

JHU - Krieger School of Arts & Sciences / Whiting School of Engineering  
 ASEN.2012.Spring

Course: EN.600.120.01.SP12 : Intermediate Programming

Instructor: Yair Amir \*

1 - The overall quality of this course is:												
Response Option	Weight	Frequency	Percentage	Percent Responses			Means					
Poor	(1)	0	0.00%			4.47	4.01	4.09				
Weak	(2)	0	0.00%									
Satisfactory	(3)	2	6.25%									
Good	(4)	13	40.63%									
Excellent	(5)	17	53.13%									
N/A	(0)	0	0.00%									
				0	25	50	75	100	Question	School Level	Department Level	
Return Rate		Mean	STD	School Level			Mean	STD	Department Level		Mean	STD
32/32 (100%)		4.47	0.62	6,964			4.01	0.94	750		4.09	0.91

2 - The instructor's teaching effectiveness is: Yair Amir												
Response Option	Weight	Frequency	Percentage	Percent Responses			Means					
Poor	(1)	0	0.00%			4.16	3.99	3.98				
Weak	(2)	0	0.00%									
Satisfactory	(3)	6	18.75%									
Good	(4)	15	46.88%									
Excellent	(5)	11	34.38%									
N/A	(0)	0	0.00%									
				0	25	50	75	100	Question	School Level	Department Level	
Return Rate		Mean	STD	School Level			Mean	STD	Department Level		Mean	STD
32/32 (100%)		4.16	0.72	7,770			3.99	1.03	760		3.98	0.98

3 - The intellectual challenge of this course is:												
Response Option	Weight	Frequency	Percentage	Percent Responses			Means					
Poor	(1)	0	0.00%			4.69	4.07	4.16				
Weak	(2)	0	0.00%									
Satisfactory	(3)	1	3.13%									
Good	(4)	8	25.00%									
Excellent	(5)	23	71.88%									
N/A	(0)	0	0.00%									
				0	25	50	75	100	Question	School Level	Department Level	
Return Rate		Mean	STD	School Level			Mean	STD	Department Level		Mean	STD
32/32 (100%)		4.69	0.54	6,911			4.07	0.90	747		4.16	0.84

JHU - Krieger School of Arts & Sciences / Whiting School of Engineering  
 ASEN.2012.Spring

Course: EN.600.120.01.SP12 : Intermediate Programming

Instructor: Yair Amir \*

4 - The teaching assistant for this course is:											
Response Option	Weight	Frequency	Percentage	Percent Responses			Means				
Poor	(1)	0	0.00%			4.75	4.05	4.08			
Weak	(2)	0	0.00%								
Satisfactory	(3)	2	6.25%								
Good	(4)	4	12.50%								
Excellent	(5)	26	81.25%								
N/A	(0)	0	0.00%								
				0	25	50	75	100	Question	School Level	Department Level
<b>Return Rate</b>	<b>Mean</b>	<b>STD</b>		<b>School Level</b>	<b>Mean</b>	<b>STD</b>	<b>Department Level</b>	<b>Mean</b>	<b>STD</b>		
32/32 (100%)	4.75	0.57		6,923	4.05	1.05	745	4.08	1.03		

# JHU - Krieger School of Arts & Sciences / Whiting School of Engineering

## ASEN.2012.Spring

**Course:** EN.600.120.01.SP12 : Intermediate Programming

**Instructor:** Yair Amir \*

5 - Please enter the name of the TA you evaluated in question 4:	
Return Rate	28/32 (87.5%)
<ul style="list-style-type: none"><li>- Tom Tantillo</li><li>- There were many, and not all were official TAs, but Daniel Obenshain is/was great</li><li>- There are many</li><li>- Tom Tantillo</li><li>- there are so many TAs</li><li>- Tom Tantillo, Daniel Obenshain</li><li>- Tom and Dano</li><li>- Tom Tantinillo, Dan Obenshain, Justin</li><li>- Dan O, Tom Tantillo</li><li>- Tom and Dano</li><li>- Dano/Tom</li><li>- Daniel and Tom</li><li>- tom , dano</li><li>- Daniel Obenshain and Tom Tantillo</li><li>- Daniel Obenshain</li><li>- Daniel Obenshain and Tom Tantillo</li><li>- Daniel An</li><li>- Tom Tantillo, Dan Obershein</li><li>- Tommy T</li><li>- Daniel Obenshain, Tom Tantillo</li><li>- Dano, Tom</li><li>- Daniel Obenshain, Tom Tantillo</li><li>- Tom Tantillo, Daniel Obenshain</li><li>- All of them, there are a ton.</li><li>- Dan Obenshain and Tom Tantillo</li><li>- Dano, Tom, Xinyuan, Steve, Amy, and this other CA whose name I cant remember</li><li>- All/not applicable</li><li>- dan and tom</li></ul>	

JHU - Krieger School of Arts & Sciences / Whiting School of Engineering  
 ASEN.2012.Spring

Course: EN.600.120.01.SP12 : Intermediate Programming

Instructor: Yair Amir \*

6 - Feedback on my work for this course is useful:											
Response Option	Weight	Frequency	Percentage	Percent Responses			Means				
Disagree strongly	(1)	0	0.00%		4.45	3.82	3.91	Question	School Level	Department Level	
Disagree somewhat	(2)	0	0.00%								
Neither agree nor disagree	(3)	2	6.45%								
Agree somewhat	(4)	13	41.94%								
Agree strongly	(5)	16	51.61%								
N/A	(0)	0	0.00%								
				0	25	50	75	100			
<b>Return Rate</b>	<b>Mean</b>	<b>STD</b>		<b>School Level</b>	<b>Mean</b>	<b>STD</b>	<b>Department Level</b>	<b>Mean</b>	<b>STD</b>		
31/32 (96.88%)	4.45	0.62		6,919	3.82	1.05	749	3.91	1.02		

7 - Compared to other Hopkins courses at this level, the workload for this course is:											
Response Option	Weight	Frequency	Percentage	Percent Responses			Means				
Much lighter	(1)	0	0.00%		4.09	3.34	3.41	Question	School Level	Department Level	
Somewhat lighter	(2)	0	0.00%								
Typical	(3)	3	9.38%								
Somewhat heavier	(4)	23	71.88%								
Much heavier	(5)	6	18.75%								
N/A	(0)	0	0.00%								
				0	25	50	75	100			
<b>Return Rate</b>	<b>Mean</b>	<b>STD</b>		<b>School Level</b>	<b>Mean</b>	<b>STD</b>	<b>Department Level</b>	<b>Mean</b>	<b>STD</b>		
32/32 (100%)	4.09	0.53		6,939	3.34	1.00	752	3.41	1.04		

# JHU - Krieger School of Arts & Sciences / Whiting School of Engineering

## ASEN.2012.Spring

**Course:** EN.600.120.01.SP12 : Intermediate Programming

**Instructor:** Yair Amir \*

8 - What are the best aspects of this course?	
Return Rate	26/32 (81.25%)
<ul style="list-style-type: none"> <li>- There is a lot of opportunity for one on one meetings.</li> <li>- It is very challenging and you will learn in a relatively short period of time to be a very proficient C programmer.</li> <li>- Doing the projects correctly will teach you programming.</li> <li>- The TAs were incredibly helpful, and devoting class time to help with the homework really made a difference. I could expect answers to emails quickly, and I never felt like I didn't have the resources to understand class material/homework.</li> <li>- The TAs have helped me an incredible amount this semester. I feel like the bulk of my learning has happened during working on projects, especially in the lab. They are very good at answering my questions and also making sure that I understand their answers as well.</li> <li>- Lots of hands-on programming and resources for help</li> <li>- The lectures have good pace and are interesting. There is a lot of access to the professor and tas. A lot of work goes into grading, so feedback is valuable. The people involved are very willing to help, so you can get as much out of this course as you are willing to put in. The data structures used in assignments are interesting.</li> <li>- The TAs are fantastic. Yair clearly cares about the material and his students. I feel like I have learned a tremendous amount</li> <li>- I love how we were really challenged with difficult programs.</li> <li>- Projects are comprehensive; enough time is allocated to them. Project topics could be more interesting. The garage and deli tie-ins for the final projects were interesting, but the first projects were bland.</li> <li>- helps us understand prngng how to program the right way</li> <li>- It was easy to tell that everyone involved really cared about making sure that the students learned. The projects were difficult without being unreasonable and did a good job of teaching the materials.</li> <li>- Lots of help opportunities.</li> <li>- You come out of the course with enough experience under your belt to feel confident in your programming skills. The courses teaches you to think like a programmer, which, alongside the more technical skills, is really what you take out of the course.</li> <li>- The hand on programming projects make sure that you learn the material well. the TAs are wonderful and they are very helpful and resourceful</li> <li>- Gives a thorough example of the new things to learn and there is ample time to work on programs.</li> <li>- The TAs and CAs were extremely helpful, and genuinely dedicated to helping students succeed.</li> <li>- The tutorials on Friday (interactive)</li> <li>- Very hands on</li> <li>- Learning a new computing language, having lots of help from TA's who really knew what they were doing and who were very dedicated to helping out the students</li> <li>- The teaching staff is really good. Theres lots of time for developing in class alongside TAs and the professor. Good amounts of hands on learning.</li> <li>- The TAs and Yair are incredibly good at helping people personally, and talking to you one on one and helping you with your code. This was by far my favorite aspect of the course.</li> <li>- Only a handful of projects.</li> <li>- got a lot of practice, enough to learn programming but not too much workload</li> <li>- -focused, one-on-one help from TAs and professor for programs</li> <li>-valuable feedback on submitted projects</li> <li>-intellectually challenging programming assignments</li> <li>- Close instruction</li> </ul>	

# JHU - Krieger School of Arts & Sciences / Whiting School of Engineering

## ASEN.2012.Spring

Course: EN.600.120.01.SP12 : Intermediate Programming

Instructor: Yair Amir \*

9 - What are the worst aspects of this course?	
Return Rate	22/32 (68.75%)
<ul style="list-style-type: none"><li>- There honestly isn't enough time given to C++ given the complicated and not entirely intuitive nature of the language. I think a whole semester of C and a separate C++ course would be better.</li><li>- Project deadlines make doing other work difficult if you're taking over 16 credits.</li><li>- There was one particular week I felt overwhelmed b/c we were working on project 4 while going over material for the final project and expected to do mini-assignment on class. I felt like I couldn't devote all my time to understanding project 4 and was rushed into understanding things that required the background for project 4.</li><li>- The projects are very long, and I have often worked over 40 hours on them. Also, the teaching style of just printing out programs that are already written and going over them does not really work for me.</li><li>- Reading source-code. I find this an ineffective approach to learning programming. I would suggest to do actual code on the screen, and ask students what needs to be done next and have students engaged.</li><li>- The assignments regarding OOP can be a bit vague. The course focuses on more procedural parts of programming, and the functional benefits of object oriented concepts, but the assignments ask students to create object oriented designs that require concepts that seem removed from the goals of the course.</li><li>- The lectures can be boring</li><li>- Sample code is available, but could be better commented upon.</li><li>- You underestimate the difficulty of the assignments.</li><li>- A lot of expected out of the students in this course, especially considering the course level. This is a good thing for the most part since you do learn a lot, but it also means that certain things are left unexplained, which can be confusing at times. The assignments were often defined very vaguely as well, which can be very frustrating.</li><li>- the assignments cannot be done quickly. If you know the material well, the assignments will be easier, but they are not easy to complete quickly.</li><li>- Large workload from other classes can make the possibility of finishing a programming project almost impossible.</li><li>- The assignments were long and difficult for novice programmers.</li><li>- When I don't understand what's going on in class</li><li>- Lots of information is presented at once.</li><li>- Sometimes there is a gap of understanding between instructor and student. I know the instructor is knowledgeable about his topic, but sometimes in class when I would ask questions, he didn't seem to be able to fully understand my difficulties. Also, unlike Intro to Programming in Java, this course goes very fast and we learn coding techniques not by coding ourselves, but by reading code, and I feel like that is not the most efficient way for us to learn how to code. I sometimes felt a lack of communication between instructor and student concerning how our projects should be--some projects were worded in such an open way that I would think that I had accomplished all of the requirements and I would get feedback on how I did not implement something that was not even specified in the requirements.</li><li>- The material is hard and the projects are long. One needs to manage his or her time well.</li><li>- Having to use VIM. I feel like there had to be a better way to do it.</li></ul> <p>Also, I found that all the projects were huge and really similar. I felt more like I was taking a class about learning to use C and C++ than a class where I learned much new about programming.</p> <ul style="list-style-type: none"><li>- Each project is fairly rigorous.</li><li>- we don't learn as much as programming languages themselves. aka. I was asked during an interview what's the difference between a struct and a class, I wish we talked about these more in class</li><li>- lectures are not very helpful and take up more of the course than they should</li><li>- Difficulty</li></ul>	

# JHU - Krieger School of Arts & Sciences / Whiting School of Engineering

## ASEN.2012.Spring

Course: EN.600.120.01.SP12 : Intermediate Programming

Instructor: Yair Amir \*

10 - What would most improve this class?	
Return Rate	20/32 (62.5%)
<ul style="list-style-type: none"><li>- nothing, pretty great as is, with the exception of maybe extending it to be entirely C focused.</li><li>- Have more, smaller-scale projects and less giant ones. Also, the midterms are ridiculous and do not show whether you can program or not. Just because you can't write a program in an hour doesn't mean you can't do it at all.</li><li>- I don't have a lot of suggestions. This is a well executed course.</li><li>- I think that coding the programs instead of reading them from a printout or on the screen would be useful (like writing them as a class). I know that this would decrease the number of programs we could go over, so maybe only do this for a few key examples. Also, posting solutions for the mini projects or just the normal projects would be nice so that we have examples to base our future work off of.</li><li>- Having smaller coding projects. Also, some students have never learned data structures, so I don't think there should be data structures in the homeworks (there is a separate course for this, called "data structures")</li><li>- A bit more structure on project 4 and part two of the final project.</li><li>- The class would be improved if the lectures were more interactive.</li><li>- Maybe having it in a bigger classroom so more students could take the class</li><li>- As mentioned, the projects could be more interesting. Maybe using a hash function to manage race cars on a race track rather than roast beef in a deli, or golfers on a course rather than cars in a garage. In fact, the golf course could be very polymorphic.</li><li>- More frequent, forced feedback?</li><li>- More clearly articulated assignments. There were many times when I felt like I didn't know what I was supposed to be doing or when I felt like I had done something entirely different from how others had done it.</li><li>- more time for tutorial and maybe less lecture. I learn a lot through actually doing the assignments, though lectures are needed for a foundation in writing the code.</li><li>- Seems fine?</li><li>- Nothing really</li><li>- The professor recently came from teaching Datastructures, as I understand, and it feels like he is still trying to teach datastructures. It'd be much more helpful if we didn't do such complex data structures (ie: nothing more complex than a linked list) and did many more smaller programs.</li><li>- More hands-on learning rather than just reading code to learn the process of coding, less focus on data structures and data management in projects (or keep the focus since it is valuable but spend more time teaching it), more specificity in project requirements</li><li>- perhaps a programming lab 1 day a week for a few hours.</li><li>- Maybe more varied projects. Or more frequent, smaller programming assignments.</li></ul> <p>Also, I feel like I only really learned how to use pointers from this class. That and the languages, but syntax isn't too difficult. I feel like I'm a better programmer by virtue of practicing, but I don't think I should need a class to force me to practice.</p> <ul style="list-style-type: none"><li>- see above</li><li>- shorten the lectures and add more programming time</li></ul>	

# JHU - Krieger School of Arts & Sciences / Whiting School of Engineering

## ASEN.2012.Spring

**Course:** EN.600.120.01.SP12 : 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

22/32 (68.75%)

- Having already taken Data Structures could be a plus. Either way, whichever one you take first will be pretty difficult but will help a lot with the other.
- Start assignments on time, also learn to code incrementally and test things with driver programs.
- Know some C to begin with. It will help lower the learning curve significantly. Try taking Data Structures FIRST if you have taken AP Comp Sci
  - You shouldn't be too concerned when enrolling for this course as long as you are willing to put in the effort and time required to do the assignments.
  - The workload is heavy, but if you put in enough effort you will get a lot out of the course.
  - Take data structures first, and then this course.
  - There is a decent workload but a lot that can be gained from this course.
  - Some projects are difficult but help is always available. Seek help early on with the project. Yair and the TAs are fantastic
  - This course will take a lot of your time, but in the end you will understand programming a lot better
  - This course requires a lot of abstract thought. Regarding the questions below, I did not think the purpose of the course was to improve team-skills or deepen my understanding of the social impact of computing. There are other courses that do this, so no need for this one to do so.
  - The projects can take a large amount of time, try not to wait until the last minute.
  - Estimate how long each program will take you, and then triple it. That's how much time you need to spend.
  - It helps quite a bit if you are familiar with unix/linux environments as that's primarily what you will be working with. Also, start early on assignments (if you don't, you make get stuck with confusing bugs a day before the due date) and make a detailed design document (your design weighs heavily into the final grade of each exercise).
  - Make sure you take the time to code and debug. Try to get the programs done a couple days before and approach the TAs whenever you need help. they answer within the hour if you ask asking questions during the day.
  - Not easy, but very rewarding to go through it.
  - A solid foundation of basic Java is very helpful. Make sure to start the projects way in advance, you might encounter problems if you put them off until the day before they are due.
  - Make sure you DON'T procrastinate
  - It requires a lot of initiative.
  - The classload is heavy. Projects will take hours to complete and students should definitely start early on every project; never procrastinate. The plus side is that the class is very geared towards helping the student, and some classes are set aside for students to work on coding and to get help from numerous TA's.
  - As long you're willing to put time in, this class is very informative.
  - Some of the projects are going to take FOREVER. I spent 16 hours in a 24 hour period working on a project I was half way done with, and I still couldn't completely finish it.
  - The course has many personnel to assist you should you ever need help. And they should be utilized.
  - depending on who's teaching it the workload is very different



JHU - Krieger School of Arts & Sciences / Whiting School of Engineering  
 ASEN.2012.Spring

Course: EN.600.120.01.SP12 : Intermediate Programming

Instructor: Yair Amir \*

12 - This course improved my appreciation for and/or ability to engage in life-long learning.					
Response Option	Weight	Frequency	Percentage	Percent Responses	Means
Disagree strongly	(1)	0	0.00%		
Disagree somewhat	(2)	0	0.00%		
Neither agree nor disagree	(3)	1	3.23%		
Agree somewhat	(4)	16	51.61%		
Agree strongly	(5)	15	48.39%		
Return Rate				0 25 50 75 100	
31/32 (96.88%)					

13 - This course deepened my understanding of the social impact of computing.					
Response Option	Weight	Frequency	Percentage	Percent Responses	Means
Disagree strongly	(1)	0	0.00%		
Disagree somewhat	(2)	2	6.45%		
Neither agree nor disagree	(3)	9	29.03%		
Agree somewhat	(4)	14	45.16%		
Agree strongly	(5)	7	22.58%		
Return Rate				0 25 50 75 100	
31/32 (96.88%)					

14 - This course enhanced my ability to work effectively in a team.					
Response Option	Weight	Frequency	Percentage	Percent Responses	Means
Disagree strongly	(1)	6	19.35%		
Disagree somewhat	(2)	2	6.45%		
Neither agree nor disagree	(3)	14	45.16%		
Agree somewhat	(4)	7	22.58%		
Agree strongly	(5)	2	6.45%		
Return Rate				0 25 50 75 100	
31/32 (96.88%)					

**JHU - Krieger School of Arts & Sciences / Whiting School of Engineering**  
**ASEN.2012.Spring**

**Course:** EN.600.120.02.SP12 : Intermediate Programming

**Instructor:** Yair Amir \*

1 - The overall quality of this course is:												
Response Option	Weight	Frequency	Percentage	Percent Responses			Means					
Poor	(1)	0	0.00%			4.50	4.01	4.09				
Weak	(2)	0	0.00%									
Satisfactory	(3)	2	8.00%									
Good	(4)	8	32.00%									
Excellent	(5)	14	56.00%									
N/A	(0)	1	4.00%									
				0	25	50	75	100	Question	School Level	Department Level	
<b>Return Rate</b>		<b>Mean</b>	<b>STD</b>	<b>School Level</b>			<b>Mean</b>	<b>STD</b>	<b>Department Level</b>		<b>Mean</b>	<b>STD</b>
25/28 (89.29%)		4.50	0.66	6,964			4.01	0.94	750		4.09	0.91

2 - The instructor's teaching effectiveness is: Yair Amir												
Response Option	Weight	Frequency	Percentage	Percent Responses			Means					
Poor	(1)	0	0.00%			4.42	3.99	3.98				
Weak	(2)	0	0.00%									
Satisfactory	(3)	3	12.00%									
Good	(4)	8	32.00%									
Excellent	(5)	13	52.00%									
N/A	(0)	1	4.00%									
				0	25	50	75	100	Question	School Level	Department Level	
<b>Return Rate</b>		<b>Mean</b>	<b>STD</b>	<b>School Level</b>			<b>Mean</b>	<b>STD</b>	<b>Department Level</b>		<b>Mean</b>	<b>STD</b>
25/28 (89.29%)		4.42	0.72	7,770			3.99	1.03	760		3.98	0.98

3 - The intellectual challenge of this course is:												
Response Option	Weight	Frequency	Percentage	Percent Responses			Means					
Poor	(1)	0	0.00%			4.62	4.07	4.16				
Weak	(2)	0	0.00%									
Satisfactory	(3)	0	0.00%									
Good	(4)	9	36.00%									
Excellent	(5)	15	60.00%									
N/A	(0)	1	4.00%									
				0	25	50	75	100	Question	School Level	Department Level	
<b>Return Rate</b>		<b>Mean</b>	<b>STD</b>	<b>School Level</b>			<b>Mean</b>	<b>STD</b>	<b>Department Level</b>		<b>Mean</b>	<b>STD</b>
25/28 (89.29%)		4.62	0.49	6,911			4.07	0.90	747		4.16	0.84

**JHU - Krieger School of Arts & Sciences / Whiting School of Engineering**  
**ASEN.2012.Spring**

**Course:** EN.600.120.02.SP12 : Intermediate Programming

**Instructor:** Yair Amir \*

4 - The teaching assistant for this course is:												
Response Option	Weight	Frequency	Percentage	Percent Responses			Means					
Poor	(1)	0	0.00%			4.76	4.05	4.08				
Weak	(2)	0	0.00%									
Satisfactory	(3)	0	0.00%									
Good	(4)	5	20.00%									
Excellent	(5)	16	64.00%									
N/A	(0)	4	16.00%									
				0	25	50	75	100	Question	School Level	Department Level	
<b>Return Rate</b>		<b>Mean</b>	<b>STD</b>	<b>School Level</b>			<b>Mean</b>	<b>STD</b>	<b>Department Level</b>		<b>Mean</b>	<b>STD</b>
25/28 (89.29%)		4.76	0.44	6,923			4.05	1.05	745		4.08	1.03

5 - Please enter the name of the TA you evaluated in question 4:	
<b>Return Rate</b>	15/28 (53.57%)
<ul style="list-style-type: none"> <li>- Daniel Obenshian, Tom Tantillo</li> <li>- Tom and Dano</li> <li>- Dan Obenshain, Tom Tantillo</li> <li>- Tom Tantillo, Daniel Obenshain</li> <li>- Daniel Obenshain and Tom Tantillo</li> <li>- Daniel, Tom, Amy and the rest</li> <li>- Daniel, Tom</li> <li>- All of the TA's for the course.</li> <li>- Daniel Obenshain</li> <li>- All of them</li> <li>- Dan and Tom</li> <li>- N/A</li> <li>- Dano and Tom the best</li> <li>- Amy, Tom, Dano, Ben, all the rest of the people who helped out as well</li> <li>- Daniel, Tom</li> </ul>	

JHU - Krieger School of Arts & Sciences / Whiting School of Engineering  
 ASEN.2012.Spring

Course: EN.600.120.02.SP12 : Intermediate Programming

Instructor: Yair Amir \*

6 - Feedback on my work for this course is useful:											
Response Option	Weight	Frequency	Percentage	Percent Responses			Means				
Disagree strongly	(1)	0	0.00%			4.75	3.82	3.91			
Disagree somewhat	(2)	0	0.00%								
Neither agree nor disagree	(3)	0	0.00%								
Agree somewhat	(4)	6	24.00%								
Agree strongly	(5)	18	72.00%								
N/A	(0)	1	4.00%								
				0	25	50	75	100	Question	School Level	Department Level
Return Rate	Mean	STD		School Level	Mean	STD	Department Level	Mean	STD		
25/28 (89.29%)	4.75	0.44		6,919	3.82	1.05	749	3.91	1.02		

7 - Compared to other Hopkins courses at this level, the workload for this course is:											
Response Option	Weight	Frequency	Percentage	Percent Responses			Means				
Much lighter	(1)	0	0.00%			4.21	3.34	3.41			
Somewhat lighter	(2)	0	0.00%								
Typical	(3)	5	20.00%								
Somewhat heavier	(4)	9	36.00%								
Much heavier	(5)	10	40.00%								
N/A	(0)	1	4.00%								
				0	25	50	75	100	Question	School Level	Department Level
Return Rate	Mean	STD		School Level	Mean	STD	Department Level	Mean	STD		
25/28 (89.29%)	4.21	0.78		6,939	3.34	1.00	752	3.41	1.04		

**JHU - Krieger School of Arts & Sciences / Whiting School of Engineering**  
**ASEN.2012.Spring**

**Course:** EN.600.120.02.SP12 : Intermediate Programming

**Instructor:** Yair Amir \*

8 - What are the best aspects of this course?	
Return Rate	13/28 (46.43%)
<ul style="list-style-type: none"> <li>- Great professor, they all really care that we learn the material, everyone is accessible</li> <li>- The TA's and how talented they are. I don't know if I would have done half as well on the assignments without the tas' help. Also, the ideology behind the class concerning getting help with code and how code is graded. The fact that code is graded based on logic instead of compilation is very attractive and means that the class is serious about turning out good programmers.</li> <li>- Challenging and fun. We are taught C and C++ in a way that is not just a rehash of "Intro to Programming: Java", but that is a continuation of our programming development through larger scale projects.</li> <li>- The programming assignments, though time consuming, were really not very complicated. I was afraid to take this class because I've heard previous classes have had to program a Battleship-playing program that detects cheating and tries to cheat if it can get away with it...luckily, we just had to make a deli that sold products or a car shop that takes in cars with lists of tasks associated with them.</li> <li>- I think the best aspect was all the support provided by the TAs and the professor who helped us on every step along the way</li> <li>- One really learns to program. All of the projects are challenging yet intriguing. I would say that this course has brought me to another level in terms of my programming abilities.</li> <li>- The individual attention each student gets.</li> <li>- Instructor's teach-by-example style lectures            Outstanding amount of individual attention            Friday tutorial sessions allow access to one-on-one help            Challenging projects require application and extension of concepts from class            Professor cares about each student's success</li> <li>- All</li> <li>- This was an overall really great course. Although there is a lot of work, Dr. Amir is very very concerned that everyone learn how to program and develop good programming habits. Both Dr. Amir and the TAs put in a considerable amount of time and effort to help you learn the material and complete programming assignments. The TAs were very helpful with explaining the assignments and responding almost immediately to emails. The assignments were also graded line by line and so the TAs gave very good feedback about how to improve your code. I thought that this was particularly important. Dr. Amir was also very enthusiastic about his own work in CS and inspired you work hard towards writing good code. The concepts can get complex, but overall they were explained very clearly. Excellent course.</li> <li>- Yair is a great professor great TAs</li> <li>- This was without a doubt, the best course I have ever taken at Hopkins. I have never had a professor who invested so much of their own time into the success of each and every student in the course. Yair and the course assistants treated each and every student with unbelievable amounts of respect and their efforts always motivated me to keep pushing and keep trying my best. This was an environment which really brought out everyone's best work. Yair and the assistants dissected each and every line of code. I have never learned a language better than C and C++. The feedback on projects was so individually focused that it actually difficult to make the same mistake twice in this course. I really feel like I cannot express how awesome this course was. I was skeptical of Dr. Amir's goals at the beginning of the semester, but I truly can say that I could program anything in C/C++ on one machine on my own. I've got a skill I won't forget.</li> <li>- The lab sessions and exercises</li> </ul>	

# JHU - Krieger School of Arts & Sciences / Whiting School of Engineering

## ASEN.2012.Spring

**Course:** EN.600.120.02.SP12 : Intermediate Programming

**Instructor:** Yair Amir \*

9 - What are the worst aspects of this course?	
Return Rate	10/28 (35.71%)
<ul style="list-style-type: none"> <li>- It was very hard, and I had very little time to turn out designs assigned on Wednesday and expected Friday, but that might just be a personal issue.</li> <li>- The focus on larger projects, which makes perfect sense for an intermediate level course, does logically come with the trade-off that the new syntax-based material may not be as well ingrained into our minds because we don't use it as immediately and frequently as we did for new material in the intro course.</li> <li>- Oftentimes, my program was way off what it should have been, yet none of the TA's that saw my code managed to figure out there was a problem until after I submitted. I had a totally unnecessary Customer class in my deli, all because the first TA who talked to me on the day we brought our design documents told me I would need it. As was submitting, I found out that the customer class was redundant since every customer is associated with a cart, but by this point I would have to change numerous parts of my code to accomodate the change. TA's were great for debugging, but otherwise their advice tended to set me back as often as it helped me out.</li> <li>- I think sometimes, all the help we were provided was counterproductive too because it made us somewhat dependent. I don't think I would have been able to do my assignments without any help at all.</li> <li>- Sometimes assignments are not described fully and require clarification. Lectures move through material very quickly; can be difficult to follow along</li> <li>- None</li> <li>- The midterm requires that you write a complete program in the span of one hour and twenty minutes. If you don't program as fast as others, there is a good chance that you won't finish and the exam will hurt your grade significantly. It takes some time just to understand what is being asked of you.</li> <li>- Yair yells, but is nice in long run</li> <li>- The fact that it is over. :(</li> <li>- lectures can be more related to the exercises</li> </ul>	
10 - What would most improve this class?	
Return Rate	12/28 (42.86%)
<ul style="list-style-type: none"> <li>- Curving the grades so the students are less concerned about grading and more concerned about becoming better programmers. The people who really want to do well will, and the people who want to do well and struggle have a safety net.</li> <li>- I was given some useful tips for improvement at the very end of the course. This kind of personal feedback would have been very helpful to have had earlier in the course, like what balance I should aim for between code simplicity, brevity, elegance, and clarity when I need to choose between functionally identical approaches.</li> <li>Perhaps the sample programs which we are given in our packets should have clearer titles than "fig15+14.cpp".</li> <li>- Group assignments can be a lot more fun and take the pressure off of big programming assignments. This class never assigned anything to work on in groups.</li> <li>- I would suggest teaching the students in slower, simpler steps and then building on the complexity, instead of giving us really huge and complex assignments which most of us wouldn't be able to solve on our own.</li> <li>- Having group projects would probably be a nice addition to the learning. I usually find that I will learn better in a group.</li> <li>- Being earlier in the day.</li> <li>- More hands-on practice writing short programs during lecture.</li> <li>- Nothing</li> <li>- The directions for some of the assignments were a bit unclear and that occasionally led to confusion, or implementing something the wrong way. This was a small problem.</li> <li>- Funnier programs</li> <li>- Nothing...this class is quite literally the ideal perfect class.</li> <li>- quality of lecture</li> </ul>	

**JHU - Krieger School of Arts & Sciences / Whiting School of Engineering**  
**ASEN.2012.Spring**

**Course:** EN.600.120.02.SP12 : 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	13/28 (46.43%)
<ul style="list-style-type: none"> <li>- Don't fall behind. GO TO CLASS</li> <li>- It's very challenging, but you do ultimately become a much better programmer. The TA's are always there to help and they are very very very intelligent.</li> <li>- It is a heavier workload than the intro course, spread over four larger assignments. It's rewarding.</li> </ul> <p>The class is also smaller than the intro course and TA's and CA's are very readily accessible and an excellent resource for help.</p> <ul style="list-style-type: none"> <li>- the programming is not easy, pointers take a while to get used to, but this class is nothing you can't handle if you have time to make your program submissions perfect before you hand them in.</li> <li>- It's really tough.</li> <li>- You will probably get a bad grade on the first project because you don't have many of the fundamentals that you need, but this is ok as you will get better.</li> <li>- Be ready to do a lot of work for the projects; start early!</li> <li>- The projects are challenging, and you have to pay attention in class to keep up with the material and understand the concepts.</li> <li>- Hard</li> <li>- I was a less than 0.2 of a point away from the next letter grade, but I don't think that they gave it to me. There appears to be no curve at all. I would definitely recommend the course to others, especially with Dr. Amir as the professor.</li> <li>- This course is awesome</li> <li>- Dr. Amir is extremely friendly, approachable, helpful, and knowledgeable. You need to put in your best effort because Dr. Amir and the course assistants can recognize when you put in the time and you will get much more out of the course if you spend time carefully crafting designs and tediously perfecting efficient functions. You absolutely must take this course with Dr. Amir if possible. I wish I could have a professor of his caliber in all of my classes at Hopkins.</li> <li>- some programming experience will be a plus</li> </ul>	






12 - This course improved my appreciation for and/or ability to engage in life-long learning.					
Response Option	Weight	Frequency	Percentage	Percent Responses	Means
Disagree strongly	(1)	0	0.00%		
Disagree somewhat	(2)	0	0.00%		
Neither agree nor disagree	(3)	4	16.67%		
Agree somewhat	(4)	8	33.33%		
Agree strongly	(5)	12	50.00%		
				0 25 50 75 100	
Return Rate					
24/28 (85.71%)					

JHU - Krieger School of Arts & Sciences / Whiting School of Engineering  
 ASEN.2012.Spring

Course: EN.600.120.02.SP12 : Intermediate Programming

Instructor: Yair Amir \*

13 - This course deepened my understanding of the social impact of computing.						
Response Option	Weight	Frequency	Percentage	Percent Responses	Means	
Disagree strongly	(1)	1	4.17%			
Disagree somewhat	(2)	3	12.50%			
Neither agree nor disagree	(3)	7	29.17%			
Agree somewhat	(4)	7	29.17%			
Agree strongly	(5)	7	29.17%			
				0 25 50 75 100		
<b>Return Rate</b>						
24/28 (85.71%)						

14 - This course enhanced my ability to work effectively in a team.						
Response Option	Weight	Frequency	Percentage	Percent Responses	Means	
Disagree strongly	(1)	5	20.83%			
Disagree somewhat	(2)	3	12.50%			
Neither agree nor disagree	(3)	10	41.67%			
Agree somewhat	(4)	3	12.50%			
Agree strongly	(5)	3	12.50%			
				0 25 50 75 100		
<b>Return Rate</b>						
24/28 (85.71%)						