writing a detailed design document before beginning to write code for each project is just a waste of your time, but it actually does help quite a bit, especially when the logic required for the projects starts to increase in complexity (around Project 3). Focus intently on writing a near-perfect design so that when you go to actually write the code, it will be much more streamlined and easier to do. The course staff will review your code line-by-line, which is very helpful, but this means that no sloppy code is acceptable. Be sure to write clean, organized code the way they tell you to do it. Finally, follow the instructions for the projects to every detail! I overlooked an instruction in the final project that could have created a big problem during grading if it didn't turn out to be so easily fixed. Definitely talk to your TAs/CAs or the professor if you're unclear on the instructions -- the course staff are all extremely helpful and willing to sit with you until you know what you're doing, so leverage the opportunity to get help when you need it!

Overall, this is an awesome course, and if you care about becoming a better programmer, then there is no better course than Professor Yair's Intermediate Programming class. Work hard, give it your best, and you'll get so much out of it.

- It is a heavier workload.

- It will be a very time-consuming class.

- Be prepared to work.  But Yair is an incredible teacher.

- -When stuck on a problem, ask for help after trying for +5 minutes.
-Focus on design before starting to code (Reduces chance of errors)
-Stay positive! The course will be tough at first, but at the end your programming techniques will be greatly improved.
-The TAs for this course are very dedicated, do ask for help!!!

- 1. Do very well in Intro to Java or equivalent.

2. The time frame for projects is usually around 2 weeks. Start early. Don't procrastinate.

- Having a solid background in programming will help a lot.