- 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-requsite 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%

DATA 100 Course Syllabus

DATA 100 001 - Introduction to Data Science in Python (3)

Course Description

DATA 100 (3) Introduction to Data Science in Python (3)

> Fundamentals of data science ~~with an emphasis on computational thinking~~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.

Prerequisite: None

Equivalence: COSC 100

Learning Outcomes

Upon successful completion of this course, students will be able to:

> 1. ~~develop the ability to "think computationally" using programming principles.~~
> 1. 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.
> 2. practice the creation of loops, conditionals, and functions to analyze data.
> 3. identify and use different data types to accomplish a variety of data science tasks.
> 4. apply common workflows to load, process, clean, and analyze data.
> 5. appraise the quality of data and assess its limitations in answering questions.
> 6. understand the role of testing and version control to writing sustainable code.
> 7. create reproducible, ethical, and sustainable data analyses.
> 8. apply the skills and techniques in this course to generate reproducible analyses.

Assessment

Item	Weight	~~Weight~~	Frequency
Learning Logs	5%	~~10%~~	Weekly
Labs	25%	~~30%~~	Weekly
Project	25%	~~30%~~	Weekly
Tests	25%	~~30%~~	Bi-weekly
Final Exam	20%		Scheduled during the exam period

Passing Criteria

All students must satisfy ALL conditions to pass the course:

> 1. Pass the Labs with an average grade of at least 50%, with no more than 4 missed labs.
> 2. Pass the Tests with an average grade of at least 50%.
> 3. Pass the Project with a grade of at least 4050%.
> 4. Pass the Course overallFinal Exam with a grade of at least 50%
> 5. Pass the Course overall with a grade of at least 50%.

If a student does not satisfy the appropriate requirements, the student will be assigned the **lower** of their earned course grade or, a maximum overall grade of 45 in the course.

DATA 100 Course Syllabus

DATA 100 001 - Introduction to Data Science in Python (3)

Course Description

DATA 100 (3) Introduction to Data Science in Python (3)

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.

Prerequisite: None

Equivalence: COSC 100

Learning Outcomes

Upon successful completion of this course, students will be able to:

1. 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.
2. practice the creation of loops, conditionals, and functions to analyze data.
3. identify and use different data types to accomplish a variety of data science tasks.
4. apply common workflows to load, process, clean, and analyze data.
5. appraise the quality of data and assess its limitations in answering questions.
6. understand the role of testing and version control to writing sustainable code.
7. create reproducible, ethical, and sustainable data analyses.
8. apply the skills and techniques in this course to generate reproducible analyses.

Assessment

Item	Weight	Frequency
Learning Logs	5%	Weekly
Labs	25%	Weekly
Project	25%	Weekly
Tests	25%	Bi-weekly
Final Exam	20%	Scheduled during the exam period

Passing Criteria

All students must satisfy ALL conditions to pass the course:

1. Pass the Labs with an average grade of at least 50%, with no more than 4 missed labs.
2. Pass the Tests with an average grade of at least 50%.

3. Pass the Project with a grade of at least 50%.
4. Pass the Final Exam with a grade of at least 50%
5. Pass the Course overall with a grade of at least 50%.

If a student does not satisfy the appropriate requirements, the student will be assigned the **lower** of their earned course grade or, a maximum overall grade of 45 in the course.

Textbook

Portions of the following (open source) textbooks will be assigned as reading:

- Python Data Science Handbook, by Jake VanderPlas
- Python for Data Analysis, by Wes McKinney

Eventually, an open textbook will be developed using open resources.

Schedule

Wk	Starting	Topics	Project	Lab	Learning Logs	Tests
1	Week 1	Introduction to Data Science			LL 1	
2	Week 2	Terminal and Jupyter Notebook		L1	LL 2	Test 1
3	Week 3	Version Control with Git	PM1	L2	LL 3	
4	Week 4	Introduction to Python		L3	LL 4	Test 2
5	Week 5	Loading and working with data	PM2	L4	LL 5	
6	Week 6	Data Types: Lists and Dictionaries		L5	LL 6	Test 3
7	Week 7	Computation with `numpy`	PM3	L6		
8	Week 8	Controlling the flow			LL 7	Test 4
9	Week 9	Organizing your code	PM4	L7	LL 8	
10	Week 10	Objects in Python			LL 9	Test 5
11	Week 11	Data analysis with `scipy` and `pandas`	PM5	L8	LL 10	
12	Week 12	Data visualization		L9	LL 11	
13	Week 13	Releases and Reproducibility	PM6	L10	LL 12	