ORIGINAL COURSE IMPLEMENTATION DATE: September 2019

REVISED COURSE IMPLEMENTATION DATE:

COURSE TO BE REVIEWED (six years after UEC approval)**:** February 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 386	**Number of Credits:** 3 Course credit policy (105)

Course Full Title: Coding Best Practices

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:

Provides students with experience using software development tools and workflows. The content will emphasize best practices, quality assurance strategies, style guidelines, and professionalism. Students will learn to select and evaluate emerging technologies for use in their software development workflows.

Prerequisites (or NONE):	Admission to the Coding Skills associate certificate, COMP 359, and COMP 370.
Corequisites (if applicable, or NONE):	
Pre/corequisites (if applicable, or NONE):	

Antirequisite Courses *(Cannot be taken for additional credit.)*

Former course code/number:

Cross-listed with:

Dual-listed with:

Equivalent course(s):

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):

Expected Frequency of Course Offerings:

(Every semester, Fall only, annually, etc.)

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

Department / Program Head or Director: Carl Janzen	**Date approved:**	October 2018
Faculty Council approval	**Date approved:**	November 2018
Dean/Associate VP: Tracy Ryder Glass	**Date approved:**	November 2018
Campus-Wide Consultation (CWC)	**Date of posting:**	January 18, 2019
Undergraduate Education Committee (UEC) approval	**Date of meeting:**	February 1, 2019

Learning Outcomes:

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

- Choose software development tools and workflows that are well-suited for a given software development project.
- Determine the specific need for software development tools and workflows in a given business environment.
- Evaluate the effectiveness of software development tools and workflows.
- Incorporate software development tools and workflows for software systems design.
- Apply best practices during software design and implementation.

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.)*

Any combination of lecture or online instruction

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.	Bernstein, D.S.	Beyond Legacy Code: Nine Practices to Extend the Life (and Value) of Your Software	☒	Pragmatic Programmers LLC	2015
2.			☐		
3.			☐		
4.			☐		
5.			☐		

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

Typical Evaluation Methods and Weighting

Final exam:	%	Assignments:	40%	Field experience:	%	Portfolio:	%
Midterm exam:	%	Project:	60%	Practicum:	%	Other:	%
Quizzes/tests:	%	Lab work:	%	Shop work:	%	Total:	100%

Details (if necessary):

Typical Course Content and Topics

Course topics may by instructor. A typical example of course topics would be:

- Dependency management systems
- Version control systems
- Continuous integration
- Unit testing
- Mocking frameworks
- Integration testing
- Acceptance testing
- Coding style guidelines
- Design style guidelines
- Collaboration and workflow systems
- Quality assurance