Faculty of Engineering
COURSE OUTLINE

SENG 475 & ECE 569A— Advanced Programming Techniques for Robust Efficient Computing

Summer 2018

University Calendar Entry for Course:

SENG 475: https://web.uvic.ca/calendar2018-05/CDs/SENG/475.html
ECE 569A: https://web.uvic.ca/calendar2018-05/CDs/ECE/569A.html

Instructor:

Dr. Michael Adams
Office: EOW 311
Email: mdadams@ece.uvic.ca
Web: https://www.ece.uvic.ca/~mdadams

Course Web Site:

Home Page: https://www.ece.uvic.ca/~mdadams/courses/cpp
Username: cpp
Password: as announced during lectures

Office Hours:

As announced during the lectures and posted on the course web site.

Lectures:

Sections: SENG 475 A01 (CRN 30708), SENG 475 A02 (CRN 30709), and ECE 569A A03 (CRN 30317)
Time/Location: Tuesdays, Wednesdays, and Fridays 11:30-12:20 in ECS 104

Tutorials:

Tutorial attendance is mandatory.

Section(s): SENG 475 B01 (CRN 30710) and ECE 569A B01 (CRN 30318)
Time/Location: Wednesdays 13:30–14:50 in ELW B238

Description and Objectives:

Advanced programming techniques for robust efficient computing are explored in the context of the C++ programming language. These techniques are applied to solving a variety of engineering-related problems.

Topics:

1. Software development tools
   - C++ compiler (i.e., GCC and Clang), linker, build tools (i.e., CMake), version control systems (i.e., Git), code sanitizers (e.g., ASan, UBSan, LSan, and TSan)
2. Very accelerated introduction to basic C++ (mostly covered via video lectures with some in-class lecture time devoted to more difficult material)
   - classes, templates, lambda expressions, C++ standard library
3. Data structures and algorithms
   - time and space complexity, asymptotic complexity, abstract data types (ADTs), containers, iterators, implementations of ADTs, trade offs between different implementations of ADTs, intrusive and non-intrusive data structures
4. Compile-time versus run-time computation
   - mechanisms for compile-time computation and their limitations, compile-time polymorphism
5. Error handling

■ exceptions, exception safety, exception guarantees, exception-safe resource management, resource acquisition is initialization (RAII) idiom

6. Memory management
    ■ memory allocation, memory-leak avoidance, smart pointers, move semantics, intrusive and nonintrusive containers, iterators
7. Computer arithmetic
    ■ finite-precision arithmetic, floating-point arithmetic, interval arithmetic, rational arithmetic, exact arithmetic, advantages and disadvantages of various approaches to arithmetic
8. Cache-efficient code
    ■ memory hierarchy and caches, cache-aware and cache-oblivious algorithms
9. Concurrency
    ■ multithreading, sequential consistency, data races, thread safety, threads, mutexes, condition variables
10. Applications
    ■ signal processing (e.g., fast Fourier transform, filter design), numerical analysis (e.g., interval arithmetic, matrix transposition, matrix multiplication), computational geometry (e.g., robust geometric predicates, Delaunay triangulations)

Learning Outcomes:

Upon completion of the course, students should be able to:
■ identify many of the factors that can impact the performance and robustness of code
■ select data structures and algorithms that are appropriate for solving a given problem and justify the choices made
■ demonstrate a detailed understanding of a variety of concepts related to: data structures, algorithms, compile-time computation, error handling, resource management, memory management, computer arithmetic, cache-efficient algorithms, and concurrency
■ develop software to meet a detailed set of specifications
■ recognize the importance of thoroughly testing code
■ demonstrate an intermediate-level competency in the C++ programming language
■ demonstrate a basic competency with the C++ standard library as well as several other libraries, such as Boost and CGAL
■ make effective use of the tools available in a typical C++ software development environment, such as: a compiler, linker, and tools for building software, version control, debugging, testing, and performance analysis

Required Texts/Materials:

The following references are required for the course:
Lecture Slides (print-on-demand book; available from University Bookstore):
M. D. Adams, *Lecture Slides for Programming in C++ — The C++ Language, Libraries, Tools, and Other Topics (Version: 2018-02-15)*, University of Victoria, Victoria, BC, Canada, 2018, ISBN 978-1-55058-624-4 (paperback).
The above lecture slides can be obtained in PDF format (free of charge) from https://www.ece.uvic.ca/~mdadams/cppbook.

Optional Texts/Materials:

For students wanting an additional reference on C++, the following book is recommended:
B. Stroustrup, *The C++ Programming Language, Fourth Edition*, Addison-Wesley, Upper Saddle River, NJ, USA, 2013, ISBN 978-0-321-56384-2 (paperback). On Amazon: https://www.amazon.ca/dp/0321563840.
This book is available from the University Bookstore as well as many online book retailers (e.g., Amazon).

Video Lectures:

Numerous video lectures are available on topics related to the course materials. Information about this video-lecture content can be found on the course web site.

Other Important Documents Available from the Course Web Site:

1. Course-Materials Bug-Bounty Program Handout (See section titled "Course-Materials Bug-Bounty Program")

2. Course-Materials Errata Handout (See section titled "Course-Materials Bug-Bounty Program")
3. Video Lectures Handout (See section titled "Video Lectures")
4. Assignment Handouts (See section titled "Assignments")
5. Project Handout (See section titled "Project")

Importance of Email:
Important course announcements are often sent to students via email. Therefore, students are responsible for checking their email regularly.

Lecture and Tutorial Attendance:
Students are required to attend all lectures and tutorials in the course. If a student is unable to attend a lecture/tutorial due to illness or some other reason, the student is solely responsible for any information missed (including any course-related announcements). Any student who does not attend lectures/tutorials regularly places themself at a **very serious disadvantage and a very greatly increased risk of failure** in the course.

Assessment:

SENG 475

50%	Assignments†
50%	Final Exam‡

ECE 569A

50%	Assignments†
10%	Project§
40%	Final Exam‡

SENG 475 and ECE 569A: Course-Materials Bug-Bounty Program Bonus*: 2% (of course mark)

***Note:** See the handout titled "Course-Materials Bug-Bounty Program" for more details.

†**Note:** The submission deadlines for assignments will be posted on the course web site. The assignments are to be done independently by each student. The instructor reserves the right to, at any time, question a student regarding any aspect of their submitted work in order to ensure that this work is the student's own. Furthermore, the instructor reserves the right to use plagiarism-detection software in the review and grading of student work. Assignments are not necessarily equally weighted. The relative weighting of assignments will be posted on the course web site.

‡**Note:** All exams are closed book. Calculators are not permitted in exams.

§**Note:** See the handout titled "Project" for more details.

Plagiarism, Cheating, and Other Forms of Academic Misconduct:
Plagiarism, cheating, and other forms of academic misconduct are taken very seriously by the instructor. Due to the nature of the course, the instructor is particularly sensitive to the issue of plagiarism in the context of programming assignments. The instructor and teaching assistants may, at any time, question a student regarding any aspect of their submitted work in order to ensure that this work is the student's own. Furthermore, the instructor and teaching assistants may employ plagiarism detection tools in the review and grading of some or all student work. **All incidents of plagiarism will be reported to the Department Chair.**

Maintenance of Marks:
Course marks are maintained (securely) on a server external to the University. Typically, a service like Google Docs/Sheets is used. Only the instructor and teaching assistants for the course have access to this information. If any student has concerns about this practice, they should contact the instructor as soon as possible during the first week of the term.

Percentage to Letter-Grade Conversion:
The final grade obtained from the above marking scheme for the purpose of GPA calculation will be based on the percentage-to-grade point conversion table as listed in the current Undergraduate/Graduate Calendar. See https://web.uvic.ca/calendar2018-05/undergrad/info/regulations/grading.html (in the Undergraduate Calendar) and https://web.uvic.ca/calendar2018-05/grad/academic-regulations/grading.html (in the Graduate Calendar).

Supplemental Exams:
There will be no supplemental examination for this course.

Note to Students (Regarding Handling Concerns About Course):
Students who have issues with the conduct of the course should discuss them with the instructor first. If these

discussions do not resolve the issue, then students should feel free to contact the Chair of the Department by email or the Chair's Secretary to set up an appointment.

Accommodation of Religious Observance:

See https://web.uvic.ca/calendar2018-05/undergrad/info/regulations/religious-observanc.html (in the Undergraduate Calendar) and https://web.uvic.ca/calendar2018-05/grad/registration/Registration.1.16.html (in the Graduate Calendar).

Policy on Inclusivity and Diversity:

See https://web.uvic.ca/calendar2018-05/general/policies.html.

Standards of Professional Behaviour:

You are advised to read the Faculty of Engineering document *Standards for Professional Behaviour*, which contains important information regarding conduct in courses, labs, and in the general use of facilities. See https://www.uvic.ca/engineering/assets/docs/professional-behaviour.pdf.

Cheating, plagiarism and other forms of academic fraud are taken very seriously by both the University and the Department. You should consult the entry in the current Undergraduate/Graduate Calendar for the UVic policy on academic integrity. See https://web.uvic.ca/calendar2018-05/undergrad/info/regulations/academic-integrity.html (in the Undergraduate Calendar) and https://web.uvic.ca/calendar2018-05/grad/academic-regulations/academic-integrity.html (in the Graduate Calendar).

Equality:

This course aims to provide equal opportunities and access for all students to enjoy the benefits and privileges of the class and its curriculum and to meet the syllabus requirements. Reasonable and appropriate accommodation will be made available to students with documented disabilities (physical, mental, learning) in order to give them the opportunity to successfully meet the essential requirements of the course. The accommodation will not alter academic standards or learning outcomes, although the student may be allowed to demonstrate knowledge and skills in a different way. It is not necessary for you to reveal your disability and/or confidential medical information to the course instructor. If you believe that you may require accommodation, the course instructor can provide you with information about confidential resources on campus that can assist you in arranging for appropriate accommodation. Alternatively, you may want to contact the Resource Centre for Students with a Disability located in the Campus Services Building. The University of Victoria is committed to promoting, providing, and protecting a positive, and supportive and safe learning and working environment for all its members.

Course Lecture Notes:

Unless otherwise noted, all course materials supplied to students in this course have been prepared by the instructor and are intended for use in this course only. These materials are not to be re-circulated digitally, whether by email or by uploading or copying to websites, or to others not enrolled in this course. Violation of this policy may in some cases constitute a breach of academic integrity as defined in the UVic Calendar.

Plagiarism Detection Tools:

Plagiarism detection software may be used to aid the instructor and/or teaching assistants in the review and grading of some or all student work.