

Video-Lecture Schedule

1 Preamble

The vast majority of the instructional content for the course is provided in the form of prerecorded video lectures. This document identifies all of the video lectures that a student must watch and provides a schedule indicating the minimum pace at which the material in these video lectures can be covered. It is important to understand that the provided schedule corresponds to the slowest pace at which a student could watch the video lectures and still have a reasonable expectation of a favorable outcome in the course. **It is absolutely critical that students not fall behind the minimum-paced schedule presented.** Furthermore, each student is encouraged, to whatever extent is possible, to cover the course material at a faster pace than that indicated in the provided schedule so that the student can better handle unanticipated events (such as illness or computer/network problems) that could cause unexpected delays on the work in this course.

Since it is likely to be very beneficial for the student to understand the rationale for the particular schedule provided, some comments in this regard are in order. The basic idea behind the scheduling is to have the student finish viewing the lecture content needed for an assignment at least two weeks before the submission deadline for the assignment. This ensures that a student has at least two weeks to work on each assignment, excluding Assignment 0 (which does not involve any coding) and Assignment 1 (which is intended as review). The assignments in the course are challenging. The course material is also challenging. It is critically important that the student watch the video lectures far enough in advance that they have sufficient time to ask questions about the lecture material and gain a reasonable understanding of it before the time at which they need to use their knowledge of the material for an assignment.

2 Schedule

The video lectures that comprise the lecture content for the course are listed below along with the time period when various video lectures should be watched. The designators CVL, BR, BM, and CWS are used to indicate where the link or URL for a video can be found, and have the following meanings:

- CVL: on the SENG 475 2019-05 Video-Lecture Handout (which is available from the *Video Lectures* section of the course web site)
- BR: in the course textbook in the appendix titled “Video Lectures” under the section titled “Rudimentary C++”
- BM: in the course textbook in the appendix titled “Video Lectures” under the section titled “Miscellaneous Video Presentations”
- CWS: on the course web site under the section titled “Video Lectures”

The items in the list below are **not all strictly sequential**. That is, some items have overlapping time periods.

1. **[Lecture 1] Wed May 3 (2023)**
information explaining how to effectively utilize the video lectures
 - (a) CWS — Course Video-Lecture Information Package
2. **[Lectures 1–3] Wed May 3 – Tue May 9 (2023)**
software tools; prerequisite knowledge for Assignment 0 (`cpp_tools`)
 - (a) BR — Getting Started — Compiling and Linking
 - (b) BR — Version Control — Introduction
 - (c) BR — Git — Introduction
 - (d) BR — Git — Demonstration
 - (e) BR — Build Systems — Introduction
 - (f) BR — CMake — Introduction
 - (g) BR — CMake — Examples

- (h) CWS — Accessing the SDE by Remote Login to the Lab Machines
 - (i) CWS — Accessing the SDE Using VM Software
 - (j) CWS — `assignment_precheck`
 - (k) BM — Assertions and CMake Build Type Demonstration
 - (l) BM — Address Sanitizer (ASan) Demonstration
 - (m) BM — Undefined Behavior Sanitizer (UBSan) Demonstration
 - (n) BM — Lcov Demonstration
3. **[Lectures 1–6] Wed May 3 – Tue May 16 (2023)**
basics; prerequisite knowledge for Assignment 1 (`cpp_basics`)
- (a) CVL — Lecture 2 (2019-05-08) — Algorithms and Data Structures
 - (b) CVL — Lecture 3 (2019-05-10) — Data Structures
 - (c) CVL — Lecture 4 (2019-05-14) — Data Structures, Some C++ Review (Const and Other Stuff)
 - (d) CVL — Lecture 5 (2019-05-15) — Some C++ Review (Const and Other Stuff)
 - (e) CVL — Lecture 6 (2019-05-17) — Some C++ Review (Const and Other Stuff), Compile-Time Computation
 - stopping at 41:02 [basics] The `constexpr` Qualifier for Functions
4. **[Lectures 6–8] Tue May 16 – Fri May 19 (2023)**
compile-time computation; prerequisite knowledge for Assignment 2 (`cpp_compile_time`)
- (a) CVL — Lecture 6 (2019-05-17) — Some C++ Review (Const and Other Stuff), Compile-Time Computation
 - starting at 41:02 [basics] The `constexpr` Qualifier for Functions
 - (b) CVL — Lecture 7 (2019-05-21) — Compile-Time Computation
 - (c) CVL — Lecture 8 (2019-05-22) — Compile-Time Computation, Temporary Objects
 - stopping at 42:16: [temporaries] Temporary Objects
5. **[Lectures 8–13] Fri May 19 – Wed May 31 (2023)**
move semantics; prerequisite knowledge for much of what follows
- (a) CVL — Lecture 8 (2019-05-22) — Compile-Time Computation, Temporary Objects
 - starting at 42:16: [temporaries] Temporary Objects
 - (b) CVL — Lecture 9 (2019-05-24) — Temporary Objects, Moving/Copying, Value Categories
 - (c) CVL — Lecture 10 (2019-05-28) — Value Categories, Moving/Copying
 - (d) CVL — Lecture 11 (2019-05-29) — Copy Elision
 - (e) CVL — Lecture 12 (2019-05-31) — Copy Elision, Implicit Move
 - (f) CVL — Lecture 13 (2019-06-04) — Copy Elision, Implicit Move, Exceptions
 - stopping at 17:34: [exceptions] Exceptions
6. **[Lectures 13–17] Wed May 31 – Fri June 9 (2023)**
arithmetic; prerequisite knowledge for Assignment 3 (`cpp_arithmetic`)
- (a) CVL — Lecture 13 (2019-06-04) — Copy Elision, Implicit Move, Exceptions
 - starting at 17:34: [exceptions] Exceptions
 - (b) CVL — Lecture 14 (2019-06-05) — Exceptions
 - (c) CVL — Lecture 15 (2019-06-07) — Exceptions, Interval Arithmetic
 - (d) CVL — Lecture 16 (2019-06-11) — Interval Arithmetic, Geometric Predicates and Applications
 - (e) CVL — Lecture 17 (2019-06-12) — Geometric Predicates and Applications, Memory Management
 - stopping at 34:01: [memory management] Memory Management
 - (f) CVL — Extra (2019-06-16) — Meshlab/Geomview Demo
 - (g) for a student who plans to attempt Part C of Assignment 3 (`cpp_arithmetic`), the following **optional** videos might be helpful (as the `Triangulation` class provided in Part C utilizes the `halfedge` type from the `Polyhedron_3` class in the CGAL library):

- i. BR — CGAL — Introduction
- ii. BR — CGAL — Polygon Meshes

7. **[Lectures 17–22] Fri June 9 – Wed June 21 (2023)**

containers; prerequisite knowledge for Assignment 4 (`cpp_containers`)

- (a) CVL — Lecture 17 (2019-06-12) — Geometric Predicates and Applications, Memory Management
 - starting at 34:01: [memory management] Memory Management
- (b) CVL — Lecture 18 (2019-06-14) — Memory Management
- (c) CVL — Lecture 19 (2019-06-18) — Memory Management
- (d) CVL — Lecture 20 (2019-06-19) — Memory Management
- (e) CVL — Lecture 21 (2019-06-21) — Memory Management, Intrusive Containers, Pointers to Members
- (f) CVL — Lecture 22 (2019-06-25) — Pointers to Members, Intrusive Containers, Caches
 - stopping at 27:03: [cache] The Memory Latency Problem

8. **Prior to Assignment 4** (`cpp_containers`)

more exceptions, resource management, and smart pointers; this material is not essential but is likely to be helpful for better understanding some aspects of Assignment 4

- (a) CVL — Lecture 31 (2019-07-17) — Concurrency, More Exceptions
 - starting at 03:44: [exceptions] Resource Management
- (b) CVL — Lecture 32 (2019-07-19) — Smart Pointers
- (c) CVL — Lecture 33 (2019-07-23) — Smart Pointers, Vectorization
 - stopping at 32:58 [vectorization] Vector Processing

9. **[Lectures 22–25] Wed June 21 – Wed June 28 (2023)**

cache-efficient coding; prerequisite knowledge for Assignment 5 (`cpp_cache`)

- (a) CVL — Lecture 22 (2019-06-25) — Pointers to Members, Intrusive Containers, Caches
 - starting at 27:03: [cache] The Memory Latency Problem
- (b) CVL — Lecture 23 (2019-06-26) — Caches, Cache-Efficient Algorithms
- (c) CVL — Lecture 24 (2019-06-28) — Cache-Efficient Algorithms
- (d) CVL — Lecture 25 (2019-07-03) — Cache-Efficient Algorithms, Concurrency
 - stopping at 37:40: [concurrency] Processors

10. **[Lectures 25–31] Wed June 28 – Fri July 14 (2023)**

concurrency; prerequisite knowledge for Assignment 6 (`cpp_concurrency`)

- (a) CVL — Lecture 25 (2019-07-03) — Cache-Efficient Algorithms, Concurrency
 - starting at 37:40: [concurrency] Processors
- (b) CVL — Lecture 26 (2019-07-05) — Concurrency
- (c) CVL — Lecture 27 (2019-07-09) — Concurrency
- (d) CVL — Lecture 28 (2019-07-10) — Concurrency
- (e) CVL — Lecture 29 (2019-07-12) — Concurrency
- (f) CVL — Lecture 30 (2019-07-16) — Concurrency
- (g) CVL — Lecture 31 (2019-07-17) — Concurrency, More Exceptions
 - stopping at 03:44: [exceptions] Resource Management

3 Additional Remarks

In order to provide students with more time to work on the course project closer to the end of the lecture schedule, the unit on vectorization (which has been included in past offerings of the course) has been omitted. Since vectorization is extremely important in modern computing systems, having knowledge of vectorization can be quite beneficial when

looking for jobs. Therefore, students are encouraged to watch the following videos sometime before looking for employment upon graduation:

1. vectorization; not part of course; only to help make students more marketable to employers
 - (a) CVL — Lecture 33 — Lecture 33 (2019-07-23) — Smart Pointers, Vectorization
 - starting at 32:58 [vectorization] Vector Processing
 - (b) CVL — Lecture 34 — Lecture 34 (2019-07-24) — Vectorization
 - (c) CVL — Lecture 35 — Lecture 35 (2019-07-26) — Vectorization
 - (d) CVL — Lecture 36 — Lecture 36 (2019-07-30) — Vectorization

Any parts of the video lectures from SENG 475 in the 2019-05 term that deal with course administrative issues (such as student evaluation/assessment, schedules/deadlines, assignments, projects, and so on) should be ignored as they most likely do not apply to the current offering of the course. For example, most of the content of the following lectures falls into the category of dealing with course administrative issues:

- CVL — Lecture 1 (2019-05-07) — Course Introduction
- CVL — Extra (2019-07-25) — Preliminary Information for Final Exam
- CVL — Lecture 37 (2019-07-31) — Final Course Wrap-Up