



ORIGINAL COURSE IMPLEMENTATION DATE: September 2005
 REVISED COURSE IMPLEMENTATION DATE: September 2019
 COURSE TO BE REVIEWED (six years after UEC approval):
 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 152	Number of Credits: 4 Course credit policy (105)														
Course Full Title: Introduction to Structured Programming Course Short Title: <i>(Transcripts only display 30 characters. Departments may recommend a short title if one is needed. If left blank, one will be assigned.)</i>															
Faculty: Faculty of Professional Studies	Department (or program if no department): Computer Information Systems														
Calendar Description: This course is an introduction to structured computer programming. Students will study algorithms and top-down design, and will implement algorithms in a procedural programming language. Lab exercises and programming assignments will emphasize scientific and numerical applications. Note: Competency in computer skills is required. See CIS Required Skills section on the CIS department website for details. Note: Students with credit for COMP 150 cannot take this course for further credit.															
Prerequisites (or NONE):	C+ or better in one of the following: Principles of Mathematics 12, Pre-calculus 12, MATH 093, or MATH 095.														
Corequisites (if applicable, or NONE):	NONE														
Pre/corequisites (if applicable, or NONE):	NONE														
Antirequisite Courses <i>(Cannot be taken for additional credit.)</i> Former course code/number: Cross-listed with: Dual-listed with: Equivalent course(s): <i>(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.)</i>	Special Topics <i>(Double-click on boxes to select.)</i> This course is offered with different topics: <input checked="" type="checkbox"/> No <input type="checkbox"/> Yes <i>(If yes, topic will be recorded when offered.)</i> Independent Study If offered as an Independent Study course, this course may be repeated for further credit: <i>(If yes, topic will be recorded.)</i> <input type="checkbox"/> No <input type="checkbox"/> Yes, repeat(s) <input type="checkbox"/> Yes, no limit														
Typical Structure of Instructional Hours <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Lecture/seminar hours</td><td style="text-align: center;">45</td></tr> <tr><td>Tutorials/workshops</td><td></td></tr> <tr><td>Supervised laboratory hours</td><td style="text-align: center;">15</td></tr> <tr><td>Experiential (field experience, practicum, internship, etc.)</td><td></td></tr> <tr><td>Supervised online activities</td><td></td></tr> <tr><td>Other contact hours:</td><td></td></tr> <tr><td style="text-align: right;">Total hours</td><td style="text-align: center;">60</td></tr> </table>	Lecture/seminar hours	45	Tutorials/workshops		Supervised laboratory hours	15	Experiential (field experience, practicum, internship, etc.)		Supervised online activities		Other contact hours:		Total hours	60	Transfer Credit Transfer credit already exists: <i>(See bctransferguide.ca.)</i> <input type="checkbox"/> No <input checked="" type="checkbox"/> Yes Submit outline for (re)articulation: <input checked="" type="checkbox"/> No <input type="checkbox"/> Yes <i>(If yes, fill in transfer credit form.)</i>
Lecture/seminar hours	45														
Tutorials/workshops															
Supervised laboratory hours	15														
Experiential (field experience, practicum, internship, etc.)															
Supervised online activities															
Other contact hours:															
Total hours	60														
Labs to be scheduled independent of lecture hours: <input checked="" type="checkbox"/> No <input type="checkbox"/> Yes	Grading System <input checked="" type="checkbox"/> Letter Grades <input type="checkbox"/> Credit/No Credit Maximum enrolment (for information only): 35 Expected Frequency of Course Offerings: Once per year <i>(Every semester, Fall only, annually, etc.)</i>														
Department / Program Head or Director: Talia Q	Date approved: December 2028														
Faculty Council approval	Date approved: December 7, 2018														
Dean/Associate VP: Dr. 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:

- Design a structured solution to a problem by repeatedly breaking the problem into a sequence of simpler subproblems.
- Implement a structured design as a computer program, with separate functions corresponding to the subproblems of the design.
- Describe parameter-passing mechanisms for functions, and choose appropriate mechanisms for programming tasks.
- Describe common looping and selection structures, and choose appropriate structures for the steps of an algorithm.
- Write programs which perform sequential input and output using either keyboard and screen or files.
- Create and use one- and two-dimensional arrays for storing and manipulating data.
- Create and use record structures for storing and manipulating data.

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.*)

The course will be delivered in lecture-lab format, with numerous demonstrations and hands-on activities. The lab portion gives students and the instructor the ability to view and interact with current projects.

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. Savitch, W.	Problem Solving with C++	<input checked="" type="checkbox"/>	Pearson	2014
2.		<input type="checkbox"/>		
3.		<input type="checkbox"/>		
4.		<input type="checkbox"/>		
5.		<input type="checkbox"/>		

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

Final exam:	40%	Assignments:	15%	Field experience:	%	Portfolio:	%
Midterm exam:	%	Project:	%	Practicum:	%	Other:	%
Quizzes/tests:	25%	Lab work:	20%	Shop work:	%	Total:	100%

Details (if necessary):**Typical Course Content and Topics**

- Overview of programming
- Problem-solving and C++
- Variables, constants, and assignment statements
- Built-in and programmer-defined functions
- Selection structures (branching)
- Repetition structures (looping)
- Sequential access files
- Arrays
- Structs