CMPT 120 - D100 Intro.Cmpt.Sci/Programming I

Instructor(s): Victor Cheung

Calendar Objective/Description:
Intro.Cmpt.Sci/Programming I

Instructor's Objectives:
An elementary introduction to computing science and computer programming, suitable for students with little or no programming background. Students will learn fundamental concepts and terminology of computing science, acquire elementary skills for programming in a high-level language and be exposed to diverse fields within, and applications of computing science. Most if not all course components (lectures, assignments, and exams) might be in an online format.

* CMPT 120 PLACEMENT TEST – do you have enough computing knowledge to pass our placement test? Have you taken Computing Science courses in High School or elsewhere? Have you worked in a business or volunteered as a computer programmer? Have you written programs of 200-300 lines of code or more? IF YES you may NOT have to take CMPT 120 and could enroll directly into the next required courses – CMPT 125 and 127. Challenge yourself and take our placement test. https://courses.cs.sfu.ca/forms/cmpt-cmpt-120-placement-test/ IF NO – please go ahead and enroll in CMPT 120.

(Added on June 6th) Midterm Date: The CMPT 120 midterm exam will be held on Wednesday, October 27, 2021, from 6:30-8:20 PM Vancouver time. All students enrolled in this course are expected to be available during this time. If you experience a course conflict when registering for this course on goSFU due to the midterm, see http://www.sfu.ca/computing/current-students/undergraduate-students/faq.html#item8link for procedures to enroll in this course.

* As of 26 May, 2021, the university has announced its plan for approximately 70-80% of teaching in person in Fall 2021. It has also stated that: "not all courses will be delivered in person. The fall will be a transitional term. Deans, supported by the work of chairs and directors, will make final decisions about whether courses will be taught remotely or in person."

Please continue to check our course outline for further information. Should this course be taught remotely, students must have access to a computer with internet access, allowing the use of a conferencing system such as Zoom or BB Collaborate Ultra. Some components of the course will require synchronous (real-time) participation during the scheduled lecture and/or exam times. Visual proctoring may be required, subject to university approval.

"This course will be offered remotely."

Prerequisites:
see go.sfu.ca

Topics:
- History of computing science
- Algorithms and computational thinking
- Procedural programming in Python
- Data types and control structures
- Application areas within computing science
- Fundamental algorithms, including searching, sorting, basics of recursion
- Computability and complexity, introduction
- Basics of binary encoding

**Grading:**
There will be assignments and multiple quizzes/exams. A more detailed marking scheme will be provided in the first class of the semester.

Students must attain an overall passing grade on the weighted average of exams in the course in order to obtain a clear pass (C- or better).

**Required Books:**
Think Python - How to Think Like a Computer Scientist: Interactive Edition
https://runestone.academy/runestone/books/published/thinkcspy/index.html,

**Reference Books:**
Computer Science Illuminated, Nell Dale, John Lewis, Jones & Bartlett, 2012, 9781449672843
Starting out with Programming Logic and Design, Tony Gaddis, Pearson, 2015, 9780133985078

**Academic Honesty Statement:**
Academic honesty plays a key role in our efforts to maintain a high standard of academic excellence and integrity. Students are advised that ALL acts of intellectual dishonesty will be handled in accordance with the SFU Academic Honesty and Student Conduct Policies (http://www.sfu.ca/policies/gazette/student.html).