



Course Syllabus

MEC 510: Object-Oriented Programming for Scientists and Engineers Spring 2025

Instructor	Amin Fakhari, Ph.D., Department of Mechanical Engineering
Office	165 Light Engineering, Stony Brook University
Office Hours	TuTh 11:00 AM – 12:00 PM (and, any other time by appointment)
Phone	(631) 632-8329
Email*	amin.fakhari@stonybrook.edu

* All non-personal course-related questions should be posted on Brightspace Discussions Forum (see section Tools below). Email should be used only for strictly personal issues. I will respond to your emails as soon as possible, however, please allow up to 48 hours for a response. Your SBU email must be used for all your communications.

Teaching Assistant	Penghao Dong (penghao.dong@stonybrook.edu)
Office Hours & Office	Only by Appointment

Course Detail

Title	MEC 510: Object-Oriented Programming for Scientists and Engineers
Credit	3
Lecture	Tue. 3:30 – 6:20 PM, Humanities 1023
Prerequisites	BS/BE in science or engineering.

Recommended References

- Paul Deitel, Harvey Deitel, *C++ How to Program*, 11th Edition, Pearson, 2024 [[Publisher](#), [Amazon](#)].
- Stanley Lippman, Josée Lajoie, Barbara Moo, *C++ Primer*, 5th Edition, Addison-Wesley Professional & Pearson, 2013 [[Publisher](#), [Amazon](#)].
- Bjarne Stroustrup, *Programming: Principles and Practice Using C++*, 3rd Edition, Addison-Wesley Professional & Pearson, 2024 [[Publisher](#), [Amazon](#)].
- Recommended Online Resources: [cppreference](#), [cplusplus](#), [learncpp](#).

Course Description

Practical introduction to C++ and object-oriented programming for a first programming course for scientists and engineers. Covers basics of application software development such as problem decomposition, structure charts, object modeling, class diagrams, incremental code building, and testing at a beginner's level. Features the concepts of Abstract Data Types (ADT), encapsulation, inheritance, composition, polymorphism, operator and function overloading besides studying Unified Modeling Language (UML) as a graphical representational design technique. The course follows the evolution of programming ideas from the use of a single function to the use of structural charts and functions to modularize and finally to the use of object-oriented programming.

Course Learning Objectives

- Know the basics of C++ programming and different operators.
- Understand the concept of Object-Oriented Programming (OOP).
- Use control statements (if, if...else, while, do...while, for, switch,...) and functions.
- Work with data structures (e.g., arrays, vectors, and sets), references, and pointers.
- Use classes to organize C++ programs and create user-defined data types.
- Understand operator overloading, inheritance, and polymorphism.
- Solve problems in scientific applications using OOP.

Tools

Brightspace: It is required that you use the [Brightspace](#) for this course. Brightspace is used for facilitation of communications between faculty and students, posting of the course materials, important announcements, and grades, and submission of assignments. You need to check your SBU email or Brightspace announcements regularly [[Android App](#), [iOS App](#)].

Brightspace Discussions Forum: By using the Discussions tool/forum in Brightspace, you can get help fast and efficiently from your classmates and the instructor. All non-personal course-related questions that might be of interest to other students should be posted (either anonymously or identified) on the Brightspace Discussions forum and not emailed to the instructor. Email should be used only for strictly personal problems or issues.

Note that this discussion forum is for additional learning and assistance. It is not the place for cyber-bullying, memes, grade complaints, concerns/comments/criticisms about the course, or in general, anything unrelated to the course material and student learning. Improper behavior will result in reporting of the individual's behavior to the Office of Student Conduct and Community Standards.

C++ IDE: There are many C++ IDEs (Integrated Development Environments) to write/compile your source code and you can choose among them. [Visual Studio Code](#) (VS Code) is a free, open-source, cross-platform IDE that you can [install on your computer](#). You can also use online IDEs like [OnlineGDB](#).

Homework Assignments and Final Project

- Homework assignments will be posted on Brightspace, and must be submitted through OnlineGDB. To submit your homework assignments, create a free account using your SBU email address and your official FULL NAME on [OnlineGDB](#) and enroll in the course (MEC 510 - Spring 2025) using the following link:

<https://onlinegdb.com/classroom/invite/cg3gOxZjg>

- If your code passes the tests on OnlineGDB, it does not necessarily mean that your code is written properly.
- You have up to 15 late days for use on any homework assignment throughout the semester, but no homework may be more than 5 days late. Once you used your budget of 15 late days for the semester, each late day will be assessed a 25% penalty on your grade for that assignment(s).

Examinations

Midterm Exam	Tuesday, Mar. 25, 2025
Final Exam/Project	Tuesday, May 13, 2025, 5:30 – 8:00 PM

- (a) Make-up exams are considered only for students who provide official documentation of a compelling reason (e.g., medical emergency) before, or within three days following the missing exam. There will be no make-up exams for reasons that can be within your control (e.g., pre-arranged travel or other engagements).
- (b) The exam dates are subject to change. Students will be notified in a timely manner of any changes.

Grading Policy

Homework	40%
Midterm Exam	30%
Final Exam/Project	30%

Grading Scale

A	[100, 90]%	A⁻	(90, 85]%	
B⁺	(85, 80]%	B	(80, 75]%	B⁻ (75, 70]%
C⁺	(70, 65]%	C	(65, 60]%	C⁻ (60, 55]%
F	(55, 0]%			

General Coding Rules

- Add a brief introductory description of the program at the beginning of your code in form of a comment.
- Add proper comments at different parts of the code to describe functions, classes, and input/output parameters.
- Choose self-descriptive names for variables, functions, and classes.
- Use proper indentation and line breaks to make the code structure clear.
- Avoid hard-coding (i.e., putting specific input values in the code).

Tentative Course Schedule

Week 1: Introduction, Input-Output, and Operators
Week 2: Classes, Objects, and Member Functions
Week 3: Classes, Objects, and Member Functions
Week 4: Control Statements
Week 5: Control Statements
Week 6: Functions and Recursion
Week 7: Functions and Recursion
Week 8: References and Pointers
Week 9: Data Structures
Week 10: namespace, enum, struct, Class Templates
Week 11: namespace, enum, struct, Class Templates
Week 12: Classes, Operator Overloading
Week 13: Inheritance, Polymorphism, and Virtual Functions

Syllabus Disclaimer

The instructor views the course syllabus as an educational understanding between the instructor and students. Every effort will be made to avoid changing the course schedule, materials, assignments, and deadlines, but the possibility exists that unforeseen events will make syllabus changes necessary. The instructor reserves the right to make changes to the syllabus as deemed necessary. Students will be notified in a timely manner of any syllabus changes via email or Brightspace announcements.

University Policies and Statements

Academic Integrity Statement

Each student must pursue his or her academic goals honestly and be personally accountable for all submitted work. Representing another person's work as your own is always wrong. Faculty is required to report any suspected instances of academic dishonesty to the Academic Judiciary. Faculty in the Health Sciences Center (School of Health Technology & Management, Nursing, Social Welfare, Dental Medicine) and School of Medicine are required to follow their school-specific procedures. For more comprehensive information on academic integrity, including categories of academic dishonesty please refer to the academic judiciary website at http://www.stonybrook.edu/commcms/academic_integrity/index.html.

Student Accessibility Support Center (SASC) Statement

If you have a physical, psychological, medical, or learning disability that may impact your course work, please contact the Student Accessibility Support Center, Stony Brook Union Suite 107, (631) 632-6748, or at sasc@stonybrook.edu. They will determine with you what accommodations are necessary and appropriate. All information and documentation is confidential. Students who require assistance during emergency evacuation are encouraged to discuss their needs with their professors and the Student Accessibility Support Center ([SASC](#)). For procedures and information go to [Evacuation Guide for People with Physical Disabilities](#) and search Fire Safety and Evacuation and Disabilities.

Critical Incident Management Statement

Stony Brook University expects students to respect the rights, privileges, and property of other people. Faculty are required to report to the Office of Student Conduct and Community Standards any disruptive behavior that interrupts their ability to teach, compromises the safety of the learning environment, or inhibits students' ability to learn. Faculty in the HSC Schools and the School of Medicine are required to follow their school-specific procedures. Further information about most academic matters can be found in the Undergraduate Bulletin, the Undergraduate Class Schedule, and the Faculty-Employee Handbook.

Copyright Statement

Course material accessed from Brightspace or the Stony Brook website is for the exclusive use of students who are currently enrolled in the course. Content from these systems cannot be reused or distributed without written permission of the instructor and/or the copyright holder. Duplication of materials protected by copyright, without permission of the copyright holder, is a violation of the Federal copyright law, as well as a violation of SUNY copyright policy.