

1. **Course Number and Title:** COMP.2010.102 Computing III
2. **Instructor's Name:** Robert Marceau, Ph.D. email: <mailto:rmarceau@cs.uml.edu>

Please include the course number and section in the subject of your email.

3. **Instructor's Office:** DAN-307
4. **Office Hours:** Due to COVID-19, this class is strictly on-line and physical meetings are not possible. I tend to keep early hours and read my email between 5:00 am and 6:00 pm. Email that arrives outside of this range may not receive a response until the following morning. I will attempt to respond to email that arrives within this time range within a few hours.
5. **Course Description:** This course presents an introduction to object-oriented software development using the C++ programming language. The main ideas are encapsulation, class hierarchy and inheritance, polymorphism, copy control, operator overloading, dynamic memory management, and templates. Additional topics include the Standard Template Library (containers and iterators), class design diagrams, and exception handling.
6. **Course Co-requisites:** COMP.2010R Computing III Lab.
7. **Course Prerequisites:** COMP.1020 Computing II.
8. **Course Category:** Required.
9. **ABET Student Outcomes:** Upon successful completion of the course, students will:

- (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline.

This course will introduce students to the fundamental techniques and practices of object oriented software design. The computational model studied will be based specifically on the C++ language, but will cover essential aspects of all modern object oriented programming languages including an introduction to object-oriented design and analysis, object decomposition, C++ language extensions such as class hierarchy and inheritance, and polymorphic object behavior that supports the object-oriented paradigm.

- (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

The presentation of the material will be strongly based on providing a motivation for existing practices and language features. New material will be introduced as a solution to a problem based on the services provided by available computing hardware. Throughout the course, computing will be presented in a framework of:

1. Understanding known tools and their applicability to desired results;
2. Reformulating the problem specification to require a more sophisticated solution, beyond these tools' limitations; and
3. Introducing new tools or facilities that can meet the new requirements.

In this way, computational problems are presented as admitting a wide range of solutions of varying degrees of complexity.

- (c) An ability to use current techniques, skills, and tools necessary for computing practice.

The techniques and skills covered in this course are fundamental to the discipline: good programming practices for programming style, documentation and modularity, and the use of an integrated development environment for code creation.

- (d) An ability to apply design and development principles in the construction of software systems of varying complexity.

The practical component of the course will entail implementing solutions to specific problems based on specification documents. As the student's repertoire of programming techniques expands, programming assignments will gradually allow more freedom in implementation details, requiring selection of the most appropriated tools and careful consideration of the trade-offs involved in selecting one approach over another.

10. Course Outcomes:

1. An understanding of terminology associated with computing systems. ABET (a)
Assessment: Exams and/or quizzes
 2. The ability to create structured modular programs ABET (b)
Assessment: Programming exercises and/or exams
 3. An understanding of the basics of procedural and object oriented programming in C++ including STL containers and algorithms, pointers vs. reference parameters, generic algorithms and data structures, copy control, smart pointers, inheritance, polymorphism, overloaded operators, virtual functions, and dynamic binding. ABET (a) and (b)
Assessment: Programming exercises and/or exams
 4. Recognition of the importance and proficiency in the use of good programming style. ABET (c) and (d)
Assessment: Programming exercises
 5. The ability to use the basic features of an Integrated Development Environment and the basic features of a Unix/Linux environment to create, debug, and execute programs in C++. ABET (c)
Assessment: Programming exercises
11. **Date, Time and location:** Mondays, Wednesdays and Fridays September 4, 2024 - December 20, 2024 9:00 - 9:50 via Zoom (Links are on Blackboard, in the Meetings folder). Recorded meetings are also available (click the "Cloud Recordings" button).
12. **Required Textbook:** Frank Vahid and Roman Lysecky (zyBooks) *Programming in C++ with zyLabs* (online) ISBN 979-8-203-34245-4

If you are using Financial Aid, you should purchase access via the campus bookstore. If you are using a credit card, you can purchase directly from <https://learn.zybooks.com/>:

- (a) Create an account
- (b) Enter zyBook code: UMLCOMP2010MarceauFall2024
- (c) ISBN: 979-8-203-34245-4
- (d) Subscribe (cost is \$64.00).

Some students found this useful for group lab work:
<http://collabedit.com/>

13. **Recommended Books:** Walter Savitch and Kenrick Mock *Absolute C++ (6th Edition)* Pearson, 2015 ISBN: 978-0133970784

Nicolai M. Josuttis *The C++ Standard Library: A Tutorial and Reference (2nd Edition)* Addison-Wesley Professional, 2012 ISBN: 978-0321623218

Bjarne Stroustrup *The C++ Programming Language (4th Edition)* Addison-Wesley Professional, 2013 ISBN: 978-0321958327

14. Course Expectations:

- a. **Attendance:** The classroom is the heart of the educational experience because it provides, uniquely, a formal setting for the important exchanges among faculty and students. Regular and punctual attendance at all classes, essential for maximum academic achievement, is a major responsibility of the students. Failure to attend and contribute to the classroom environment significantly and demonstrably reduces the quality of the educational experience for everyone in the classroom. As a result, absences almost always impact the quality of performance. Students are expected to arrive on-time and with all required material for the taking of notes (pencil and notebook preferred).

Students are expected to attend the Zoom meetings at the scheduled time.

In the event of prolonged illness, accident, or similar emergency, it is the responsibility of the student to notify the instructor. Students must remember that it is always their responsibility to make up the work they may have missed during an absence from class. Students are directed to confer with their instructor when their absences jeopardize satisfactory progress.

- b. **Academic Honesty Policy:** Integrity and honesty are extremely important qualities of any academic or professional person. That being said, science is a collaborative effort. I encourage students to discuss course material outside of class and form study groups for exams and quizzes. **However, you must do the homework by yourself. If you need help, I will be happy to help you!** You may also seek assistance from the course Teaching Assistants.

The use of material from previous classes, solution manuals, material from the Internet or other sources (e.g., parents, siblings, friends, etc.) that directly bears on the answer is strictly prohibited.

Examples of academic misconduct include copying from another student's exam, using unauthorized information from notes, calculators, phones, or computers during exams, copying all or part of another student's assignments such as code or diagrams and providing one's work to other students for their use. These examples are not exhaustive.

The Computer Science Department considers any instance of academic misconduct to be a serious offense. The penalty for an instance of academic misconduct may include, among other things, a zero for the assignment(s) in question, a failing grade for the course in question, or expulsion from the University.

Allowing another student to examine and/or copy your work constitutes an instance of academic misconduct both by you and by the other student. Thus academic misconduct includes not only taking others' work and submitting it as your own, but also allowing others to use your work or submit your work as their own. Students allowing others to copy their work will be charged with academic misconduct. They are subject to the same penalties listed. Each student is responsible for safeguarding his/her own work. The use of a personal flash drive is recommended if you will be using a shared computer.

Examining and/or copying the work of another without their permission constitutes, in itself, a separate instance of academic misconduct. Thus it is not permissible to examine the work of others which is found either in printed form, as in a lab, or in electronic form, as on a public hard drive. Students who take the work of others in this manner will be charged with academic misconduct. They are subject to the same penalties listed above.

- c. **Lecture Notes and Handouts:** Only a part of the lecture notes will be available in tradi-

tional digital-file formats (*.doc, *.ppt, and *.pdf) on the BlackBoard site related to our course: <https://uml.edu/myuml>. Some handouts will be distributed in class and some notes will be written on the classroom board. Please, take notes every class and share the notes if someone needs them.

- d. **University Email System:** The University of Massachusetts - Lowell has established a University electronic mail ("email") system as a means of the University sending official information to enrolled students, and for students to send communication to their instructors and University personnel. All registered students will be assigned a University email account/address to be used as the only email address for all email communication: 1) sent to the students from their instructors and from all University personnel; and, 2) sent by the students to their instructors and to all University personnel.

In addition:

- Students should check their University email account regularly to ensure they are staying current with all official communications. Official communication includes, but is not limited to, policy announcements, registration and billing information, schedule changes, emergency notifications and other critical and time sensitive information.
 - Students should also check their University email account to be sure that they are current with all email communication from their faculty.
 - The student email account/address should be the only e-mail address students use to send email to faculty and University personnel so that student email is recognized and opened.
 - Accounts are for individual use only, and are not transferable or to be used by any other individual.
- e. **Plagiarism Policy:** Plagiarism is a serious violation of a student's academic integrity and the trust between a student and his or her teachers. Plagiarism is the act of a person presenting another person's work as if it were his or her own original work. Such acts of plagiarism include, but are not limited to:
1. A student submitting as his or her own work an entire essay or other assignment written by another person.
 2. A student taking word for word a section or sections of another person's work without proper acknowledgment of the source and that the material is quoted.
 3. A student using statistics or other such facts or insights as if these were the result of the student's efforts and thus lacking proper acknowledgment of the original source.
 4. The paraphrasing of another person's unique work with no acknowledgment of the original source.
 5. Copying another student's work on a quiz or test.

When a student is found to have plagiarized an academic assignment, it will be up to each instructor to determine the penalty. Depending on the severity of the incident, this could range from a warning to a loss of credit for the assignment. In all cases of plagiarism, the student's program coordinator will automatically be notified and the incident will be documented. If any further incidents of plagiarism are reported to the student's program coordinator, additional sanctions will be imposed. These may include notification of the Dean; loss of credit for the course; suspension or dismissal from a department program; academic probation; and/or expulsion from the University.

Additional information regarding University Policies is available:

- i. Academic Policies:

<https://www.uml.edu/Catalog/Undergraduate/Policies/Academic-Policies/Academic-Policies.aspx>

ii. Kennedy College of Sciences Policies:

<https://www.uml.edu/Catalog/Undergraduate/Sciences/policy/default.aspx>

iii. Computer Science Department Policies:

<https://www.uml.edu/Catalog/Undergraduate/Sciences/policy/Continuance-appeal-dismissal.aspx>

iv. Advising Resources:

<https://www.uml.edu/Academics/Provost-office/Faculty-success/Professional-Engagement/advising-resources.aspx>

15. **Evaluation Methodology:**

Item	Percentage
Programming Assignments (HW0-HW9)	30
Quizzes	10
zyBook Assignments	20
Exams	20
Final Exam	20

The grade calculated above will be combined with your lab grade (75% recitation, 25% lab). Any requests for correcting grades must be made within 2 weeks of the Quiz/Exam/Homework due date. There will not be any opportunity to “re-do” an assignment, quiz or exam. No extra credit will be assigned or accepted. All Labs must be completed and submitted to the Lab TA by midnight of the last day of classes (December 12, 2024). The following table describes the translation from your numeric grade to letter grade (You will be given the same letter grade for both the recitation section and your lab section).

Numeric range	Letter grade
90.0 – 100	A
87.0 – 89.9	B+
83.0 – 86.9	B
80.0 – 82.9	B-
77.0 – 79.9	C+
73.0 – 76.9	C
70.0 – 72.9	C-
67.0 – 69.9	D+
60.0 – 66.9	D
0 – 59.9	F

16. **Due Dates:** Programming assignments (HW0-HW9) are due at 11:59 pm of the Wednesday class meeting of the indicated week. Most of these assignments build upon a previous assignment. All assignments must be submitted as plain text files on Blackboard. (Depending upon the particular assignment, you may need to submit .cpp (c++ source), .h (c++ header), and .txt (captured output files). Any images or “project” files from an IDE will receive a grade of zero. Please heed any suggestions in following assignments that make use of previous work.

Late assignments will *not* be accepted without prior approval. Students must consult with the instructor at least three days prior to the scheduled due date to make alternative arrangements. The instructor is under no obligation to grant any such request and penalties such as a reduced score may be applied at the instructor's discretion.

The zyBook Assignments will consist of both "Participation" and "Challenge" activities in the appropriate zyBook chapter(s). Please do all of the "Participation & Challenge" Activities as you read each chapter (*Before class*). Due dates and zyBook assignments can be found on the zyBook web page. You are responsible for completing these assignments by the due date. Due dates for assignments will *not* be extended. ZyBook assignments are typically due early Monday morning. The due dates are listed in the zyBook.

In addition to the zyBook material, "packets" of questions will be available on BlackBoard. Please try to do each packet *Before* looking at the answers (also available on BlackBoard). The packet questions will be the basis for the biweekly quiz. The Packet Quiz will typically occur on Friday. The quiz will be announced no later than Wednesday of that week (both in class and on BlackBoard). Before the quiz, we will discuss any packet questions you have had trouble with (several of the questions require *very* careful reading).

17. **Exams:** Exams will be due at 11:59 PM on the last meeting day of the indicated week. If you have an accommodation and wish to use an 'extra time' accommodation, you must make arrangements at least 3 days before the exam.

The date and time of the final exam will be determined by the Registrar's office. Please do *not* make any travel arrangements that will make you unavailable between December 14, 2024 and December 21, 2024. Please also note that the final exam may be scheduled for Saturday December 14, 2024 or December 21, 2024. You must submit the final exam by the 'end time' of the assigned time slot. These deadlines also apply to those that may have an accommodation and have an 'extra time' accommodation. If this is the case, you must email the instructor at least 3 days before the exam. You will be emailed a copy of the exam *before* the normal start time. You must submit your exam by the deadline that applies to everyone.

In addition to exams, there will be quizzes (about every other week). The quizzes will consist of selected questions from the most recent "packet".

18. **Schedule** (subject to adjustment):

Week	Topic	Savitch	Online	Events
1	Introduction & C++ basics	1, 2	1, 2	
2	Input & Output Streams	3, 5, 12	3, 4	HW0 due
3	Reference parameters Introduction to Classes	4, 6	6, 7	HW1 due
4	Dynamic memory management in C++	10	8	
5	Constructors & Operator Overloading	8	7.6, 7.11	HW2 due
6	Strings & Vectors	9	5	HW3 due
7	Templates	16	13	Exam 1
8	Name Spaces, STL Containers	11, 19	7.6, 14	HW4 due
9	Inheritance	14	10	HW5 due
10	Polymorphism	15	10.4	HW6 due
11	Exception Handling	18	12	HW7 due
12	Exception Handling	18	12	Exam 2
13	STL Algorithms			
14	STL Algorithms			HW8 due
15	Generic Algorithms Review			HW9 due
16	Review			
Final Exams				

Week	Days		Dates	
1	W	F	9/4	9/6
2	M	W	F	9/9 9/11 9/13
3	M	W	F	9/16 9/18 9/20
4	M	W	F	9/23 9/25 9/27
5	M	W	F	9/30 10/2 10/4
6	M	W	F	10/7 10/9 10/11
7	T	W	F	10/15 10/16 10/18
8	M	W	F	10/21 10/23 10/25
9	M	W	F	10/28 10/30 11/1
10	M	W	F	11/4 11/6 11/8
11		W	F	11/13 11/15
12	M	W	F	11/18 11/20 11/22
13	M			11/25
14	M	W	F	12/2 12/4 12/6
15	M	W		12/9 12/11
	Final Exams		12/14 – 12/21	