SFU OFFICE OF THE VICE-PROVOST, LEARNING & TEACHING

Strand Hall 3034
8888 University Drive
Burnaby B.C. Canada V5A 1S6

TEL + 1 778 782 5433
avplt@sfu.ca
SFU.CA/vpacademic/learnteach

MEMORANDUM

ATTENTION:	Senate
FROM:	Elizabeth Elle, Vice-Chair, Senate Committee on Undergraduate Studies
RE:	New Course Proposals
DATE:	June 2, 2023

For information:

Acting under delegated authority at its meeting of June 1, 2023 SCUS approved the following curriculum revisions effective Spring 2024.

a. Faculty of Applied Sciences *(SCUS 23-58)*

1. School of Computing Science

(i) New Course Proposal for CMPT 201-4, Systems Programming

Senators wishing to consult a more detailed report of curriculum revisions may do so on the Senate Docushare repository at https://docushare.sfu.ca/dsweb/View/Collection-12682.

SFU SENATE COMMITTEE ON UNDERGRADUATE STUDIES

NEW COURSE PROPOSAL

COURSE SUBJECT CMPT **NUMBER** 201

COURSE TITLE LONG — for Calendar/schedule, no more than 100 characters including spaces and punctuation

Systems Programming

COURSE TITLE SHORT — for enrollment/transcript, no more than 30 characters including spaces and punctuation

Systems Programming

CAMPUS where course will be normally taught: ☑ Burnaby ☑ Surrey ☐ Vancouver ☐ Great Northern Way ☐ Off campus

COURSE DESCRIPTION — 50 words max. Attach a course outline. Don't include WQB or prerequisites info in this description box.

An introduction to a UNIX-like application-OS interface from a programmer's perspective. Introduces operating systems and their interfaces for user-level programs. Students learn how to programmatically interact with an OS efficiently, correctly, and securely. Topics include: command-line tools, programming with memory, processes, threads, IPC, as well as basics of OS security.

REPEAT FOR CREDIT ☐ YES ☑ NO Total completions allowed ☐ Within a term? ☐ YES ☐ NO

LIBRARY RESOURCES

NOTE: Senate has approved (S.93-11) that no new course should be approved by Senate until funding has been committed for necessary library materials. Each new course proposal must be accompanied by the email that serves as proof of assessment. For more information, please visit www.lib.sfu.ca/about/overview/collections/course-assessments.

RATIONALE FOR INTRODUCTION OF THIS COURSE

A systems programming course has become a required course in leading CS programs. This is because (i) systems programming is considered one of the foundational skills to learn in CS, (ii) it serves as an advanced programming course where students get a chance to practice their programming skills further, and (iii) it prepares students better for numerous upper-level courses, such as operating systems, distributed systems, networking, databases, systems security, programming languages, software engineering, and computer architecture. However, the current curriculum for the School of Computing Science does not have a dedicated systems programming course. Instead, systems programming topics are combined with operating systems topics and taught in a single course, CMPT 300 Operating Systems. Because of that, students are not exposed sufficiently to either systems programming or operating systems.

The proposed course aims to provide a more thorough introduction to systems programming. It will cover the systems programming topics that CMPT 300 currently covers and expand the coverage to other systems programming topics that CMPT 300 does not currently cover. Upon taking the proposed course, students will have a deeper understanding of systems programming.

Since the proposed course will partially cover what CMPT 300 currently covers, we are also submitting a separate course revision plan for CMPT 300 that removes systems programming topics and adds operating systems topics to CMPT 300. The revised CMPT 300 aims to provide a more thorough introduction to operating systems. It will have the proposed course as a prerequisite.

SCHEDULING AND ENROLLMENT INFORMATION

Effective term and year (e.g. FALL 2016) Spring 2024

Term in which course will typically be offered ☑ Spring ☐ Summer ☑ Fall

Other *(describe)*

Will this be a required or elective course in the curriculum? ☑ Required ☐ Elective

What is the probable enrollment when offered? Estimate: 200

UNITS

Indicate number of units: 4

Indicate no. of contact hours: 3 Lecture ☐ Seminar ☐ Tutorial 1 Lab ☐ Other; explain below

OTHER

FACULTY

Which of your present CFL faculty have the expertise to offer this course?

Steve Ko
Tianzheng Wang
Harinder Khangura

WQB DESIGNATION

(attach approval from Curriculum Office)

PREREQUISITE AND / OR COREQUISITE

(CMPT 125 or CMPT 135) and MACM 101, both with a minimum grade of C-.

EQUIVALENT COURSES [For more information on equivalency, see Equivalency Statements under Information about Specific Course components.]

1. SEQUENTIAL COURSE [is not hard coded in the student information management system (SIMS).]

Students who have taken (*place relevant course(s) in the blank below (ex: STAT 100)*) **first** may not then take this course for further credit.

CMPT 300

2. ONE-WAY EQUIVALENCY [is not hard coded in SIMS.]

(*Place relevant course(s) in the blank below (ex: STAT 100)*) will be accepted in lieu of this course.

3. TWO-WAY EQUIVALENCY [is hard coded and enforced by SIMS.]

Students with credit for (*place relevant course(s) in the blank below (ex: STAT 100)*) may not take this course for further credit.

Does the partner academic unit agree that this is a two-way equivalency? ☐ YES ☐ NO

Please also have the partner academic unit submit a course change form to update the course equivalency for their course(s).

4. SPECIAL TOPICS PRECLUSION STATEMENT [is not hard coded in SIMS.]

FEES

Are there any proposed student fees associated with this course other than tuition fees? ☐ YES ☑ NO

COURSE – LEVEL EDUCATIONAL GOALS (OPTIONAL)

* Intro to systems programming (OS roles, syscalls, etc.)
* Programming tools (command-line tools, shell scripting, build systems, debugging, etc.)
* Programming with memory (memory layout, allocation, memory safety, memory hierarchy, etc.)
* Programming with processes (intro to processes, fork-exec-wait, signals, scheduling basics, etc.)
* Programming with threads (pthread, mutex, semaphore, deadlock & livelock, etc.)
* Programming with files (disk abstractions, permissions, etc.) and file systems (inodes, mounting, etc.)
* Programming with IPC (pipe, shmem, mmap, and domain sockets)
* Programming with sockets and RPC
* Security/protection and programming with crypto functions

Optional topics:
* Undefined behavior, reliability via redundancy (coding, replicas, etc.)
* Performance optimization (loop inefficiency, unnecessary function calls and memory references, profiling, and optimization)
* Kernel hooks (FUSE and eBPF)

RESOURCES

List any outstanding resource issues to be addressed prior to implementation: space, laboratory equipment, etc:

OTHER IMPLICATIONS

Final exam required ☑ YES ☐ NO

Criminal Record Check required ☐ YES ☑ NO

OVERLAP CHECK

Checking for overlap is the responsiblity of the Associate Dean.

Each new course proposal must have confirmation of an overlap check completed prior to submission to the Faculty Curriculum Committee.

Name of Originator

Steve Ko