Computer Science

CWSEI Department Summary

This document summarizes the Computer Science Department's activities as part of the Carl Wieman Science Education Initiative over the years 2007-2018. In this initial version, it replaces a series of web pages that were updated over the lifetime of the CWSEI. It may be updated in future with more detail on impact.

For questions, please contact:
Rachel Pottinger (rap@cs.ubc.ca) or Warren Code (warcode@science.ubc.ca)

Contents

Overview

Computer Science received seed funding from CWSEI in 2007 and began the efforts listed below in the Fall. The department moved to full funding starting in mid-2008.

Computer Science deployed the CWSEI's scientific approach to improve student learning throughout our curriculum as an iterative process including:

- articulating clear learning goals for our individual courses and our program as a whole,
- developing both formative assessments to guide teaching and learning techniques and summative assessments (i.e., exams) that evaluate those learning goals,
- revising our teaching techniques in response to assessments,
- and establishing administrative structures to sustain these efforts.

Poster (CWSEI EOY 2012): Computer Science SEI: An Overview

Ten Year Effort to Transform Learning

- Funding for pedagogical innovation
- Seed funding provided in 2007
- Full funding started in 2008
- David Cheriton donation in 2010

Key ideas

- Use active learning techniques with proven benefits
- State learning goals
- Measure results
- Sustain innovations despite shifting instructional staff
- Science Teaching & Learning Fellow Model: Hire staff with both domain and pedagogical expertise to support course transformations. (Model met with mixed success in Computer Science due to challenges in retention at all ranks.)

People

CWSEI Dept. Director: Rachel Pottinger, Ian Mitchell (emeritus), Paul Carter (emeritus)

STLFs:

Alice Campbell – involved in CPSC 100, 103, 110, 210
Jessica Dawson – involved in CPSC 100, 103, 110, 210, 320, 344, 444, 430
Hassan Khosravi – APSC 160, CPSC 259, 304
Allison Elliott Tew – involved in CPSC 110, 210, 211, 260, 310, and 317
Ryan Golbeck - involved in CPSC 110 and 210
Ben Yu - involved in CPSC , 111, 121, 211, 213, 221, 304, 310, 322, 404, and APSC 160
Raymond Lister - involved in CPSC 111, CPSC 260, and APSC 160
Beth Simon - involved in the early work of CPSC 101, 111, 121, 211, 213, and 221

Part-time Faculty STLFs (roughly 20% appointments for two years):

Don Acton – involved in CPSC 213, 313, 317
Ed Knorr – involved in CPSC 259, 304, 404
Steve Wolfman– involved in developing a concept inventory for the "foundations of computing" stream (CPSC 121, 221, 320)

Faculty: D. Acton, M. Allen, P. Belleville, G. Carenini, P. Carter, C. Conati, A. Condon, M. Dulat, K. Eiselt, M. Feeley, M. Friedlander, W. Heidrich, H. Hoos, N. Hutchinson, G. Kiczales, E. Knorr, K. Leyton-Brown, J. Luk, K. Maclean, J. McGrenere, I. Mitchell, G. Murphy, R. Ng, R. Pottinger, D. Poole, G. Tsiknis, K. Voll, S. Wolfman

Post-docs: Frank Hutter, Gabriel Murray

Activities

Course Transformation: We are working, or have worked in the past, on more than 25 courses ranging from 1st to 4th year level. read more...

Research: The department has been conducting a variety of computer science education research studies, including studies on the use of the PeerWise collaborative multiple-choice question repository, Parson's puzzles, Just-in-Time-Teaching, and decomposition techniques in teaching proof by induction. read more...

Learning Goals Development: A comprehensive set of learning goals (both course-level and topic-level) has been developed for most 1st and 2nd year core courses. Most of these courses use LGs regularly to some extent in class (e.g., many 111 instructors now show the LGs associated with each unit as they lecture on the unit). CPSC 313 also has rough draft LGs. Draft course-level learning goals for the new CPSC 110 and CPSC 210 have been completed. read more...

Learning Goals Study completed (Journal of College Science Teaching, November/December 2009):

What is the Value of Course-Specific Learning Goals?

Courses

Course Transformations

CPSC 100	CPSC 210	CPSC 301	CPSC 404
CPSC 103	CPSC 213	CPSC 302	CPSC 406
CPSC 110	CPSC 221	CPSC 303	CPSC 410
CPSC 121	CPSC 259	CPSC 304	CPSC 411
APSC 160	CPSC 261	CPSC 310	CPSC 415
		CPSC 311	CPSC 416
		CPSC 312	CPSC 418
		CPSC 313	CPSC 420
		CPSC 314	CPSC 421
		CPSC 317	CPSC 422
		CPSC 319	CPSC 424
		CPSC 320	CPSC 425
		CPSC 322	CPSC 426
		CPSC 340	CPSC 430
		CPSC 344	CPSC 444
			CPSC 445

Status as of May 2016:

Course	Learning goals	New Assessments	Improved Methods
CPSC 100: Computational Thinking (Sept '16 start) Faculty: Rachel Pottinger, Will Evans STLF: Jessica Dawson	Course-level goals: complete Topic-level learning goals: drafted	Developing assessments to evaluate first offering of course. Includes pre- and post- surveys on student experience and attitudes (using CAS) to compare with other intro CS courses (CPSC 103, 110, 301)	
CPSC 101: Connecting with Computer Science (Sept '07 start) Faculty: Meghan Allen, Anne Condon, Steve Wolfman, Holger Hoos STLF: Ben Yu, Allison Tew click here to view course materials.	Course-level goals: revision complete Topic-level goals: revision complete	Performed study of instructor & student perception and use of learning goals. Developing assessment to probe student understanding of JavaScript code. Piloted a new Computing Attitude Survey (CAS) in the Fall of 2011 as part of the survey validation process. Pre- and post- surveys of student perceptions of computing science. Peer review of student-generated images through Mechanical TA. Analysis of exam questions to determine coverage and student performance on	Developed and used a broad set of clicker questions Adjusted delivery of course to use Just-in-Time teaching methods with pre-class readings and in-class learning activities. Based on the pre-readings, students submit "reading questions": questions about pre-reading material that was not clear, or questions that go beyond the pre-reading. TAs summarize common themes and pass them along to the instructor, who adapts the classroom session appropriately. Developed instructor course manual. Developed bank of previous exam questions keyed to individual learning goals. Conducted analysis of student retention (how many go on to take a second CPSC course).

		individual learning goals.	
CPSC 103: Introduction to Systematic Program Design (Sept '16 start) Faculty: Meghan Allen STLF: Jessica Dawson	Course-level goals: drafted	Developing assessments to evaluate first offering of course. Includes pre- and post- surveys on student experience and attitudes (using CAS) to compare with other intro CS courses (CPSC 103, 110, 301)	
CPSC 110: Computation, Programs and Programming (Sept '09 start) Faculty: Gregor Kiczales, Paul Carter, Kurt Eiselt, Meghan Allen STLF: Jessica Dawson, Allison Tew, Ryan Golbeck	Course-level and topic-level goals: complete	Have per-question analysis of midterm and final exam data. Developed weekly problem sets that provide students with timely feedback on their learning. Piloted a new Computing Attitudes Survey (CAS) in the Fall of 2011 as part of the survey validation process. 2015: Pre-surveys to compare student experience in 110 between general UBC students and Vantage cohort Winter 2015. 2016: Pre- and post-surveys to evaluate student experiences and attitudes (using CAS). Will continue to	Developing a series of relevant and engaging labs. A plug-in was developed for Dr. Racket IDE so that students can submit assignments electronically from the development environment. This reduces the number of tools that students have to master and allows the course to focus on concepts. Introduced peer-instruction questions at the beginning of each lab, and a peer-review exercise partway through the lab. At the same time updated lab problems to be more in-sync with lecture material.

		compare with other introductory courses (CPSC 301, plus 100, 103 for Winter 2016). Interviews with students who have failed or withdrawn from 110 on barriers to success in the course. Evaluated changes made to labs	
CPSC 111: Introduction to Computation (Sept '07 start) Faculty: Kurt Eiselt, Cristina Conati, Wolfgang Heidrich, J. Luk STLF: Ben Yu, Ray Lister This course is no longer offered. It has been replaced by CPSC 110.	Course-level goals: complete Topic-level goals: complete	Attitudinal survey revised and administered at start and end of term in all sections of summer and fall terms. Cognitive pre-test developed and administered at start of course - the same test was administered to students in APSC 160. Questions targeting specific learning goals have been included on exams. A lab checklist has been developed to track the kinds of problems that students run in to during labs.	
CPSC 121: Models of Computation	Learning goals have been further	Attitudinal surveys developed and	Two-stage exam conducted in summer term. Results published

(Sept '07 start) Faculty: Steve Wolfman, Patrice Belleville, Kimberly Voll, Meghan Allen STLF: Ben Yu Poster (CWSEI EOY 2012): Effective Closed Labs in CPSC 121: Lessons from Eight Terms of Action Research Poster (CWSEI EOY 2010): Adaptation of JiTT in CPSC 121 Poster (CWSEI EOY 2010): Changes in CPSC 121: Towards a coherent picture of computation	categorized to identify pre-class learning goals. Students are expected to achieve pre-class learning goals on their own in advance of the corresponding class.	administered in summer and fall terms. Student interviews conducted in summer and fall terms. Pre and post-tests developed and administered in fall term. MCQs developed for final exam that target specific learning goals. A scenario based think-aloud survey tool has been developed to study how students approach solving mathematical induction problems. The tool is being used to examine the effectiveness of a decomposition technique that teaches students to approach such problems by breaking them down into more manageable pieces.	in ICERI 2009 and further analysis appeared at SIGCSE 2010. Refined online quizzes used to assess pre-class learning goals on the basis of previous term's quiz results. Re-structured in-class problem solving activities to be based on progressive clicker questions with solo- and group-response format - approximately 160 clicker questions developed and used in class. Continued work on labs to make them "open-ended" and driven by student exploration rather than closed-ended. Continued to re-design labs so that they are more closely aligned with core learning goals, avoiding extraneous detail. Established a protocol for preparing TAs to a consistent level and for developing a community of support among the teaching staff.
APSC 160: Introduction to Computation in Engineering Design (Sept '09 start) Faculty: Paul Carter, Ed Knorr	Topic-level goals: complete	Attitudinal survey developed and administered at start and end of term. Analysis pending. Surveys assessing impact of Peer	A series of approximately 30 screencasts have been developed to introduce students to basic concepts. Students are asked to study the screencasts before coming to class. Clicker questions have been

STLF: Hassan Khosravi, Ray Lister, Ben Yu Poster (CWSEI EOY 2010): Student perceptions of online multimedia instruction with JiTT click here to view course materials.		Instruction conducted in week 4 and week 8 of term. Increased number of midterms from 1 to 2 in an effort to provide students with more regular and more accurate feedback on their progress. Piloted a new Computing Attitudes Survey (CAS) in the Fall of 2011 as part of the survey validation process.	developed to assess students' comprehension of the concepts presented in the screencasts. A series of in-class problem sets have been developed that allow students to further develop their understanding of the concepts learned in the screencasts. The new format was incorporated into all four sections of the course offered in 2009/2010. Over 800 students were enrolled. Feedback from students on surveys has been overwhelmingly positive. Analysis of learning gains is in progress. Peerwise online peer question system used in 2015W.
CPSC 210: Software Construction (Jan '10 start) Faculty: Gail Murphy, Meghan Allen STLF: Jessica Dawson, Allison Tew, Ryan Golbeck Poster (CWSEI EOY 2011): Measuring Student Confidence and Lab Material Balance in a CS Course	Topic-level goals: solid draft in place	Parallel assessment in progress with CPSC 211 (the course that CPSC 210 will eventually replace). 2016: Post survey conducted on student experience and attitudes towards CS (using CAS).	Examining the use of cell phones in some labs to increase relevance and student engagement. Poster (CWSEI EOY 2010): A study of student engagement in a course project through the development of cell phone applications
CPSC 211: Introduction to Software	Course-level goals: complete	Attitudinal survey developed and administered at start	Subversion repository has been developed that will facilitate distribution of code to students

Development (Sept '07 start) Faculty: David Poole, Margaret Dulat STLF: Allison Tew, Ben Yu This course has been replaced by CPSC 210	Topic-level goals: complete	and end of both sections of fall term. Parallel assessment in progress with CPSC 210 (the course that will eventually replace CPSC 211)	for labs and lectures. A subversion plug-in for the Eclipse IDE will be used to minimize the number of different tools that students have to master.
CPSC 213: Introduction to Computer Systems & CPSC 261: Basics of Computer Systems (Sept '07 start) Faculty: George Tsiknis, Don Acton STLF: Ben Yu	Course-level goals: complete, under review Topic-level goals: complete, under review	Pre and post-tests developed and administered during summer term. Two-stage exams (published ICERI 2009 & SIGCSE 2010).	All assignment and exam questions were tied to specific learning goals. A set of scripts was developed to provide students with individualized feedback web pages from which they can determine how they are doing - not just on a given assignment, but also on individual learning goals.
CPSC 221: Basic Algorithms and Data Structures (Sept '07 start) Faculty: Kimberly Voll, Ed Knorr, Steve Wolfman STLF: Ben Yu Poster (CWSEI EOY 2013): "Dictionary Wars": An Inverted, Leaderboard-Driven Project for Learning Dictionary Data Structures	Course-level goals: complete Topic-level goals: complete	Post-test developed and administered at the end of summer term. Attitudinal survey revised and administered at the end of 2009 fall term.	Used JITT, in-class group problem solving and peer instruction, clickers or non-electronic clicker equivalents, and web-based quizzes to shift focus of courses to higher-level analysis and problem solving. Based on outcomes from the Foundations of Computing Concept Inventory, a "crash-course" on arrays has been added.

Poster (CWSEI EOY 2012): Planning Assessment for a Game-Like, Highly Reusable Data Structures Assignment Poster (CWSEI EOY 2010): Workshop-Based Learning - Retention and Learning in Data Structures and Algorithms			
CPSC 259: Data Structures & Algorithms for Electrical Engineers (Fall '12 start) Faculty: Ed Knorr STLF: Hassan Khosravi Poster (Science Ed. Open House 2016): Studying the Effects of Adding 'In-Lab' Programming Tests to a CS Service Course Poster (CWSEI EOY 2014): Extending and Improving the Role of Deliberate Practice in CPSC 259 Poster (CWSEI EOY 2013):	Course and topic-level learning goals: complete	Apply pre-test as diagnostic of retention of learning from prerequisite course (APSC 160). Online quizzes for weekly pre-reading Bi-weekly individual programming quizzes alternate with regular pair-programming labs. End-of-term survey of student confidence on learning goals. Per-question final exam analysis	Pre-class readings Developed in-class materials that build on pre-class readings. Revised labs Online simulation for hands-on practice with pointers, types, memory & addressing. In-lab debugging exercise to enforce hands-on practice with the debugger. Peerwise online peer question system used in 2015W.

Introducing Pair Programming in Intermediate C to Non-Specialists			
CPSC 260: Object-Oriented Program Design (Sept '09 start) Faculty: Don Acton STLF: Allison Tew, Ray Lister This course has been replaced by CPSC 213/259/261	Topic-level goals: complete	Pre-test developed and administered in 2009/2010 that measure retention of learning from APSC 160.	A series of clicker questions has been developed.
CPSC 301: Computing in the Life Sciences (Sept '07 start, Jan '14 restart) Faculty: Ian Mitchell, George Tsiknis STLF: Ben Yu	Course & topic-level learning goals: complete	Weekly student surveys in first offering of the course to judge workload, relevance of topics Lab exam 2015: Slightly modified version of CAS administered 2016: Pre- and post-surveys on student experience and attitudes towards CS (using CAS).	Clicker questions In-class group exercises Pair programming in labs 2014: Significant expansion of clicker questions and in-class group exercises accompanied by drastic reduction of traditional lecture slides 2015: Out of 24 classes, 15 include clicker questions and 17 include in-class exercises (for participation credit)
CPSC 304: Introduction to Relational Databases (Sept '09 start) Faculty: Ed Knorr, Rachel Pottinger,	Topic-level goal: complete	Attitudinal survey developed and administered at start & end of term. Pre and post-tests developed to assess change in learning.	Two-stage exams conducted in both midterms of fall 2009 term. A set of new tutorials has been developed and tested in spring 2010 in response to poor attendance in previous terms. The new tutorials are

Raymond Ng STLF: Hassan Khosravi, Ben Yu Poster (CWSEI EOY 2010): CPSC 304 Course Transformation		Student interviews conducted during fall term. Repository of clicker questions developed, including historical scoring data and Bloom's taxonomy classification. Per-question final exam analysis Interaction graph between students in Peerwise collected for further analysis. Isomorphic clicker questions presented several lectures after initial use to measure retention in context of peer instruction.	designed to incorporate active learning and have resulted in higher attendance. Tutorials were improved upon for summer 2010 offering of course to include reflection exercises such as the development of a concept map. Two new tutorials added in summer 2014, plus creation of a parallel version of the 2010 tutorials using alternative database software. Peerwise online peer question system used in 2016S.
CPSC 310: Introduction to Software Engineering (May '10 start) Faculty: Meghan Allen STLF: Allison Tew, Ben Yu	Course and topic level goals: in progress	Diagnostic test developed to assess student preparation in learning	New lecture material under development on topics that are more relevant (e.g., distributed version control systems)
CPSC 313: Computer Hardware & Operating Systems (Sept '12 start) Faculty: Don Acton	Course and topic level goals: draft	Two-stage exams Student survey of unclear topics at the end of term	Lecture-by-lecture timeline with commentary plus recordings of 2013W1 lectures Adoption of a modern version control system for assignment

			distribution and collection (will maintain a record of student solutions for future analysis)
CPSC 314: Introduction to Software Engineering (Dec '14 start) Faculty: Dinesh Pai	Revised course-level learning goals: in progress	Survey to understand student motivation in preparation for upcoming course revisions.	
CPSC 317: Internet Computing (Sept '09 start) Faculty: Don Acton, Norm Hutchinson STLF: Allison Tew Poster (CWSEI EOY 2012): An Evidence-Based Transformation of a Computer Networking Course	Course-level goals: complete Topic-level goals: complete	Conducted initial survey to gather baseline data on student experience in course Baseline data collected in the form of per-question analysis of exam data and attitudinal survey Piloting Pearson's learning catalytics product as an alternative to clickers	Developed group based, in-class learning activities and associated discussion questions focused on specific learning goals. These methods will be administered for the first time in Spring 2012.
CPSC 320: Intermediate Algorithms and Data Structures (Sept '09 start) Faculty: Kimberly Voll, Steve Wolfman STLF: Jessica Dawson		A test of expected prerequisite knowledge was developed and administered at the start of the term Regular, weekly COPUS observations to provide rapid feedback on use of in-class activities and worksheets. (Winter 2015)	Fully flipped version of the course developed and piloted by Steve Wolfman in Winter 2015. Pre-readings selected for each class with quiz at beginning of class. In-class worksheets developed and used as central content and activity for every lecture.

CPSC 322: Artificial Intelligence (Summer '08 start) Faculty: Giuseppe Carenini, Kevin Leyton-Brown STLF: Ben Yu Post-doc: Frank Hutter Graduate Student: Byron Knoll	Course-level goals: complete Topic-level goals: complete	A large body of questions have been developed to be used as the core of future exams. Conducted survey on student use of practice problems and perceived usefulness for their learning.	A set of 19 practice problems complete with solutions have been developed and made available at www.aispace.org/exercises.shtml. 12 of these 19 exercises are integrated with AI Space applets. Background reading for each exercises was also identified in the course textbook. These exercises were also integrated into webCT and two quizzes for each problem were created. An additional set of 19 quizzes covering 7 of the practice problems were made available towards the end of the term. Two new AI Space applets have been developed.
CPSC 340: Machine Learning and Data Mining (Fall '14 start) Faculty: Raymond Ng Post-doc: Yashar Mehdad	Course and topic level learning goals: complete	Post-class student surveys Some repeated exam questions	8 new problem-based laboratory modules to give students hands-on, TA supported practice with lecture concepts. The modules focused on applying different approaches to just two large datasets so that students could apply different techniques to the same data. 4 new assignments to complement the lab modules.
CPSC 344: Introduction to Human Computer Interaction Methods (Fall 2013 start)	Course level learning goals: complete Lecture-level and pre-reading-level learning	Pre-readings and quizzes Replaced course project with a more gradual and scaffolded	Pre-reading allows time for interactive activities in the lecture Developed in-lecture worksheet activities for 19 lectures (of 20 usual lecture slots). Documented

Faculty: Karon Maclean STLF/Faculty: Jessica Dawson Grad student: Oliver Schneider	goals: draft, used in 2014W1	project. Increased frequency of feedback on project progress through regular weekly meetings with TA in tutorials. pre & post student surveys	activities with 'how-to' guides for future instructors. Reduction in the workload for expert TAs (so that more students can be supported without requiring more expert TAs). Development of TA roles for undergraduate TAs in addition to graduate TAs.
CPSC 402: Numerical Linear Algebra (Fall '13 start) Faculty: Michael Friedlander Post-doc: Ting Kei Pong		Same as CPSC 406 below	Same as CPSC 406 below
CPSC 404: Advanced Database Systems (Sept '09 start) Faculty: Ed Knorr STLF: Ben Yu	Course and topic-level learning goals: complete	Attitudinal survey developed and administered at start and end of term Pre and post-tests developed to assess change in learning Clicker questions Per-question final exam analysis	Pre-reading worked examples In-class exercises (using CC workbooks) for almost all lectures Students must submit solutions to pre- and in-class exercises
CPSC 406: Computational Optimization (Fall '12 start) Faculty: Michael Friedlander Post-doc: N. Krislock	Course-level learning goals: complete	Conducted per-question analysis of relevant exam data before and after introduction of case studies to measure change in learning. Case study homework	Developed four case studies for major modules in the course that give students practical, hands-on practice at solving a problem in the field. Each case study is accompanied by related in-class activities (two lecture hours) and a homework assignment.

		and write-up replace roughly half of traditional homework assignments. Survey of student opinion on the effectiveness of the case-study approach.	
CPSC 410: Advanced Software Engineering (Sept '11 start) Faculty: Eric Wohlstadter	Learning goals revised to provide a stronger connection between CPSC 410 and its major prerequisite CPSC 310.		
CPSC 422: Intelligent Systems (Sept '09 start) Faculty: Cristina Conati, Kevin Leyton-Brown, Giuseppe Carenini Post-doc: Frank Hutter	Topic / lecture level learning goals: draft		All assignments have been revised with respect to learning goals and two new assignments have been developed. Exploration of IBM's recently released Watson tools to see whether they might be used to create new hands-on assignments; unfortunately, the tools turned out to be inappropriate.
CPSC 425: Computer Vision (Spring '12 start) Faculty: Bob Woodham, Jim Little, David Lowe Graduate Student: Tristram Southey	Course-level learning goals: complete	Conducted per-question analysis of exam data before and after introducing revised materials to measure change in student learning Student survey on course content, pacing Added seven "practice quizzes" to provide	Development of simulations for use in class. Development of framework that allows students to apply concepts learned in course to real-world computer-vision tasks. Switched language from Matlab to Python so that students can more easily access material outside of the lab

		more regular and timely self-assessment Modified homework assignments to better align with the types of questions asked on exams	
CPSC 430: Computers and Society (Spring ’12 start) Faculty: Kevin Leyton-Brown, Jessica Dawson STLF: Jessica Dawson Graduate Students: Chris Thornton & James Wright	Course-level learning goals: complete	Added 11 weekly essays with an automated calibrated peer review system Compared final exam results across multiple years Post-survey to assess student attitudes toward calibrated peer review system (Fall 2014, 2015)	Identified weekly pre-class reading assignments. Developed related mini-essays that students must complete prior to class. Essays are peer-reviewed. Class time can then focus on discussion, group exercises and analysis of arguments. Developed Mechanical TA, a software system to manage peer review of essays and reduce TA marking effort. Improved user interface developed and tested in fall 2014. Developed bank of calibration essays for students (and TAs) to practice on. Calibration essays are also silently & randomly included into the peer review process to spot-check quality of independent reviewers. 2015: Revised in-lecture activities and developed a set of worksheets for every lecture to facilitate peer-discussions on case studies during lectures.
CPSC 444: Advanced Methods for Human Computer Interaction (Sept '10 start)	Course-level learning goals: complete	Post survey to assess course revisions (Winter 2016).	Added practical component to each tutorial in response to feedback from students. Reworked project and labs to

Faculty: Joanna McGrenere, Jessica Dawson STLF: Jessica Dawson Graduate Student Kailun Zhang			streamline descriptions and milestones. Improved tutorial instructions for TAs Created a new lab to teach Android phone development skills. 2016: Transitioned to a more blended-learning approach: In-person tutorials were eliminated, and content was adapted a combination of online pre-class tutorials and in-lecture activities. Designed and deployed two online experiments that students use to collect experiment data at home (as opposed to in-lab). Developed step-by-step tutorials and Connect assignments to guide students in conducting the experiments and collecting data, performing data analysis, and interpreting their results. Revised and adapted tutorial activities and developed worksheets to be used in lecture. Pre-lecture quizzes and open-ended responses developed in Connect to guide students in completing pre-reading assignments and performing self-assessment of their understanding.

Poster (UBC Science Education Open House 2016): Retiring the Red Pen: Marking Exams Digitally

Poster (UBC Science Education Open House 2016): Using Learning Analytics for Providing

Personalized Content and Feedback in Large Classes

Poster (UBC Science Education Open House 2016): Student Experience in Introductory CS Courses

Poster (UBC Science Education Open House 2015): Mechanical TA: Partially Automated High-Stakes Peer Grading

Poster (UBC Science Education Open House 2015): Student Attitudes Towards Partially-Automated Peer Grading

Poster (CWSEI EOY 2014): Misconceptions & Concept Inventory Questions for Binary Search Trees & Hash Tables

Poster (CWSEI EOY 2014): Using the CWSEI Approach to Updating Computer Science Systems Courses

Poster (CWSEI EOY 2013): A Plan for Transforming Systems and Database Courses in Computer Science

Learning Goals for Core Courses (CPSC 110, 111, 121, 210, 211, 213, 221, 310, 313, 320): A comprehensive set of learning goals (both course-level and topic-level) has been developed for most 1st and 2nd year core courses. Most of these courses use LGs regularly to some extent in class (e.g., many 111 instructors now show the LGs associated with each unit as they lecture on the unit). CPSC 313 also has rough draft LGs. Draft course-level learning goals for the new CPSC 110 and CPSC 210 have been completed.

Computer Science Learning Goals

Learning goals from 5 UBC Computer Science courses (prepared by Beth Simon and numerous UBC CS faculty).

CPSC 260: Object-Oriented Program Design: Don Acton and Ben Yu investigated the correlation of student performance with different components of this course.

Attitudinal Surveys: Survey instruments have been developed for CPSC 101, 111, 221, 317, 320, 404, and APSC 160. These instruments will facilitate the tracking of students' attitudes about the curriculum, their interest in Computer Science, and their expectations throughout their undergraduate years.

Peer Evaluation Primer for CS Instruction: Jessica Dawson prepared two documents on the use of peer evaluation in the context of computer science instruction, based on a literature review and discussions with instructors in the department who have used it. The first document Introduction to Student Peer Review is a brief five-page overview of what peer review is as well as the advantages and challenges of implementing peer review; it includes an annotated bibliography. The 2nd document Resources and Guidance for Student Peer Review contains a detailed checklist of issues that should be considered when thinking about implementing peer review, as well as an overview of peer review software systems available as of fall 2014.

Research

PeerWise

Hassan Khosravi used the PeerWise system in APSC 160 and CPSC 259 and 304 in 2014–2016. PeerWise supports an online repository of multiple-choice questions that are created, answered, rated and discussed by students. This peer created and curated content was used for formative assessment by the students. The tool provides the instructor access to extensive analytics on student behaviours, and Hassan has downloaded this information for further analysis; for example, interaction graphs for students mediated by the questions. Hassan also developed a series of scripts which allow questions and answers to be extracted from the PeerWise system and used independently (for example, PeerWise could be used in a single offering and then an instructor curated subset of questions could be hosted locally in subsequent offerings); however, the benefits of this additional functionality was not sufficient to offset the disadvantages of losing access to the more comprehensive user interface provided by using the PeerWise tool directly, so its development has been suspended.

Computer Science Student Experience Project

Jessica Dawson began a research study to examine the outcomes and experiences of students in CS introductory courses, and in particular, to understand how these experiences may differ for difference students (for examples for CS majors and non-CS majors). In 2015 baseline data collection began via pre-post surveys in (CPSC 110 and CPSC 301) and student interviews and focus groups (CPSC 110). In collaboration with the course instructors, this data will be used to evaluate two new introductory CS courses (CPSC 100 and CPSC 103), each of which has a different target student audience than the existing introductory courses (CPSC 110, 301) the department offers. As part of this project, version 4 of the Computing Attitudes Survey (CAS) developed by Alison Tew is being used. The CAS has also been administered in CPSC 210 to evaluate changes in attitudes of CS-major students after their first and second programming course.

Poster (UBC Science Education Open House 2016): Student Experience in Introductory CS Courses

Foundations of Computing Concept Inventory

Steve Wolfman has been developing a set of related concept inventories to assess student progress through our Foundations of Computing Stream (CPSC 121, 221, 320). The process began with the high level learning goals from the course, and then analyzed data from exams, project submissions and think-aloud interviews. Preliminary results were presented at SIGCSE 2014, and a special session on concept inventories in CS was run at SIGCSE 2015. A draft multiple choice CI covering the basic material has been piloted on students at the start of CPSC 121 and at the end of CPSC 121 and 221. New questions are still being developed, and further offerings of the CI will be undertaken.

Poster (CWSEI EOY 2014): *Misconceptions and Concept Inventory Questions for Hash Tables and Binary Search Trees*

Paper (SIGCSE 2014): *Misconceptions and concept inventory questions for binary search trees and hash tables*

Talk by Steve Wolfman, Nov 2013: *Developing a Concept Inventory for the Foundations of Computing Course Sequence*

Poster (CWSEI EOY 2013): *Developing a Formative Assessment of Instruction for the Foundations of Computing Stream*

Mechanical TA Software for Peer Review

CPSC 430 has traditionally used essay questions on the midterm and final exams to judge students' ability to express concepts discussed in this non-technical class; however, because of the high cost of grading essays it was not feasible to provide opportunities during the term for students to practice such essays. Kevin Leyton-Brown has developed a software system called Mechanical TA (MTA) which allows students to submit brief essays through an online portal and then shares the essays out for peer review. While many such systems are available, the novel feature of MTA is that it divides the students into two groups based on the quality of their peer reviews. The "supervised" pool of students submit their essays and peer reviews as normal, but TAs provide grades on both their essays and their peer reviews. Students whose peer reviews are consistently good graduate into the "unsupervised" pool, where their essays are assigned the median score among the peer reviews and their peer reviews are assumed to be good; TAs need grade only a subset of spot-checks and appealed reviews. Not only does this system reduce TA workload, but the students have incentive to produce high-quality peer reviews and (with the recent addition of a pool of example essays for calibration) the means to improve their reviewing. A paper has appeared at the ACM Technical Symposium on Computer Science Education (SIGCSE) 2015: Wright, Thornton & Leyton-Brown, *Mechanical TA: Partially Automated High-Stakes Peer Grading*, doi: 10.1145/2676723.2677278. A survey was run at the end of the fall 2014 offering to explore student opinions about MTA, and the data is currently being analyzed. There is also ongoing work on the user interface (both student-side and instructor-side).

The MTA software has been tested with three other courses (CPSC 101, 110 and 301) for other types of assignment (images and code). It is not currently being used in the other courses for several reasons: the instructor interface is still rather fragile, the department's tech staff have concerns about the stability and security of the software, and it appears that the approach is most effective when used for many assignments (such as the weekly essays in 430) but not worth the overhead when used for a small number of assignments. Avenues to overcome the first two issues have been identified, but it remains to be seen whether there are other instructors and courses which can overcome the latter issue.

Computing Attitudes Survey (CAS)

Allison Elliot Tew developed and validated the Computing Attitudes Survey (CAS), a new assessment instrument to gauge student attitudes and perceptions about learning computer science. The CAS is based on the Colorado Learning Attitudes about Science Survey (CLASS) and extends that work to include specific computing issues such as debugging and data representation. The CAS will be applicable to a broad range of students and was piloted in three introductory courses in Fall of 2011: CPSC 101, CPSC 110 and APSC 160. Various versions of the survey were run in various classes in 2012, 2013 and 2014.

Paper: B. Dorn and A. E. Tew. *Empirical Validation and Application of the Computing Attitudes Survey* in Computer Science Education, 25(1), 2015, doi: 10.1080/08993408.2015.1014142 and version 4 of the survey has been released.

Poster (CWSEI EOY 2012): *Adapting the CLASS for Use in Computer Science*

Longitudinal Study of Student Learning

Allison Elliott Tew is designing a research study into assessment of student learning across a sequence of software design courses running from 1st to 4th year. Implementation details are currently under development. Initial meetings have been held with faculty who teach the courses. The first step is to

move from learning goals that focus on particular courses to learning goals that capture the progression from novice to expert over a sequence of courses.

Decomposition techniques in teaching proof by induction

Kim Voll applied a decomposition technique when teaching proof by induction in CPSC 121 in spring 2010. Ben Yu have interviewed students from both sections of the course taught last term using a think-aloud protocol developed in conjunction with Wendy Adams (UC). The results will be analyzed to determine if students taught with the decomposition technique demonstrate a stronger ability to perform proof by induction.

Just-in-time-teaching (JiTT) in APSC 160

Instructor has developed screencasts to introduce basic content to students. Students are expected to watch one or more screencasts before coming to class and are assessed on their grasp of this introductory material using clicker questions at the start of class. A collection of in-class problem sets has also been developed that will allow students to explore their understanding of more advanced content. We plan to conduct an assessment of retention of learning at the start of the follow-on course (CPSC 260) in the Fall and compare with results from last year where students had taken APSC 160 with more traditional instruction.

 *APSC 160: Student perceptions of online multimedia instruction with JiTT*

Just-in-time-teaching in CPSC 121

Instructor has identified a subset of learning goals called `pre-class` learning goals. These are goals that students are expected to meet before coming to class. On-line tests have been developed to assess student learning for those goals. A set of in-class problems have been developed that address more advanced learning goals. Comparative survey work indicates dramatic increases in percentages of students that use the textbook and find it useful to their learning.

 *Adaptation of JiTT in CPSC 121*

Just-in-time-teaching (JiTT) in CPSC 221

One instructor taught both sections in 2008/09 Winter term 2. Students in one section are seeing a JITT approach and the use of in-class activities involving peer instruction and discussion. Students in the other section are receiving more traditional instruction. Students in both sections are writing the same exams and completing the same homework assignments.

Study on the use of online collaborative multiple-choice question repository (PeerWise)

Donald Acton and Beth Simon conducted a study of the use of PeerWise by students in Computer Science 2nd and 4th year courses in 2007/08. A survey was given to the students about how they use PeerWise and whether they feel that submitting or answering questions helps them learn.

 Paul Denny's poster (April 2009): *PeerWise - Students sharing and evaluating their multiple choice questions*

Self-theories

Beth Simon and collaborators conducted a study of the impact of students' self-theories (based on the work of Carol Dweck) on success and persistence in beginning programming courses. Through 30 years of research, Dweck has shown that students who adopt a "growth mindset" – that is believe that through hard work and learning from errors they can improve their intelligence – perform better in a

variety of academic settings. A minimal intervention to encourage students to adopt a growth mindset was investigated in both CPSC 111 and APSC 160. This work was presented at the 2008 International Computing Education Research Workshop in Sydney, Australia. Read the paper:
Saying isn't necessarily believing: influencing self-theories in computing

Learning Goals study

Beth Simon in collaboration with Jared Taylor, STLF in Life Sciences, conducted a study of student and faculty perceptions of the usefulness of learning goals. Explicit use of learning goals has also spread to the Computer Science and Engineering department at UC San Diego, home institution of our first STLF. Read the paper (published in the Journal of College Science Teaching):
What is the Value of Course-Specific Learning Goals?

Parson's puzzles

Conducted a study in 2007/08 of a new type of exam question for assessing similar skills to code writing questions. Results have been published in the proceedings of the Fourth International Computing Education Research Workshop.

Learning Goals

Learning goals for core courses (CPSC 110, 111, 121, 210, 211, 213, 221, 310, 313, 320): A comprehensive set of learning goals (both course-level and topic-level) has been developed for most 1st and 2nd year core courses. Most of these courses use LGs regularly to some extent in class (e.g., many 111 instructors now show the LGs associated with each unit as they lecture on the unit). CPSC 313 also has rough draft LGs. Draft course-level learning goals for the new CPSC 110 and CPSC 210 have been completed.

Here are some good examples of actual learning goals:

Computer Science Learning Goals
Learning goals from 5 UBC Computer Science courses (prepared by Beth Simon and numerous UBC CS faculty).

Learning Goals Study: Beth Simon in collaboration with Jared Taylor, STLF in Life Sciences, conducted a study of student and faculty perceptions of the usefulness of learning goals. Read the paper (Published in the Journal of College Science Teaching, Nov/Dec 2009): What is the Value of Course-Specific Learning Goals?

Developing Learning Goals; The UBC Computer Science Department Experience:

Developing Learning Goals 101
How to Develop Learning Goals for an Established Course: The Computer Science Model. A document that Beth Simon put together that describes the successful process that the UBC Computer Science Department went through to establish learning goals in multiple courses.

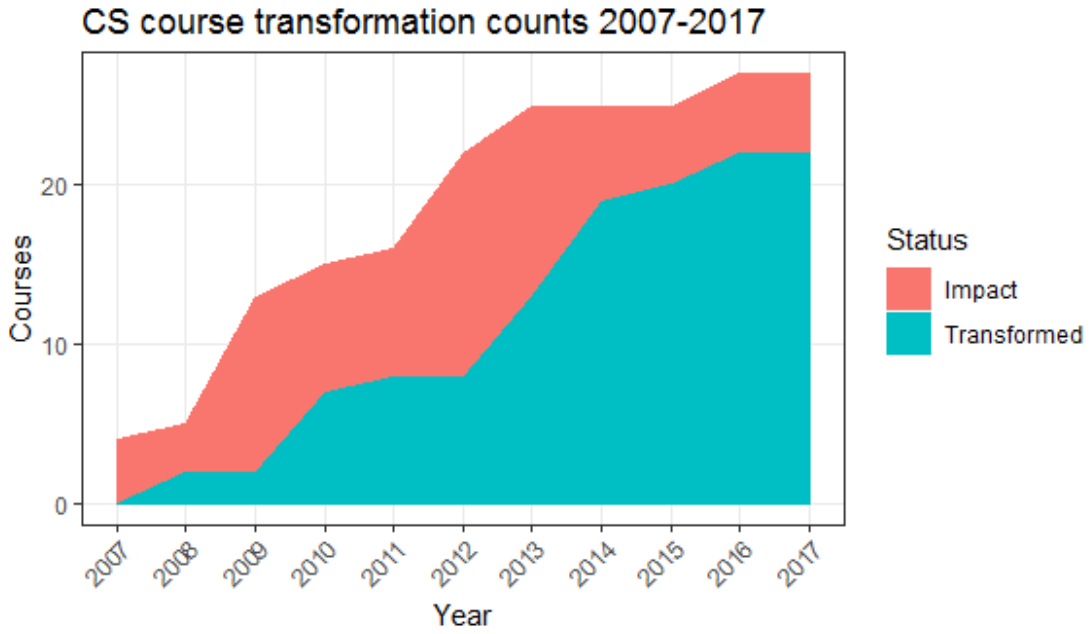
Tracking Changing Learning Goals - Steve Wolfman's experience (1-page version)
Tracking Changing Learning Goals - (3-page version)
Account written by Steve Wolfman on the trajectory they went through in developing learning goals for CPSC 101.

A Glimpse into the Process of Creating Learning Goals
Script of a role-play discussion between Steve Wolfman and STLF Beth Simon. It attempts to re-enact and give the feel for the process used in the UBC Computer Science Department to create learning goals for their courses. They often started by looking at an exam question previously used in the class – and used this to stimulate discussion and refinement of the actual goals faculty had for students taking the course. The discussion is modeled from an exam question used in CPSC 101 (a course for non-majors) in Summer 2006.

Impact

Transformed course counts

As of Spring 2018, we had 27 CPSC courses with CWSEI and/or Skylight influence:



Impact in terms of seats/registrations

We can look at this in terms of the **39 undergraduate CS courses offered in 2017** with LEC, LAB or DST as primary activity and excluding distance ed. sections. In terms of **seats/registrations** (with LEC, LAB or DST as primary activity; excludes distance ed. sections):

EFFECT	Seats. 2012	Seats.2013	Seats.2014	Seats.2015	Seats.2016	Seats.2017
Transformed	4908	5559	6298	7169	8351	9199
Impact	1035	1300	1619	2005	2121	2156
Other/None	1369	1276	1147	1145	1407	1659

EFFECT	Prop.Seats. 2012	Prop.Seats. 2013	Prop.Seats. 2014	Prop.Seats. 2015	Prop.Seats. 2016	Prop.Seats. 2017
Transformed	67.1%	68.3%	69.5%	69.5%	70.3%	70.7%
Impact	14.2%	16.0%	17.9%	19.4%	17.9%	16.6%
Other/None	18.7%	15.7%	12.7%	11.1%	11.8%	12.7%

Publications and Presentations

Paper	**Empirical Validation and Application of the Computing Attitudes Survey** Brian Dorn & Allison Elliott Tew (Univ. Nebraska & Computer Science, UBC) Computer Science Education, 25(1) (2015)	*restricted access link*
Paper	**Misconceptions and concept inventory questions for binary search trees and hash tables** Kuba Karpierz and Steven Wolfman (Computer Science, UBC) Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14), pp. 109-114 (2014)	*restricted access link*
Paper	**Effective closed labs in early CS courses: lessons from eight terms of action research** Elizabeth Patitsas and Steven Wolfman (Computer Science, UBC) Proceedings of the 43rd ACM technical symposium on Computer Science Education (SIGCSE '12), pp. 637-642 (2012)	*restricted access link*
Paper	**Circuits and logic in the lab: toward a coherent picture of computation** Elizabeth Patitsas, Kimberly Voll, Mark Crowley, & Steven Wolfman (Computer Science, UBC)	*restricted access link*

	Proceedings of the 15th Western Canadian Conference on Computing Education (WCCCE '10), Article No. 7 (2010)	
Paper	**Pedagogical transformations in the UBC CS science education initiative** Donald Acton, Kimberly Voll, Steven Wolfman, & Benjamin Yu (Computer Science, UBC) Proceedings of the 14th Western Canadian Conference on Computing Education (WCCCE '09), pp. 3-8 (2009)	*restricted access link*

Related Department Links

Computer Science Dept. home: https://www.cs.ubc.ca/