Modern C++ Design Patterns

CSD 2125

VERSION FOR TRIMESTER 1, AY 2021/22

This module profile may be subject to change during the trimester – the online version is the authoritative version.

1. General Module Information

1.1 General Information

Credit*	5 SIT credits
Pre-requisite(s) needed* (module code and name)	CSD1170 High-Level Programming 2
Pre-requisite(s) for other modules* (module code and name)	CSD2160 Computer Networks CSD2181 Data Structures CSD3115 Low-Level Programming CSD3182 Artificial Intelligence for Games CSD3130 Algorithm Analysis
Co-requisite(s)* (module code and name)	NIL
Module Owner (Institution)*	DigiPen Institute of Technology Singapore
Module Owner (Programme)*	BSCS RTIS BSCS IMGD
Cross Listing* (module code and name)	NIL
Result Type*	Grade
Module Coordinator	Name: Prasanna GHALI
	Phone:
	Email: pghali@digipen.edu
Module Co-Coordinator	N/A
First taught in*	N/A

1.2 Module Descriptor*

This module builds on the foundation created in the first two High-Level Programming modules (CSD1120/CSD1170). It presents advanced topics in 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 function and class templates, operator overloading, multiple inheritance, runtime type information, the Standard Library, and performance issues.

1.3 Module Changes in Response to Previous Feedback:

N/A

2. Module Learning Outcomes*

After successfully completing this course students should be able to:

1. **Apply** low-level programming techniques to write programs that minimize memory usage and/or runtime.
2. **Apply** modern C++ facilities to implement robust, efficient, and flexible user-defined types.
3. **Develop** robust programs using the C++ standard template library.
4. **Apply** static polymorphism via templates and dynamic polymorphism via inheritance and virtual functions to design and implement new classes.
5. **Evaluate** and justify use of object-oriented principles and design patterns for specified program features.

3. Module Content and Learning Activities

The following lists the topics and learning activities that are planned for this module and the amount of study hours associated with them.

3.1 Topics covered in the module

Topic	Details	Contact hours
Course administration	Syllabus; Course overview	0.5

Low-Level Programming 1	Bitwise operations: operators, bit manipulation, bit-fields; Unions; Scoped enumerations; Binary file I/O; Mixing C and C++ code; Memory: memory management using `std::allocator<T>`, overloading `new` and `delete` operators; performance impacts of memory; reducing memory fragmentation; implementing a smart pointer	3.5
Low-Level Programming 2	Unscrambling complex declarations; Pointers: array pointers, function pointers, double pointers, opaque pointers, callback functions, jump tables	4
Standard Template Library	Using STL containers (`std::set`, `std::map`, `std::unordered_set`, `std::unordered_map`), iterators (iterator categories, iterator adaptors), generic algorithms	4
Function Objects	Function objects: overloading call operator, lambdas and closures; Wrapping function objects with `std::function`; Creating new functions from old ones; C++20 ranges and views	4
Modern C++ 1	Uniform initialization; Initializer lists; Compile time constants: `constexpr`; Using type deduction: `auto` and `decltype`; Smart pointers: `shared_ptr`, `unique_ptr`, `weak_ptr`	4
Modern C++ 2	lvalue and rvalue references; Return and Named Return value optimization; Move semantics; `std::move`	2
Test	Mid-term test	2
Inheritance	Base and derived classes; Virtual functions; Dynamic polymorphism; Virtual pointer tables; Multiple inheritance; Factories: virtual constructors, virtualizing non-member functions, covariant return types	6
Templates	Review of function and class templates; Template argument deduction; Perfect forwarding; Variadic templates; Bridging static and dynamic polymorphism	6
Design Principles	Single responsibility principle; Open/closed principle; Liskov substitution principle; Interface segregation principle; Dependency inversion principle	4
Design Patterns	Adapter, Template method, Strategy, Composite, Iterator, Proxy, Factory, Singleton	6
Test	Final test	2
Total:		**48**

3.2 Learning Activities

Type of Activity	Details	Total study load [hrs]
Lectures	One-way delivery of the content following the pace indicated in the syllabus with minor participation of the students. 2x 2h lectures per week.	48
Assignments	This is the practical portion of the module. Students will independently work on assignments to translate the knowledge acquired into an actual working program that matches an assignment specification document with the detailed requirements.	36
Assignment consultations	Students will interact with instructors individually or in small groups in a mix of face-to-face and online sessions to obtain supervision and feedback on the programming assignments.	4
Preparatory work	Students will review material from class lectures, read handouts and recommended textbooks, and complete weekly worksheets. Projected time is 3 hours per week with a scheduled assignment	36
Midterm and final test preparation	Preparation for midterm test	4
Total:		**128**

External Preparation:
It is expected that the students in this class spend **7 hours** on average per week for outside classroom activities through the trimester, including, but not limited to, homework, assignments, group discussions, preparation of tests, etc.

A tentative schedule is listed below, and it is subject to change. Please follow the module details on the LMS for any updates.

Week	Activity Type	Topics	Duration [hrs]
1	Lectures	Course administration; Low-Level Programming 1: Bitwise operations, operators, bit manipulation, bit-fields; Unions; Scoped	4

Week	Activity Type	Topics	Duration [hrs]
		enumerations; Binary file I/O; Mixing C and C++ code; Memory management, memory management using `std::allocator<T>`, overloading `new` and `delete` operators; performance impacts of memory; reducing memory fragmentation; implementing a smart pointer	
	Preparatory work	This week’s lecture topics	3
	Assignments and projects	Assignment #1:	3
	Assignment consultations	Assignments #1	0.5
2	Lectures	Low-Level Programming 2: Unscrambling complex declarations; Pointers: array pointers, function pointers, double pointers, opaque pointers, callback functions, jump tables	4
	Preparatory work	This week’s lecture topics	3
	Assignments and projects	Assignment #2	3
	Assignment consultations	Assignment #2	0.5
3	Lectures	Standard Template Library: Using STL containers (`std::set`, `std::map`, `std::unordered_set`, `std::unordered_map`), iterators (iterator categories, iterator adaptors), generic algorithms	4
	Preparatory work	This week’s lecture topics	3
	Assignments and projects	Assignment #3	3
	Assignment consultations	Assignments #3	0.5
4	Lectures	Function Objects: overloading call operator, lambdas and closures; Wrapping function objects with `std::function`; Creating new functions from old ones; C++20 ranges and views	4
	Preparatory work	This week’s lecture topics	3

Week	Activity Type	Topics	Duration [hrs]
	Assignments and projects	Assignment #4	3
	Assignment consultations	Assignment #4	0.5
5	Lectures	Modern C++ 1: Uniform initialization; Initializer lists; Compile time constants: `constexpr`; Using type deduction: `auto` and `decltype`; Smart pointers: `shared_ptr`, `unique_ptr`, `weak_ptr`	4
	Preparatory work	This week's lecture topics	3
	Assignments and projects	Assignment #5	3
	Assignment consultations	Assignment #5	0.5
6	Lectures	Modern C++ 2: lvalue and rvalue references; Return and Named Return value optimization; Move semantics; `std::move` Midterm Test	4
	Preparatory work	Mid-term test preparations	7
	Assignment consultations		0
7		Break	
8	Lectures	Inheritance 1: Base and derived classes; Virtual functions; Dynamic polymorphism; Virtual pointer tables; Multiple inheritance	4
	Preparatory work	This week's lecture topics	3
	Assignments and projects	Assignment #6	3
	Assignment consultations	Assignment #6	0.5
9	Lectures	Inheritance 2: Factories: virtual constructors, virtualizing non-member functions, covariant return types Templates 1: Review of function and class templates; Template argument deduction; Perfect forwarding	4
	Preparatory work	This week's lecture topics	3
	Assignments and projects	Assignment #7	3

Week	Activity Type	Topics	Duration [hrs]
	Assignment consultations	Assignment #7	0.5
10	Lectures	Templates 2: Variadic templates; Bridging static and dynamic polymorphism Design Principles 1: Single responsibility principle; Open/closed principle; Liskov substitution principle	4
	Preparatory work	This week's lecture topics	3
	Assignments and projects	Assignment #8	3
	Assignment consultations	Assignment #8	0.5
11	Lectures	Design Principles 2: Interface segregation principle; Dependency inversion principle Design Patterns 1: Adapter, Template method, Strategy, Composite, Iterator, Proxy, Factory, Singleton, Observer	4
	Preparatory work	This week's lecture topics	3
	Assignments and projects	Assignment #9	3
	Assignment consultations	Assignment #9	0.5
12	Lectures	Design Patterns 2: Adapter, Template method, Strategy, Composite, Iterator, Proxy, Factory, Singleton, Observer	4
	Preparatory work	This week's lecture topics	3
	Assignments and projects	Assignment #10	3
	Assignment consultations	Assignment #10	0.5
13	Lectures	Review	4
	Preparatory work	This week's lecture topics	3
	Assignments and projects	Assignment #10	3
14/15	Final test	Assignment submission	2

4. Learning Resources

4.1 Required Resources

- N/A

4.2 Recommended Resources

- *C++ Primer*, 5th Edition, Stanley B. Lippman, Josee Lajoie, Barbara E. Moo. Published by Addison-Wesley Professional, ISBN-10: 0321714113 and ISBN-13: 978-0321714114
- *The C++ Programming Language*, 4th Edition, Bjarne Stroustrop. Published by Addison-Wesley Professional, ISBN-10: 0321563840 and ISBN-13: 978-0321563842.
- *The C++ Standard Library: A Tutorial and Reference*, 2nd Edition, Nicolai M. Josuttis. Published by Addison-Wesley Professional. ISBN-10: 032162315 and ISBN-13: 978-0321623218
- *C++ Templates: The Complete Guide*, 2nd Edition, David Vandevoorde and Nicolai M. Josuttis. Published by Addison-Wesley Professional. ISBN-10: 0321714121 and ISBN-13: 978-0321714121
- *Effective C++*, 3rd Edition, Scott Meyers, Published by Addison-Wesley Professional, ISBN-13: 978-0321334879
- *Effective Modern C++*, Scott Meyers, Published by O'Reilly Media, ISBN-13: 978-1491903995
- *The C Programming Language*, 2nd Edition, Brian W. Kernigan and Dennis M. Ritchie. Published by Prentice Hall. ISBN-10: 0131103628 and ISBN-13: 978-0131103627

4.3 Any Other Learning Resources

- Further learning resources will be published or referenced on the course's web page.

5. Assessment

5.1 Assessment Tasks*

This is a summary of the assessment in the module. For detailed information on each assessment, refer to the assignment specifications published on the course's web page. Dates for assessments are subjected to minor amendment and early notification will be made should changes be required.

Assessment Task	Weighting	Tentative week/due date
Programming Assignments Programming assignments comprises the practical portion of the course and aims to help students gain practical experience by applying principles	40%	Weekly

discussed in lectures. Students will gain experience in problem-solving, programming, debugging, principles of good program design, documentation, and testing.		
Quizzes Quizzes are assigned per topic and consist of problems that must be solved by students to obtain a review of the topic discussed during class lectures. They will roughly assigned once a week. They will be collected and graded. No late submissions will be accepted.	15%	Weekly
Midterm Test Closed-book, Closed-notes, No computers and calculators. All material covered in lectures, programming homework, and assignments. The exam will be composed of: • Short answer questions: multiple choice, true/false and fill-in-the-blank type questions. • Questions related to topics discussed in class. • Questions related to topics discussed in programming assignments.	20%	Week 6
Final Test Closed-book, Closed-notes, No computers and calculators. All material covered in lectures, programming homework, and assignments. The exam will be composed of: • Short answer questions: multiple choice, true/false and fill-in-the-blank type questions. • Questions related to topics discussed in class. • Questions related to topics discussed in programming assignments.	25%	Week 14/15

5.2 Pass Requirements

A “D” grade or above is considered as “Pass” and an “F” grade is considered as “Fail”. Students with grades “F” must repeat the module, while grades “D” or “D+” will be given the option to repeat the module. The maximum GPA for re-module will be capped at 2.0. Successful completion of the academic module is also based on fulfilment of any other requirement(s) set by the module coordinator and programme director.

Marks are final after endorsement by the Board of Examiners.

Do note that the criteria for acceptable standing in any given trimester is maintaining a minimum Cumulative Grade Point Average (CGPA) of 2.0. Refer to the SIT Academic Guide for further details.

5.3 Late Submission

Every assessment is due by the date and time indicated when the assignment is published. A student who did not submit their solution by the due date without a valid reason will receive an "F" grade. If you require an extension to an assessment because of health or extraordinary emergencies, such requests must be made ***prior to the assessment***. After the emergency has passed, you must provide documentation (for example, a medical certificate from a medical practitioner) supporting the extension. All other non-submissions by the due date will receive an "F" grade.

5.4 **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 semester. A Student Life & Advising Officer will meet with the student privately to discuss how the accommodations will be implemented.

Appendix A: Schedule

Lecture Schedule

Lecture 1	Tuesday	9 am ~ 11 am	Online
Lecture 2	Thursday	9 am ~ 11 am	Online

Contact Information

Instructor:	Prasanna Ghali
Consultation hours:	To be determined
Email:	pghali@digipen.edu (add **csd2125** in email subject field)

Appendix B: Attendance Policy

Attending lectures in this module is mandatory and it can impact the final grade in the module:

- Student more than 15 minutes late to class will be marked as absent for that entire class.
- Student may not leave the class early without the instructor's permission.
- Student absent from all classes without explanation for a period of 14 calendar days would receive a Letter of Warning from the Registrar's Office. The student must reply with the reasons of absence and resume attending classes within the next 14 calendar days to avoid being withdrawn from the Institute.
- To apply for your absences to be excused, please submit your documents (Medical Certificate, Reservist notice, etc.) to registrar.sg@digipen.edu within 7 calendar days of your return.
- Unexcused absences would result in the following penalty to your final grade.

Number of classes conducted per week in a module	Total number of sessions in 12 instructional weeks	1 letter grade down for the number of unexcused absences	2 letter grade down for the number of unexcused absences
1	12	2	4
2	24	4	8
3	42	6	12

- Example: A module was scheduled to meet once a week and a student had 2 unexcused absences. The student then attained a final grade of "A-" for the module, but his final letter grade was downgraded to "B+" due to the penalty.