

CSc 116 - Fall 2018

Course Introduction

Mauricio Andres Rovira Galvez

University of Victoria
Department of Computer Science

September 5, 2018

CSc 116 is the follow-up to CSc 111 and provides an introduction to programming in a high-level language: C++. Topics include:

- Automated build processes,
- Test-driven development,
- Modularity and reusability,
- Data structures and their uses,
- Generic types and templates,
- Functions and lambda expressions,
- Introductory algorithm analysis,
- Introduction data structure design.

Undergraduate Advisor

- Irene Statham (cscadvisor@uvic.ca)
- Office: ECS 512
- Undergraduate Advising Hours:
 - MWF 10:00-12:00
 - TR: 13:30-15:30

Administrative Announcements

- If you are taking this course for the third (or greater) time, you must request, in **writing**, permission from the Chair of the Department and the Dean of the Faculty.
- If you have not met **all** the prerequisites for this course, you must receive department permission to stay in this class. If you do not receive permission, you will be **automatically dropped from the course** and a prerequisite drop will be recorded on your transcript
- In both of the above cases, you should visit the undergraduate advisor for more information.

Instructor Information

Lectures

- Mauricio A. Rovira Galvez (marovira@uvic.ca)
- Lectures: TWF 13:30-14:20 ECS 116
- Office: ECS 617

Labs

- Arthur Firmino, Juan Gallegos, Jon Grandfield
- TWF afternoons in ECS 258 (*Register now*)
- Labs may occasionally include graded exercises, but attendance is not otherwise taken.

Mauricio's Office Hours

- TF: 14:30-16:00, W 12:00-13:20
- I'm usually in ECS 648B outside my office hours. You may drop by, but please confirm with me beforehand as my schedule fluctuates.

This course will use the Department of Computer Science's conneX course management system for grades and assignments.

<https://connex.csc.uvic.ca/>

Lecture material, code samples, etc will be provided in a separate Github page. **Lecture notes are not comprehensive, and may not include all material covered in class. You are responsible for all material covered in lectures and labs, including material which is not posted afterwards.**

https://github.com/marovira/csc116_fall2018

If you are unable to log in to conneX, you may need to activate/reactivate your Computer Science account at

<https://accounts.csc.uvic.ca/>

Evaluation Scheme (1)

Assignments

Programming Assignments (approx. 5):	10%
--------------------------------------	-----

In-class Exercises

In-class Exercises:	9%
---------------------	----

Exams

Exam 1 (October 5, 2018):	27%
---------------------------	-----

Exam 2 (November 2, 2018):	27%
----------------------------	-----

Exam 1 (December Exam Period):	27%
--------------------------------	-----

Evaluation Scheme (2)

Exams:

Exams 1 and 2 will be 50 minutes long and held during the regular lecture time. Exam 3 will be **one hour** (not 3) long, during the final exam period. All three exams will be open-book and open-notes (but no electronic devices will be permitted).

To pass the course, you must meet all of the following conditions:

- Receive at least 50% overall,
- Receive a passing grade on **at least two** of the three exams.

If you do not meet the conditions above, your final grade will be an F (capped at 49%).

Evaluation Scheme (3)

Labs:

You must attend the lab section in which you are registered; make sure you are registered in a lab section before next week.

In-Class Exercises:

Exercises worth a cumulative total of 9% of the final grade will be given occasionally during lectures and labs, without necessarily being announced in advance. The distribution and weighting of the exercises differs between A01 and A02. You will receive credit for exercises completed in the lab/lecture sections in which you are registered.

Assignments:

Assignments will be primarily programming based, with an emphasis on requirement-driven and test-driven development.

Evaluation Scheme (4)

Missed Work:

Exceptions will be made for missed work (including late assignments) **only** in cases where an academic concession (with documentation) applies. Links to the relevant university policies are available from the posted official course outline.

Academic Integrity:

Plagiarism detection software will be used on assignment submissions where appropriate. Academic integrity violations will be reported to the department's academic integrity committee with recommendations for appropriate penalties. Links to the relevant university policies are available from the posted official course outline. Note that the university's guidelines clearly state that handing an assignment which is mostly or entirely plagiarized should result in a grade of F being given for the course.

Evaluation Scheme (5)

Acceptable Collaboration:

Computer Science and Mathematics are inherently collaborative disciplines, even if the stereotypes might say otherwise. You are encouraged to discuss all aspects of this course, including assignment questions, with your peers.

However, your actual assignment submissions must be your own work, and should be created independently (in your own words). Handing in the work of another student and claiming it as your own is plagiarism. Sharing your submission with another student (or the internet), even if it is not directly copied by anyone else, is also plagiarism.

Rule of thumb: Talk to your peers about assignments and collaborate on conceptual solutions, but **do not** look at each other's code (either over the shoulder or by sharing it electronically).