

## OFFICIAL UNDERGRADUATE COURSE OUTLINE FORM

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

<b>Course Code and Number:</b> COMP 370		<b>Number of Credits:</b> 3 <a href="#">Course credit policy (105)</a>															
<b>Course Full Title:</b> Software Engineering <b>Course Short Title:</b> <i>(Transcripts only display 30 characters. Departments may recommend a short title if one is needed. If left blank, one will be assigned.)</i>																	
<b>Faculty:</b> Faculty of Professional Studies		<b>Department (or program if no department):</b> Computer Information Systems															
<b>Calendar Description:</b> <p>A detailed and comprehensive study of object-oriented and classical software engineering techniques. This is an addition to the analysis and design work covered in CIS 270. Topics cover all aspects of the software life-cycle. Emphasis is placed on object-oriented techniques and the Unified Modeling Language (UML).</p> <p>Note: Students with credit for CIS 370 cannot take this course for further credit.</p>																	
<b>Prerequisites (or NONE):</b>		COMP 251 and CIS 270. Note: Students accepted to a Computing Science major or minor may register with permission of the department. The CIS 270 prerequisite will be waived for these students															
<b>Corequisites (if applicable, or NONE):</b>		None															
<b>Pre/corequisites (if applicable, or NONE):</b>		None															
<b>Antirequisite Courses</b> <i>(Cannot be taken for additional credit.)</i> Former course code/number: <b>CIS 370</b> Cross-listed with: Dual-listed with: Equivalent course(s): <b>CIS 370</b> <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>		<b>Special Topics</b> <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>															
		<b>Independent Study</b> If offered as an Independent Study course, this course may be repeated for further credit: <i>(If yes, topic will be recorded.)</i> <input checked="" type="checkbox"/> No <input type="checkbox"/> Yes, repeat(s) <input type="checkbox"/> Yes, no limit															
		<b>Transfer Credit</b> Transfer credit already exists: <i>(See <a href="#">bctransferguide.ca</a>.)</i> <input checked="" type="checkbox"/> No <input type="checkbox"/> Yes Submit outline for (re)articulation: <input type="checkbox"/> No <input type="checkbox"/> Yes <i>(If yes, fill in transfer credit form.)</i>															
<b>Typical Structure of Instructional Hours</b> <table border="1"> <tr> <td>Lecture/seminar hours</td> <td>23</td> </tr> <tr> <td>Tutorials/workshops</td> <td></td> </tr> <tr> <td>Supervised laboratory hours</td> <td>22</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><b>Total hours</b></td> <td><b>45</b></td> </tr> </table>		Lecture/seminar hours	23	Tutorials/workshops		Supervised laboratory hours	22	Experiential (field experience, practicum, internship, etc.)		Supervised online activities		Other contact hours:		<b>Total hours</b>	<b>45</b>	<b>Grading System</b> <input checked="" type="checkbox"/> Letter Grades <input type="checkbox"/> Credit/No Credit	
Lecture/seminar hours	23																
Tutorials/workshops																	
Supervised laboratory hours	22																
Experiential (field experience, practicum, internship, etc.)																	
Supervised online activities																	
Other contact hours:																	
<b>Total hours</b>	<b>45</b>																
Labs to be scheduled independent of lecture hours: <input checked="" type="checkbox"/> No <input type="checkbox"/> Yes		<b>Maximum enrolment (for information only):</b> 35 <b>Expected Frequency of Course Offerings:</b> Once per year <i>(Every semester, Fall only, annually, etc.)</i>															
<b>Department / Program Head or Director:</b> Edward Lo		<b>Date approved:</b> December 20, 2018															
<b>Faculty Council approval</b>		<b>Date approved:</b> March 15, 2019															
<b>Dean/Associate VP:</b> Tracy Ryder Glass		<b>Date approved:</b> March 15, 2019															
<b>Campus-Wide Consultation (CWC)</b>		<b>Date of posting:</b> June 21, 2019															
<b>Undergraduate Education Committee (UEC) approval</b>		<b>Date of meeting:</b> October 25, 2019															

**Learning Outcomes:**

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

- Illustrate different software life-cycle models, activities occurring in each phase of software life-cycle, various testing techniques, methods of planning and estimating, and strategies of improving software reusability and portability.
- Draw up the requirements workflow.
- Perform structured system analysis workflow.
- Perform functional, class, and dynamic modeling workflow.
- Perform object-oriented design, data flow analysis and transaction analysis workflow.
- Apply UML in a team setting for the analysis and design of a small application.
- Apply the software process for the development of a small application.
- Perform unity, integration, product, and acceptance testing.
- Explain the importance and challenge of post-delivery maintenance.

**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, labs, and assignments.

**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. Schach, S.R.	Classical and Object-Oriented Software Engineering with UML and C++	<input checked="" type="checkbox"/>	WCB/McGraw-Hill	2010
2. Sommerville, I.	Software Engineering	<input checked="" type="checkbox"/>	Addison-Wesley	2015
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:	35%	Assignments:	20%	Field experience:	%	Portfolio:	%
Midterm exam:	15%	Project:	30%	Practicum:	%	Other:	%
Quizzes/tests:	%	Lab work:	%	Shop work:	%	Total:	100%

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

Scope of software engineering

The software process

- Requirements
- Specification
- Object-oriented analysis
- Design
- Implementation
- Integration
- Maintenance
- Retirement
- Software life-cycle models
- Testing
- Review of objects
- Reusability, portability, interoperability
- Planning and estimating