

Course Syllabus (Fall 2020)

[CS 225] Advanced C/C++

1. Course Information

1.1. General Information

Course full title:	[CS 225] Advanced C/C++
Pre-requisite(s):	[CS 170] High-level Programming II: The C++ Programming Language
Co-requisite(s):	Nil
Result type:	Grade, 3 credits
Web page:	Visit the section page at the new Moodle server: https://distance3.sg.digipen.edu/2020sg-fall/ <i>The website is accessible via student's DigiPen Singapore login credential.</i>

1.2. Description

This course builds on the foundation created in the first two high-level programming courses (CS 120/CS 170). It presents advanced topics of the C/C++ programming language in greater detail. Such topics include advanced pointer manipulation, utilizing multi-dimensional arrays, complex declarations, and standard library functions. Advanced C++ topics include class and function templates, operator overloading, multiple inheritance, runtime type information, the standard template library, and performance issues.

1.3. Objectives and Learning Outcomes

After successfully completing this course, students should:

- ✓ Have a deep understanding of C++ programming language design.
- ✓ Be able to create and compile a project solution from the command line using GNU C++ or Microsoft Visual C++ compiler.
- ✓ Know C/C++ expression evaluation algorithm and be able to write own complex declarations.
- ✓ Understand C/C++ conversion rules.
- ✓ Understand and apply move semantics, perfect forwarding and universal references.
- ✓ Understand and apply various specific design patterns – virtual constructors, type erasure, etc.
- ✓ Be able to use aggregation and inheritance to create new classes.
- ✓ Have a good understanding of function and class templates.
- ✓ Be able to use STL containers and generic algorithms to solve complex problems.

2. Course Organization

2.1. Lecture Schedule

Section A:	Tuesdays	11:30am – 1:10pm	(SIT@SP, Lecture Theatre 6B)
	Thursdays	11:30am – 1:10pm	(SIT@SP, Van Gogh)
Section B:	Tuesdays	1:30pm – 3:10pm	(SIT@SP, Lecture Theatre 4B)
	Thursdays	1:30pm – 3:10pm	(SIT@SP, Lecture Theatre 4B)

Take note that the lectures for both sections will be conducted in a live format in a classroom setting.

2.2. Instructor

Lecturer: Sławomir “Swavek” Włodkowski, M.Sc.

Consultation hours: (by appointment only, typically within the hours shown below)

Tuesdays 3:30pm – 5:30pm

Thursdays 3:30pm – 5:30pm

Contact: swavek.wlodkowski@digipen.edu

(include “[cs225f20-a]” or “[cs225f20-b]” prefix in the subject)

3. Learning Resources

3.1. Required Textbook

- Scott Meyers, *Effective modern C++: 42 specific ways to improve your use of C++11 and C++14*, O’Reilly Media, ISBN: 978-1491903995.

3.2. Recommended Reading

- Stanley B. Lippman, Josee Lajoie, Barbara E. Moo, *C++ Primer*, 5th Edition, Addison-Wesley; ISBN: 978-0321714114.
- Bjarne Stroustrup, *The C++ Programming Language*, 4th Edition, Addison-Wesley; ISBN: 978-0321563842.
- Bjarne Stroustrup, *Programming: Principles and Practice Using C++*, 2nd Edition, Addison-Wesley; ISBN: 978-0321992789.
- Scott Meyers, *Effective modern C++: 55 specific ways to improve your programs and designs*, 3rd Edition; O’Reilly Media, ISBN: 978-0321334879.
- Scott Meyers, *More effective C++: 35 new ways to improve your programs and designs*, 3rd Edition; O’Reilly Media, ISBN: 978-0201633719.
- Nicolai M. Josuttis, *The C++ Standard Library: A Tutorial and Reference*, 2nd Edition, Addison-Wesley; ISBN: 978-0321623218.
- Brian W. Kernigan and Dennis M. Ritchie, *The C programming language*, 2nd Edition, Prentice Hall; ISBN: 978-0131103627.
- Phillip J. Plauger, *The Standard C Library*, Prentice Hall; ISBN: 978-0131315099.
- Kenneth Reek, *Pointers on C*, Pearson; ISBN: 978-0673999863.

4. Course Outline

4.1. Schedule

Week	Week Start/Date	Topic	
1	7th Sept 2020	Fall trimester classes begin	Advanced Object-Oriented Programming
	7 th Sept 2020	Overview of programming paradigms	
		Layout of objects in memory	
		Aggregation, composition, inheritance; "has-a" and "is-a" idioms	
2	14 th Sept 2020	Overloading and overriding member functions	
		Non-virtual multiple inheritance, virtual inheritance	
		Virtual method tables	
	20th Sept 2020	Deadline for course withdrawal without academic penalty	
3	21 st Sept 2020	Virtualizing constructors	
		Factory method design pattern, prototype design pattern	
		Covariant return types	
		Dynamic dispatch; virtualizing non-member functions	
		Non-virtual interface idiom; template method design pattern	
4	28 th Sept 2020	S.O.L.I.D. design principles; design via interfaces	Memory
		Review of function templates and class templates	
		Template function overloading and specialization	
5	5 th Oct 2020	L-values, pr-values, x-values; l-value and r-value references	
		Move semantics, understanding <code>std::move</code> , <code>std::swap</code>	
		Template argument deduction, universal references	
		Understanding the perfect forwarding	
6	12 th Oct 2020	Type erasure pattern (TEP) and adapter design pattern	
	15th Oct 2020	Mid-term test	
7	19th Oct 2020	Recess week (no classes)	
	25th Oct 2020	Deadline for course withdrawal with grade "W"	
8	26 th Oct 2020	Bit manipulation, bit fields, unions	Standard Library
		Review of pointers and casts, complex declarations	
		Review of exceptions	
		Exception safety and RAII idiom	
		Smart pointers	
9	2 nd Nov 2020	Overloading new and delete operators	
		Placement new, explicit destructor calls; allocator classes	
		Variadic templates	
10	9 th Nov 2020	Sequences and SFINAE (substitution failure is not an error)	
		Callable objects, lambda expressions, trailing return types	
		Understanding <code>std::function</code>	
11	16 th Nov 2020	Understanding <code>std::tuple</code>	Standard Library
		Flat and recursive implementation	
		Additional template types and structured bindings	
12	23 rd Nov 2020	Review of containers, iterators, algorithms	
		Functional programming: predicates, adapters, binders	
		Callable objects (functors, lambda) with generic algorithms	
		Transforming and processing data in containers, inserters	
13	30 th Nov 2020	Course content review	
14/15	7th Dec 2020	Final test – exact date to be advised	
16	21st Dec 2020	Final grades due	

Above is a list of topics that will be covered this trimester. Depending on time, some minor topics may be added or skipped. Please note that this is a tentative organization of the course and it may change.

4.2. Format

The lectures are conducted twice a week. There are no laboratory classes accompanying this course.

A trimester is divided into halves, each lasting 6 weeks. At the last lecture of the initial 6 weeks of the trimester the mid-term test will be held. It is followed by 1 week of a scheduled recess break. The subsequent 6 weeks of education are followed by the final test in weeks 14 or 15.

4.3. Hardware and Software Requirements

To successfully participate in the course, students should ensure they have access to this software:

- GNU C++ Compiler (g++) 9.3.0 or newer.
- GNU Debugger (gdb) 8.3.0 or newer.

During this trimester all assignments will be graded automatically using Linux-based g++ compilation environment supported by the new version of Moodle that is available for the course. This will allow for receiving instant feedback on compilation errors and passing automated tests.

Compatibility of code with Visual Studio VC++ or clang++ compilers will not be included as a part of the submission evaluation process, but it is important for students to learn how to use GNU g++, Microsoft Visual Studio C++ compiler, and clang++ compilers, and test created code before submission.

5. Grading Policy

5.1. Grade Components

The composition of grades is as follows:

- | | |
|---------------------|-----|
| • Assignments (~9): | 45% |
| • Quizzes (~4): | 20% |
| • Mid-term test: | 15% |
| • Final test: | 20% |

5.2. Rubrics and Assessment

During the trimester there will be two tests and a few quizzes. Tests will be offered in a pen-and-paper format, while quizzes may be offered in a pen-and-paper format during lectures or online as homework.

In addition, there will be about 9 programming assignments to work on outside the class. Programming assignments will use the C++ programming language. More specifically, all programs must adhere to Standard C++17¹, which is what this course is about. Assignments will consist of a program specification that describes the functionality to be implemented, and it may include a list of classes or functions that you must defined to complete the assignment.

Assignment submissions will be evaluated and graded using 64-bit version of the g++ compiler available through Moodle. All submissions will be graded automatically upon upload. The detailed submission guidelines will be published alongside the first assignment specification.

¹ <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2017/n4640.pdf>

5.3. Letter Grades

Graded activities will result in one of the following letter grades defined with the qualitative statements:

Rank	Qualitative Statement	Mistakes	Grade
Excellent	The solution meets all the requirements. It is of excellent quality, well-structured and thoughtful, and takes into consideration good practices; it is free of mistakes. Well done!	None or trivial minor	A
		Accidental minor	A-
Good	The solution meets all but a few non-essential requirements or good practices. It shows minor need for improvement, but generally it is of good quality and demonstrates solid understanding of the subject; it is free of critical and major mistakes.	A few minor	B+
		Some minor	B
		Several minor, accidental major	B-
Fair	The solution meets nearly all essential requirements. It does not follow good practices, shows some need for improvement, but it is of fair, acceptable quality and demonstrates understanding of the subject with minor gaps; it is free of critical mistakes.	A few major	C+
		Some major	C
		Several major, accidental critical	C-
Poor	The solution occasionally does not meet essential requirements but can be tested and reviewed for grading. It is of insufficient quality, reveals gaps in understanding of the subject or bypasses some of the requirements by using more naïve techniques.	A few critical	D
Failure	The solution is not provided in time, it cannot be tested or does not show a potential to work. It is of low quality, reveals large gaps or misunderstanding of the subject; contains critical or excessive number of major mistakes.	Some critical	F
	Solutions infringing on academic integrity, e.g. cases of plagiarism, will receive this grade regardless of quality.		

Individual graded activities may introduce more specific criteria or quantitative statements to allow for more clear and objective assessment.

The final grade for the entire course will also be expressed as a letter grade. Take note of the following:

Fair (C-) is the minimum grade required for undergraduate students to earn credit in core courses.

Poor (D) is the minimum grade required for undergraduate students to earn credit in non-core courses.

Adjusted pass requirements will be published separately for the students of the joint degree programs.

5.4. Borderline Grades

The instructor reserves the right to raise a grade of a student with performance rated closely between two grades. For example, at the instructor's discretion a student with a score between A- and A may receive A, even if the actual calculations would put them in the upper areas of A-.

The instructor reserves the right to offer re-submission or re-assessment to a student close to a borderline passing grade who stands reasonable chances of passing the course given this opportunity.

5.5. Late Submission

Programming Assignments

Each programming assignment is due by the date and time indicated when the assignment is published.

All students are required to submit their solution by the due date. A student who did not submit their solution by the due date without a valid reason will receive an “F” grade. Students must notify the instructor and request an extension of the due date before the due date; requests after the due date will be granted extension only in extraordinary circumstances at the instructor’s discretion.

Quizzes:

Each quiz is due immediately by the end of the activity.

A student who was absent during the quiz is not expected to take it as there will be no make-up quizzes; their total score of the quizzes will represent the mean grade over fewer grades. A student who was present during the quiz class but did not submit their paper will receive an “F” grade.

Tests:

Each test is due immediately by the end of the activity.

All students are required to submit their papers by the end of the activity. A student who did not submit their paper on time without a valid reason will receive an “F” grade. Students must notify instructors in advance if they are unable to take a test. Requests to take a make-up test will be granted only in extraordinary circumstances.

Students may request for an extension should they provide valid reasons to justify their case. This will be handled on a case-by-case basis by the instructor. Request for extensions after the deadline will not be accepted except in extraordinary cases.

5.6. Grade Appeal

If a student is dissatisfied with a grade, they should firstly raise up the matter to the instructor. If the student is still not satisfied, they can do a formal grade appeal through the Registrar’s Office. A formal committee will be assembled to resolve the appeal. In such cases, students must fill-in the Grade Appeal Form available at Front Desk and submit it to the Registrar’s Office, along with supporting documents.

6. Attendance Policy

6.1. Mandatory Attendance

Attendance in the lectures is mandatory. You will be penalized for unexcused absences from class according to the following scale:

- Three (3) or more absences will result in a 10% reduction of your overall course grade.
- Six (6) or more absences will result in a 20% reduction of your overall course grade.
- Nine (9) or more absences will result in a 30% reduction of your overall course grade.
- Twelve (12) or more absences will result in your automatic failure in the course irrespective of your performance on tests, quizzes and assignments.

6.2. Medical Leave and Family Emergencies

Medical leave and family emergencies accompanied by appropriate documents will be the only exceptions to the mandatory attendance policy. Sleeping, studying for another class or exam, working on your game, etc., are not valid reasons for an absence.

6.3. Classroom Conduct

Students are expected to behave professionally at all times with regards to classroom conduct. To maintain a conducive learning environment during class, it is expected for students to:

- Be quiet during class while the instructor is talking and to keep a low noise level at all times. This is to ensure that everyone is able to listen to the lecture and discussion. Do not disturb others while doing class activities.
- Turn off your mobile phones and devices or put them on silent mode. This is to prevent unwanted interruptions due to phones ringing or vibrating. In emergency situations, get the instructor's approval to use the phone before the class.
- Reduce use of mobile phones during class where possible. Also, playing games on any device is strictly prohibited during class time. This is to reduce distractions for everyone around including the student him/herself. Penalties may be imposed if students are caught doing so.
- Keep the classrooms clean. Eating or drinking in class is strictly prohibited with the exception of bottled water. Do dispose of all wastes (such as used paper, eraser crumbs, empty bottles etc.) in the garbage bins located outside classrooms.
- Do not mistreat school equipment (such as computers, keyboards, mice, monitors etc.). There will be penalties given for such abuse cases and having broken equipment will cause inconvenience for everyone in school.

6.4. Health Advisory

Due to the COVID-19 related health crisis, at all times students must observe precautions to take care of own wellbeing and health of people around them. Students should properly observe social distancing measures, maintain good hygiene, and follow public safety advisory recommendations and restrictions issued by DigiPen Singapore, Singapore Institute of Technology or Singapore Government.

In particular, students must follow these restrictions:

- No more than 50 persons, including instructors, is allowed in a classroom at any given time.
- Students must leave a classroom immediately after the end of a class.
- Students must properly wear and not remove face masks during lectures.
- Students must avoid direct contact and concentrated sitting arrangement.

6.5. External Preparation

It is expected that the students in this class spend 6 hours on average per week for outside classroom activities through the trimester, including, but not limited to, homework, reading assignments, project implementation, group discussions, preparation of examinations, etc.

6.6. Last Withdrawal Date

In this trimester, the last date to withdraw without an academic penalty is 20th September 2020. The latest course withdrawal date is 25th October 2020.

In order to withdraw from a course, it is not sufficient simply to stop attending class or to inform the instructor. In accordance with the policy, contact your advisor or the Registrar Office to begin the withdrawal process. The last day for withdrawal from this course is cited in the official catalog.

7. Other Policies

7.1. Academic Integrity

DigiPen Institute of Technology Singapore stands for academic honesty, and professional integrity. As this course requires students to submit work for assessment, through this policy DigiPen Singapore would like to highlight the importance of the proper moral conduct and ethics.

Academic dishonesty in any form will not be tolerated. Cheating, copying, plagiarizing, or any other form of academic dishonesty (including doing someone else's individual assignments) will result in, at the extreme minimum, a grade "F" on the assignment in question, and could result in a failing grade in the entire course or even expulsion from DigiPen Singapore.

It is permissible to discuss exercises and assignments (but not solutions) with other students in the class, but the solutions must be recognizably your own. With the internet as a readily accessible source of information and help, students may feel that plagiarism is ambiguous, and thus be unable to determine what it constitutes. Here are some general guidelines to help make the distinction:

- Do **NOT** copy-paste any works online. Using works that are not yours is plagiarism.
- Do **NOT** ask online communities to solve your bugs and code issues by providing your code segments. Asking others to solve your issues is work not done by you, and thus it is plagiarism.
- You may learn from sources online, understand the workings, concepts, and implementing them again via **your own efforts**. Assume that you will be tested on the things you learn online, and if you are unable to answer the questions then you should not use said works.
- You may ask online communities about general problems, and use their insights **to work on your problem**.
- These applies to all sources on any medium, be it the internet, textbooks, friends or social media. It is the content that is important, not the medium they are on.
- The bottom line test is to ask yourself "**Did I work on this?**" If you did not, then you should not use it. Learn from it and work it out yourself.
- Do **NOT** show or share code that you submit for assessment and evaluation.

7.2. Disability Support Services

Students who have special needs or medical conditions and require formal accommodations in order to fully participate or effectively demonstrate learning in this class should contact the Student Life & Advising Office (studentlife.sg@digipen.edu) at the beginning of each trimester. A Student Life & Advising Officer will meet with the student privately to discuss how the accommodations will be implemented.