

Project Title: **2022W2 UBCO Instructor SEI Surveys**

Course Audience: **180**  
Responses Received: **62**  
Response Ratio: **34%**

## Report Comments

### Recommended Minimum Response Rates

Class Size	Recommended Minimum Response Rates based on 80% confidence & $\pm 10\%$ margin
< 10	75%
11 - 19	65%
20 - 34	55%
35 - 49	40%
50 - 74	35%
75 - 99	25%
100 - 149	20%
150 - 299	15%
300 - 499	10%
> 500	5%

## Legend

N: Expected  
n: Responded

### Frequency Distribution

SD: Strongly Disagree  
D: Disagree  
N: Neutral  
A: Agree  
SA: Strongly Agree  
N/A: Not applicable

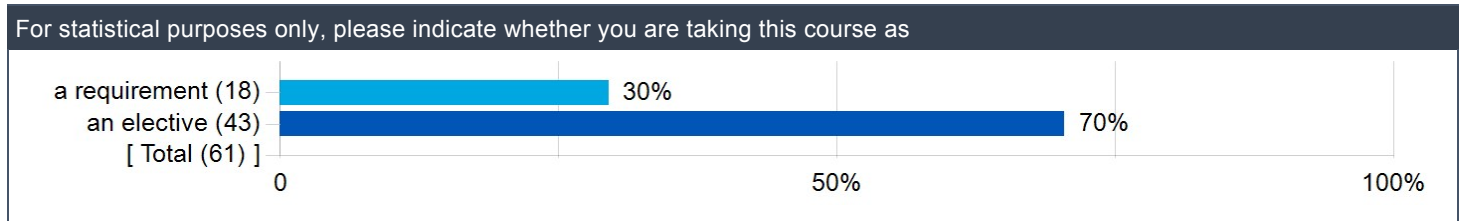
### Statistics

IM: Interpolated Median

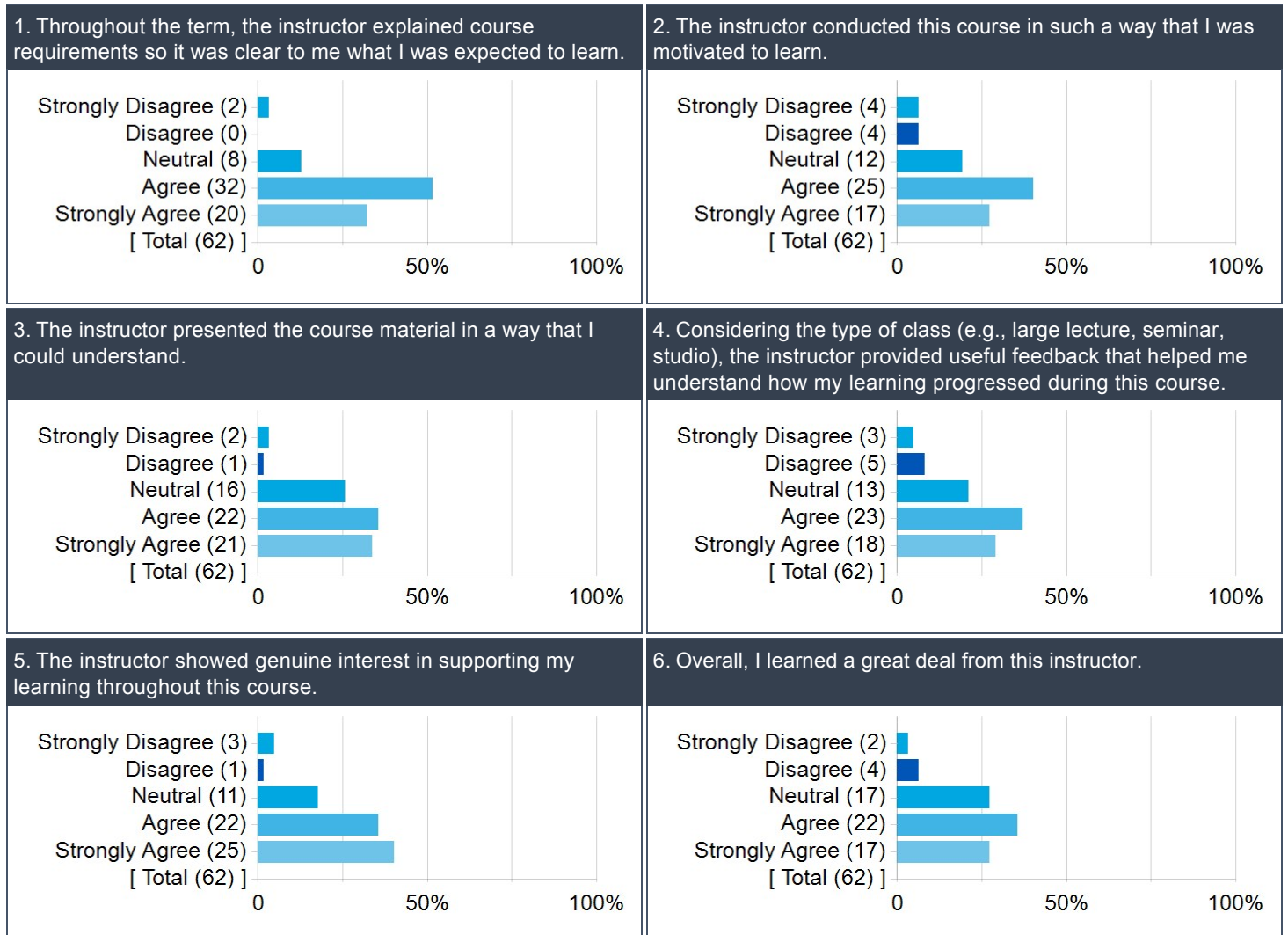
Creation Date: **Monday, May 8, 2023**

## Detailed Results

For statistical purposes only, please indicate whether you are taking this course as



### University Module Questions



UBCO Student Experience of Instruction

	N	n	SD	D	N	A	SA	IM	DI
Throughout the term, the instructor explained course requirements so it was clear to me what I was expected to learn.	180	62	2	0	8	32	20	4.2	0.4
The instructor conducted this course in such a way that I was motivated to learn.	180	62	4	4	12	25	17	3.9	0.6
The instructor presented the course material in a way that I could understand.	180	62	2	1	16	22	21	4.0	0.5
Considering the type of class (e.g., large lecture, seminar, studio), the instructor provided useful feedback that helped me understand how my learning progressed during this course.	180	62	3	5	13	23	18	3.9	0.6
The instructor showed genuine interest in supporting my learning throughout this course.	180	62	3	1	11	22	25	4.2	0.5
Overall, I learned a great deal from this instructor.	180	62	2	4	17	22	17	3.9	0.6

Question	%Favourable
Throughout the term, the instructor explained course requirements so it was clear to me what I was expected to learn.	84%
The instructor conducted this course in such a way that I was motivated to learn.	68%
The instructor presented the course material in a way that I could understand.	69%
Considering the type of class (e.g., large lecture, seminar, studio), the instructor provided useful feedback that helped me understand how my learning progressed during this course.	66%
The instructor showed genuine interest in supporting my learning throughout this course.	76%
Overall, I learned a great deal from this instructor.	63%

## Open ended feedback

**Do you have any suggestions for what the instructor could have done differently to further support your learning?**

Comments
Make Better powerpoints
Maybe using the traditional grading system like COSC 121
I do think midnight deadlines are better, so I would suggest changing those. Everything else was awesome.
N/A
good enough
This course has many shortcomings, but for the most part I don't think they were Dr. Moosvi's fault. He was probably aware that most of his students are computer science or data science majors with existing knowledge about coding, and that must be frustrating when you have to teach them coding as if it's their first time. Most of the problems I had with COSC 123 belong in the 'UBC-wide comments', not here. However, 2 things:
1. THERE ARE NO STAKES. The reason only like 20% of the class shows up to lectures is because there are no consequences for not going. Most of the content is learned through labs, and if you miss something in lecture you can catch it later. Tests are open-book, deadlines are soft, you can ask questions on Ed-discussion. While I think those are good things, Dr. Moosvi could drive engagement by using i-Clicker (or something similar) to check attendance and to encourage paying attention and actually understanding what is going on.
2. LACK OF PROCESSING KNOWLEDGE. I get it. Dr. Moosvi didn't build the course. But it seemed at times that he was only marginally more comfortable with Processing than we were. I forgive him for it, I really do, but if he continues to teach the course it is definitely something he should brush up on.
I feel like there have been a few times that I couldn't understand what a task needed, and neither did the TAs when I went to get help
All he did was cover pre-requisite material for the entire semester. If we had focused more on processing and git, and less about how to actually program (the pre-requirement for this class) I think It would have been more interesting. He also just read off of slides the entire time like it was a script. did the same thing in physics 111. beyond the technical issues he always had, He would constantly forget how to actually perform what it was he was teaching, with consistent mistakes everywhere. People will vote him up because his class is a free A, which is nice, but as a learning experience it was horrible.
Labs could be much more explicit with how they are going to be graded.
I found it hard to keep up with the deadlines for the course due to the uneven deadlines. Unfortunately, this also works and I do not have a suggestion that could be implemented to change it as it is a very functional way of testing and grading us on our abilities. Overall the instructor had a fairly well-structured course.
The instructor could improve by guiding students to use the documentation page when they are stuck while coding. Reading an understanding documentation is a key factor of being a computer scientist.
None
n/a
more tests
nope :)
Sometimes the assignments got tedious and felt repetitive as I kept doing them.
No suggestions best prof ever
Not really, everything is good
Show more examples of active coding in class.
having more practice time on class
For lab assignments, when we are expected to use past lab codes, it would be helpful if they specify which past lab assignment I am supposed to take the code from.
more music variety
Maybe it was due to missing the first few classes due to other circumstances, but I was unsure if the activity portion of the labs were required until much later in the lab. With the resubmission parts of the lab, I found that one could use future labs to mitigate the work and learning needed to do for previous labs. I would consider having a method to demonstrate learnings for previous labs that cannot be done by looking at future labs and taking their work to do previous labs.
Become more interactive with students during lectures. Implementing iClickers would be a way to maintain student attendance.

Comments
<p>Maybe it will be more practical if the deadline for labs can be moved from Wednesday to Friday, then we don't have to worry about two different lab assignments in a week.</p>
<p>he should become less firm on the 6 pm deadline and during lectures, less information dumping and more nuanced explanations</p>
<p>Way less assignments, having 5+ assignments in one week is ridiculous and very demotivating.</p>
<p>Perhaps resubmissions wouldn't be a necessary feature if assignments were not so task/information dense, because I found the formatting of the course to be egregious in this regard at points, especially while taking a full course load. COSC 123 in my understanding is an elective/Arts degree's version of 121, and yet having taken both, I can say 121 did not require nearly as much work to attain a high grade and stay on top of material.</p>
<p>lit just seemed like the teacher didn't care about what he was teaching at all.</p>
<p>The test system seemed too easy, as in most of the questions were code that could be copy-pasted into the Processing IDE and solved without thinking about it, which I'm sure many people did. The long coding questions in tests did not suffer from this, I thought they were good. I think to improve the tests, the questions could be more like word problems, ex. "explain the difference between for-loops and while-loops", instead of some problems like "what is the output of this code". Overall this is just a minor suggestion, the course was great.</p>
<p>Perhaps include a copy of prior labs practice which have been completed so we can have more practice. and a practice exam sheet for the final.</p>
<p>Overall, support and responses were well done, however, when asked questions in class for help in the open work time I received little to no guidance on how to fix the code that I had gotten wrong. I was eventually able to figure out what went wrong but that was on my own time, which gave the impression that he was slightly under-prepared for the content we were covering. Other than that one instance, the rest of the course and content was covered fully and in-depth.</p>

**Please identify what you consider to be the strengths of this course.**

Comments
Bonus test
Coding note taking
Dr. Moosvi was very kind and accommodating, he made it easy to excel in his course.
The extension requests and ability to take bonus tests.
Labs are optional to attend, tests are done with the computer, learning logs to reflect on new concepts.
construct the creativity for students
<ul style="list-style-type: none"> <li>– It is a really cool way of re–framing basic coding concepts by using graphics rather than text to introduce them</li> <li>– It is a great introduction to learning about computer graphics in general because it makes you consider colour, coordinate systems/transformations, frame rate, performance, and how interactive media actually works</li> <li>– It encourages creativity in terms of coding, problem–solving and visual aesthetic—something that is not done enough in other introductory programming courses</li> </ul>
Having all the powerpoints on Canvas and good choices on what apps and sites are used
The course is a great introduction to those who are interested in computer science but do not understand the fundamentals of how or why code works. The course is structured very well and is taught by an instructor who cares about your learning.
Helps incorporate COSC111 material into a more visual way.
Creativity
good
Being familiar with Processing and know how to be creative with code learned.
It teaches a good base for programming in a nice visual way
The strengths would be the open web/book which was super helpful and put off a lot of stress during tests.
The innovativeness that it requires
Easy to follow, introduces to a lot of helpful programs, allows flexibility.
java programming
alternate grading scheme promoted more active learning
allowing labs to be resubmitted to demonstrate learnings. The slow progression of topics were well–done and clearly taught.
This course is beginner friendly.
the flexibility of assignments and resubmissions
learning processing 4.1.1, creating visual code
It's fun. You get to explore graphics and applications of COSC 111 topics in a versatile graphics environment. It's easy, especially as an elective for experienced coders. The labs may ask a lot, but everything is easy to perform (if not juvenile) It's entry–level. Unlike COSC courses starting year two, this course has no theory involved and it's all applications.
Active–learning structure, very interesting material, opportunities to improve o graded work.
slides are clear, straght forward. There's plenty of opportunities for help to learn
The strengths of the course reside in the multiple attempts allowed for assessments, which grants one the ability to better explore the topic at hand and to experiment with the content before having a finalized grade attached to it.

**Please provide suggestions on how this course might be improved.**

Comments
Better powerpoints
TAs could mark assignments at a faster rate. Otherwise, great course!
Currently, the course seems to be at a very good standpoint of being flexible for the students but also allows them to learn a lot.
Maybe less activities tasks and lab tasks overall for the lab content of the course.
replace the final exam with a large design assignment
This is a weird course. It is taught as if the students have little previous coding experience, and yet the pre-requisite is either COSC 122 or COSC 111. Admittedly, non Comp Sci students coming from COSC 122 only have basic coding knowledge, but the reality is that the overwhelming majority of students taking this class already know how about conditionals, functions, arrays etc. This means that an awful lot of time is wasted making sure that students feel comfortable, when more time could be spent actually teaching us new stuff. There is a reason no one showed up to learn about arrays – most of us already knew what they were.
Confusingly, this course is also allowed as a pre-req for COSC 121 instead of COSC 111. I feel sorry for anyone who only takes this course before COSC 121, because it doesn't get in depth enough.
My point is: I think that the material of this course could be utilized better if basic coding knowledge was assumed. This shouldn't be an introductory coding course... it should be an introductory graphics course *for* coders. Make COSC 111 a hard pre-req, and then give COSC 123 some meat by diving deeper into the unique coding concepts we can learn in Processing. I think it's a great course to take concurrently with COSC 121 – e.g. when we learned about recursion in 121, I was able to make fractals in Processing. It offers an introduction to applied programming in ways that COSC 121 does not.
At the very least, give Dr Moosvi a smaller, more personal lecture hall so he doesn't have to reach across the void of COM 201 to talk to the 30 students who showed up.
Having alternative ways of fulfilling a task if one way is not possible for some reason, and a bit longer for submitting the learning logs
The course could improve how the labs are spaced out and the content done in the labs. Instead of labs and activities, the activities themselves could be small parts that when added to the lab help build and improve on the overall game. I found that a weakness of the labs was that they did not focus on a single project or game and rather tried to do too many at once. It would be better to either allow the student to pick a platformer or a space invaders game. The games would be built incrementally, adding in different topics that introduce new mechanics/enemies/abilities to the game. There were a lot of interesting aspects, but the fact that there was no single repository to improve on did not help teach how to maintain a repo using issues and comments. Having a more streamlined game would teach students how to plan and execute a project over a period of 4 months rather than make random small pieces of code for the sake of receiving a grade and getting them to race through a lab every week. This might help build interest in the course as the instructor has developed a great syllabus/unsyllabus for students that helps them learn at their own pace.
n/a
good
More advanced concepts
I think since I already learned the same things in this course in COSC111, the course felt a bit repetitive and boring.
Nothing
Reviews too much material from COSC111, which is a prerequisite.
more tests
answers to multiple choice questions in slides
Similar to extra tests, have resubmitted labs be done via a different requirement that still demonstrates the learnings of the lab so you cant use future labs to do the work of previous labs.
No suggestions.
the content could be more organized and specific and less general
I think COSC 123 could be improved by, instead of utilization of the Processing PDE, use of the actual Java Graphics frameworks like AWT or JavaFX could genuinely be more helpful for students trying to learn how to use these tools without the handholding nature of Processing, which does most of the work for you. The course could benefit from teaching about useful Java packages for running and rendering in Graphics threads.
The one major gripe that I came out of this class with was the speed of returning assessments. For example, it would take 4 weeks to get a test back, when we have the ability to take a re-test on that same topic the next week. So one would have to go into the re-test without knowing what one got on the first test.

## Explanatory Note

### Percent Favourable Rating

This is the percentage of respondents who rated the instructor a 4 or 5 (Agree or Strongly Agree).

### Interpolated Median

The data collected for Student Experience of Instruction (SEI) are ordinal in nature, with a natural order (from 1 to 5). While the mean may be used as a measure of central tendency for such data, it is not an appropriate or accurate representation of SEI data (cf. Stark & Freishtat, 2014). The usual measure of central tendency for ordinal data is the median. As a result, we have been reporting the mean and the median for the last several years. After considerable thought and data modeling, we now believe that the interpolated median is the best representation of the data, since it takes the frequency distribution into account.

Consider the following example from 2015W, the two course sections have identical mean (3.8). However, the instructor in section 2 received 77% favourable (4-5) ratings, compared to 53% for the instructor in section 1. The Interpolated median values of (3.7 and 4.2), much better reflects the distribution of the scores above and below their respective median. Furthermore, the interpolated median is better correlated with percent favourable rating; such that an interpolated median of 3.5 on a Likert scale of 1 to 5, corresponds to 50% favourable rating.

**Frequency Distribution**

Response for University Module Item	Section 1	Section 2
5 = Strongly agree	5	5
4 = Agree	3	5
3 = Neither agree nor disagree	6	0
2 = Disagree	1	2
1 = Strongly disagree	0	1
Mean	3.8	3.8
Median	4.0	4.0
Interpolated Median	3.7	4.2
Percent favourable rating	53%	77%

### Dispersion Index

The dispersion index is a measure of variability suitable for ordinal data (Rampichini, Grilli & Petrucci 2004). This dispersion index has values between zero and 1. A zero dispersion index indicates that all respondents in the section rated their experience of instruction the same. An index value of 1.0 is obtained when respondents are split evenly between the two extreme values (Strongly Disagree & Strongly Agree), a very rare occurrence. In SEI data at UBC, the index rarely exceeds 0.85, and mostly for surveys not meeting the minimum recommended response rate.



