

- What is the pedagogical rationale for this course being cross-listed as a COSC course?
 - The course teaches programming from a data science lens and examples used are based on data. The course is also an introduction to programming in Python, so it serves as a COSC course and would be of interest to CS majors and students potentially considering a CS major (especially BA students).
 - To better reflect those facts, the calendar description has been revised to (changes in bold):

Fundamentals of data science **and programming** with an emphasis on **problem solving, testing, debugging, and working with data sets**. Real-world applications from disciplines in the sciences, humanities, medicine, engineering, social sciences, business and others. No prior computing background is required.
- Can this course be used to replace COSC 111 for courses with COSC 111 as a prerequisite? Or be used as a prerequisite to COSC 123?
 - No, the course is too different from COSC 111 to be used to replace COSC 111, even as a pre-requisite to COSC 123.
 - This can be revisited in the future for COSC 123 if it ever switches to Python (processing has python bindings: <https://py.processing.org>)
- Should the course require any highschool level math prerequisites?
 - The philosophy of this course is to make it accessible to students of all backgrounds, even those in the Arts so there is intentionally no pre-requisites set.
- The syllabus mentions "think computationally" as a learning outcome. Likewise, the calendar description states "an emphasis on computational thinking". These terms are often used to refer to non-coding activities which don't seem to be the case here. Please rephrase these statements so as to not be confused with non-programming methodologies.
 - Interesting, the operating definition of computational thinking we had used included the following competencies: data analysis, pattern recognition, algorithm design, data visualization, debugging/error detection, and problem solving (see attached paper).
 - However, in light of the potential for confusion, we have replaced the learning outcome that mentions "thinking computationally" with: "develop the ability to use programming principles to solve problems, conduct data analyses, create data visualizations, recognize patterns in data, and detect errors in code."
 - Similarly, the calendar description has been updated to remove "computational thinking" and replace it with "...an emphasis on problem solving, ..."
- In comparison to the other course components, what makes the Project component so special that students are considered to pass the course with 40%?
 - In the revised version, the project passing requirement has been increased to 50%
 - Note that the passing requirements are *AND* not *OR* so all criteria must be met for students to pass the course; increasing the threshold makes it slightly more difficult for students to pass the course however, perhaps for simplicity, consistency, and practicality it is better to set all thresholds at 50%.
- In the evaluation criteria, "Tests" does not seem to include a final exam. This is a violation of senate regulations and the evaluation criteria need to be adjusted accordingly. <https://www.calendar.ubc.ca/okanagan/index.cfm?tree=3,41,89,1008>
 - One of the Data 100 committee members has had preliminary conversations with the Associate Dean for Teaching, Learning and Curriculum (Trudy Kavanagh) to apply for an exemption for Data 100 so it does not have a final exam. Though the application is underway, the process is lengthy and arduous with no guarantee of success.
 - Consequently, a final exam weighted at 20% has been added to the course, other assessment weights adjusted accordingly, and a requirement added that students must pass the final exam with a score of at least 50%