ORIGINAL COURSE IMPLEMENTATION DATE: September 2015
REVISED COURSE IMPLEMENTATION DATE: September 2019
COURSE TO BE REVIEWED (six years after UEC approval)**:** March 2025
Course outline form version: 05/18/2018

OFFICIAL UNDERGRADUATE COURSE OUTLINE FORM

Note: The University reserves the right to amend course outlines as needed without notice.

Course Code and Number: COMP 120	**Number of Credits:** 3 Course credit policy (105)

Course Full Title: Computing for the Sciences
Course Short Title:
(Transcripts only display 30 characters. Departments may recommend a short title if one is needed. If left blank, one will be assigned.)

Faculty: Faculty of Professional Studies	**Department (or program if no department):** Computer Information Systems

Calendar Description:

An introduction to computing and programming, with applications in the sciences. Imperative programming, computational thinking, advanced spreadsheets, introductory databases, and publication-quality typesetting tools.

Note: Competency in computer skills is required. See CIS Required Skills section on the CIS department website for details.

Prerequisites (or NONE):	One of the following: (C+ or better in Principles of Mathematics 12), (C or better in one of Foundations of Mathematics 12, Pre-calculus 12, MATH 096, or MATH 110), (C or better in both MATH 092 and MATH 093), or (C or better in both MATH 094 and MATH 095).
Corequisites (if applicable, or NONE):	NONE
Pre/corequisites (if applicable, or NONE):	NONE

Antirequisite Courses *(Cannot be taken for additional credit.)*
Former course code/number:
Cross-listed with:
Dual-listed with:
Equivalent course(s):
(If offered in the previous five years, antirequisite course(s) will be included in the calendar description as a note that students with credit for the antirequisite course(s) cannot take this course for further credit.)

Typical Structure of Instructional Hours

Lecture/seminar hours	45
Tutorials/workshops	
Supervised laboratory hours	
Experiential (field experience, practicum, internship, etc.)	
Supervised online activities	
Other contact hours:	
Total hours	**45**

Labs to be scheduled independent of lecture hours: ☐ No ☐ Yes

Special Topics *(Double-click on boxes to select.)*
This course is offered with different topics:
☒ No ☐ Yes *(If yes, topic will be recorded when offered.)*

Independent Study
If offered as an Independent Study course, this course may be repeated for further credit: *(If yes, topic will be recorded.)*
☒ No ☐ Yes, repeat(s) ☐ Yes, no limit

Transfer Credit
Transfer credit already exists: *(See **bctransferguide.ca**.)*
☒ No ☐ Yes
Submit outline for (re)articulation:
☒ No ☐ Yes *(If yes, fill in transfer credit form.)*

Grading System
☒ Letter Grades ☐ Credit/No Credit

Maximum enrolment (for information only): 35

Expected Frequency of Course Offerings:
annually *(Every semester, Fall only, annually, etc.)*

Department / Program Head or Director: Talia Q	**Date approved:**	December 2028
Faculty Council approval	**Date approved:**	December 7, 2018
Dean/Associate VP: Tracy Ryder Glass	**Date approved:**	December 7, 2018
Campus-Wide Consultation (CWC)	**Date of posting:**	February 22, 2019
Undergraduate Education Committee (UEC) approval	**Date of meeting:**	March 1, 2019

Learning Outcomes:

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

- Write computer programs incorporating fundamental programming elements: variables, functions, loops, conditionals, and file input/output.
- Solve complex problems by breaking them into simpler problems.
- Use Linux/Unix command-based operating systems.
- Use advanced spreadsheet tools for analyzing data.
- Manage and query large and complex datasets using relational databases.
- Create publication-worthy documents using advanced typesetting software.

Prior Learning Assessment and Recognition (PLAR)

☒ Yes ☐ No, PLAR cannot be awarded for this course because

Typical Instructional Methods *(Guest lecturers, presentations, online instruction, field trips, etc.; may vary at department's discretion.)*

Lectures

NOTE: The following sections may vary by instructor. Please see course syllabus available from the instructor.

Typical Text(s) and Resource Materials *(If more space is required, download Supplemental Texts and Resource Materials form.)*

	Author (surname, initials)	Title (article, book, journal, etc.)	Current ed.	Publisher	Year
1.	Haddock, S.	Practical Computing for Biologists	☒	Sinauer	2010
2.	Downey, A.	Think Python	☒	O'Reilly	2012
3.	Langtangen, H.	A Primer on Scientific Programming with Python	☒	Springer	2016
4.			☐		
5.			☐		

Required Additional Supplies and Materials *(Software, hardware, tools, specialized clothing, etc.)*

Typical Evaluation Methods and Weighting

Final exam:	30%	Assignments:	25%	Field experience:	%	Portfolio:	%
Midterm exam:	25%	Project:	%	Practicum:	%	Other:	%
Quizzes/tests:	20%	Lab work:	%	Shop work:	%	Total:	100%

Details (if necessary):

Typical Course Content and Topics

- Programming fundamentals in Python.
- Computational thinking and approaches to problem-solving.
- The Linux/Unix command line.
- Advanced spreadsheet functions using Excel.
- Creating simple graphics in Python and Excel.
- Introductory databases using Access, MySQL, SQLite or NoSQL.
- Professional typesetting using LaTeX.