

INFO 2680-061 - **C++ Programming**
10 week accelerated schedule (online) - 3 credit hours
Course Syllabus - [Spring 2019](#)

General Course Information

Instructor: Dr. Yelena Rykalova

Course Chat Schedule: Tuesday **nights from 9 to 10 pm EST**

Email: yelena_rykalova@uml.edu (please use Blackboard Mail for course related mail, only use this UML address in the rare occasion when Blackboard is down)

Course Description

This course introduces you to the C++ programming language using an object oriented programming approach.

The course covers the C++ language, including topics of class design and construction, function and operator overloading, inheritance, virtual functions, exception handling, templates, the C++ Standard IO streams classes, and the C++ Standard Template Library (STL). The use of inheritance in creating extendable libraries will be presented. Object oriented concepts will be presented in the context of the C++ language and its support for object oriented programming. Programming examples will be presented during the course. The examples are not tied to any particular compiler, so you can choose a C++ development environment.

Note that C++ is an extremely powerful and complex language, *much* more challenging to learn and understand well than C. Much of the challenge comes from trying to understand many new concepts (e.g. object oriented programming, classes, composition, inheritance, dynamic binding, operator overloading, exception handling, templates, Standard Template Library, etc.) rather than language syntax. You should allow sufficient time for study and experimenting with the many new features of C++.

Prerequisites

Prior programming experience in C is required, including use of pointers and dynamic memory management (malloc/free), and simple dynamic linked list data structures.

You should have a working knowledge of the C programming language and be familiar with a C development environment that includes a debugger. You should have an understanding of and experience using the following C programming features:

- variables, constants, and expressions
- data types (integers, floating point, characters, and strings (character arrays))
- decision making (if/then/else statements)
- loops (for and while statements)

- arrays (collections of values)
- functions
- reading values from the keyboard and files, and writing values to the console and files.
- pointers
- dynamic memory management (malloc/free)
- dynamic linked list structures

The course concentrates on introducing new concepts and C++ features such as:

- classes, objects, and object oriented programming
- methods and parameters (including overloading, overriding, and default parameters)
- scope, lifetime, namespaces
- inheritance and dynamic binding
- exception handling
- streams and file I/O
- templates
- Standard Template Library (STL)

Object oriented languages like C++, Java, and C# provide new features and ways of organizing and designing your code. One purpose of this course is to introduce you to these new concepts and features using C++, building on your current understanding of programming concepts and prior programming experience.

Course Requirements and workload

This course is presented in an accelerated 10 weeks format. The amount of time and effort required will vary depending on your background, experience, skills, and general aptitude for programming.

This course was developed by [Lance Smith](#). From the course evaluations he received, students with less programming experience typically spend at least 3-4 hours reading per week, and possibly an hour or two running some of the exercises/examples in the book in order to better understand the concepts being presented. They also spend between 3 to 10+ hours on the homework assignment each week, probably averaging about 6-8 hours. In general, they've indicated that this programming course plus one other (non-programming) course is the maximum they could handle while working full time.

The homework ranges in difficulty with those in the second half of the course being more challenging as new object oriented concepts are introduced. The assignments are available early in the course to allow you to work ahead to accommodate vacation plans, travel requirements, etc.

This course focuses on writing C++ code. The best way to learn a programming language is to write code. Although reading the book will provide an 'intellectual' understanding of concepts in many cases, when you start entering code on a blank editor screen thing can become a bit more challenging :-). Converting your reading and understanding of concepts to actually writing C++ code is where things come together and most of the learning occurs, many times by making mistakes and learning from them!

Recommended Textbooks

C++ : How to Program, Tenth Edition

Harvey Deitel and Paul Deitel
Prentice Hall 3/2016
ISBN-10: 0134448235
ISBN-13: 978-0134448237

The best-selling C++ How to Program is accessible to readers with little or no programming experience, yet comprehensive enough for the professional programmer. The Deitels' signature live-code approach presents the concepts in the context of full working programs followed by sample executions. The early objects approach gets readers thinking about objects immediately—allowing them to more thoroughly master the concepts. Emphasis is placed on achieving program clarity and building well-engineered software. Interesting, entertaining, and challenging exercises encourage students to make a difference and use computers and the Internet to work on problems. To keep readers up-to-date with leading-edge computing technologies, the Tenth Edition conforms to the C++11 standard and the new C++14 standard.

This book may be ordered from the UMass bookstore, www.barnesandnoble.com, www.superbookdeals.com, and of course www.amazon.com. Prior students have also recommended checking www.abebooks.com for low prices on used copies.

This book is available online via CourseSmart at <http://www.coursesmart.com/>, and also at the UMass library Safari eBooks online (see below).

The 9th, 8th, or 7th edition of the textbook may be used for this class. It may be available for a lower cost on the used market.

C++ : How to Program, Nineth Edition

Harvey Deitel and Paul Deitel
Prentice Hall 2/2013
ISBN-10: 0133378713
ISBN-13: 978-0133378719

C++ : How to Program, Eighth Edition

Harvey Deitel and Paul Deitel
Prentice Hall 3/2011

ISBN-10: 0132662361
ISBN-13: 978-0132662361

C++ : How to Program, Seventh Edition

Harvey Deitel and Paul Deitel

Prentice Hall 2009

ISBN-10: 0136117260

ISBN-13: 9780136117261

Required Software

The course requires current C++ development software that includes the C++ Standard Library (STL). Use of an Integrated C++ Development Environment (IDE) with online help and an interactive visual debugger [is highly recommended](#), e.g. [Microsoft Visual Studio \(VS\)](#). [I will be grading assignments using VS](#). The book's companion CD-ROM includes a list of sources for C++ compilers.

You **cannot** use the `ideone.com` site for the programming assignments for this course.

This course qualifies for a [free copy of Microsoft Visual Studio 2017 Professional Edition](#), Windows 10 or 8.1 Professional Edition, and other development tools via the Microsoft DreamSpark program. Download instructions are available at: <https://continuinged.uml.edu/msdnaa.cfm>

Microsoft makes Express editions of its software available for free download at: <https://www.visualstudio.com/vs/visual-studio-express/>

New to online Blackboard courses?

If you've not taken an online course with Blackboard before, UMass has a tutorial at: https://uml.umassonline.net/bbcswebdav/institution/Resources_and_Training/UML_Online_Learning_Tutorials/Blackboard_Learn_Video_Tutorials.htm

UMass Library Safari eBooks online

UMass has a large number of eBooks online, including the Safari Technical Reference eBooks. The steps to access these books are:

- Go to <http://libweb.uml.edu/>
- Click on **eBooks** on the left side menu
- On the eBook Collections page, scroll down to **Safari Computer Books** near the bottom of the page and click on the link.
- On the **UML Libraries** page, enter your UMass student email address and password and login

- Click on the book category of interest or enter a search term (e.g. **C++ How to Program**).
- Click on the book link to view the eBook.

Interaction Guidelines:

Chat night

Interactive chats will be held on **Tuesday** nights from **9 to 10 pm** EST. The focus of the chats is to provide additional help and explanation of C++ concepts, homework problems, etc. Participation is optional. Chat sessions are logged online and can be reviewed at any time. There will not be a chat during finals week.

In case my schedule doesn't allow me to participate in chat, I'll try to notify the class ahead of time.

Discussion forum

You are encouraged to post topics and questions to each week's discussion forum as needed, and participate in discussions on C++ features, homework issues, etc.

When posting questions seeking help on homework issues, please do **not** post your entire C++ source file or large blocks of source code to the discussion forum. A few specific lines of code is ok if it helps explain the problem.

When providing help to others, please try to help them work through the problem themselves (i.e. teach principles and concepts) rather than just giving the complete answer.

Mail (Messages - Mail on the Course Menu)

Electronic mail should be reserved for private or personal issues. You may email me at any time, and I will try to respond within 24 hours. Please note that Blackboard works with its own internal mail system, which means all messages are sent, stored and read within your course, and not through your own external mail. To access mail, log on and go to your course, and click on the mail tool on the course menu on the left side your screen. It is recommended that you check Mail at least once a day, so that you don't miss important course information. Blackboard provides a convenient indicator if you have unread mail waiting for you. Once you log into Blackboard, before clicking on a course link, if you look at the upper-left corner of your UML Online Learning page, you will see a "My Messages" area. If there is new mail waiting for you, they will show up there. You can simply click on the link and you will be brought directly into your course mail system.

Assignment Guidelines:

Grading

Your total course grade is calculated as follows:

5 assignments (12 % each)	60 %
1 project	20 %
Final exam	20 %
	100%

Your course letter grade is calculated as follows:

Numeric Grade:	Letter Grade:
95 - 100	A
90 - 94	A-
87 - 89	B+
83 - 86	B
80 - 82	B-
77 - 79	C+
73 - 76	C
70 - 72	C-
67 - 69	D+
60-66	D
Below 60	F

Please review the University's Academic Policies page at:

<https://continuinged.uml.edu/policies/grading.cfm>

Reading

You should complete the reading assignments before reading the course notes for that week. The lecture notes assume you have read and understood the material (or asked questions on the discussion board, chat, or Mail). You may want to review additional C++ materials for further information on specific topics.

Assignments

Detailed information for the course assignments is located in the "[Assignments](#)" area of the course menu, including how to submit them, coding standard, etc. Assignments are due by **Sunday midnight** of the week due in the above Course Schedule. **Five (5) assignments** and **one (1) Project** are required to complete the course. An assignment may be turned in up to **one** week late, for a 10% loss of credit.

Assignments are graded as follows (out of 100 points):

- 100 points - meets all requirements and runs without error
- 80 points - meets the minimum requirements with minor errors.
- 60 points - does not meet the minimum requirements but runs without error.
- 40 points - meets the minimum requirements with major errors.
- 20 points - does not meet the minimum requirements and/or does not compile

Final Exam

The final exam will be "take home" open book. The exams will be available online,

You will have about one week to complete it. The due date for the exam is in the Course Schedule. *You must complete and return the exam to receive a grade for the course.*

Course Add / Drop Dates

Drop, add, and withdrawal dates for different types of online courses (accelerated, etc.) vary greatly. You should review the UMass Academic Calendar page for the [important deadlines](#) for this semester at: <https://continuinged.uml.edu/general/calendar.cfm>

Make Up Policy

An assignment may be turned in/resubmitted up to **one** week late, for a 10% loss of credit. No resubmission allowed for the Project.

Academic Integrity Policy

UMass Lowell Online students are expected to be honest and to respect ethical standards in meeting academic assignments and requirements. A student who cheats on an examination or assignment is subject to administrative dismissal. Please visit the [Academic Integrity](#) Web site for specific details regarding this policy.

Student Disability Services

UMass Lowell Online students requiring academic accommodations should contact [Student Disability Services](#) for assistance.

Course Calendar/Schedule - Important Dates!

The following table lists the schedule of topics, recommended reading, homework assignments, and final exam for the course.

Week	Start Date	Topics	Recommend Reading <i>Deitel 10/e</i>	Recommend Reading <i>Deitel 9/e</i>	Recommend Reading <i>Deitel 8/e & 7/e</i>	Assignments Due Dates
1	1/22	Introduction, History, and Overview of C++ Function prototypes, C++ Standard Library headers References and reference parameters C++ pre-processor	Ch. 1 Ch. 2 Ch. 6.4 - 6.6, 6.13 Appendix E	Ch. 1 Ch. 2 Ch. 6.4 - 6.6, 6.15 Appendix E	Ch. 1 Ch. 2 Ch. 6.4 - 6.6, 6.14 Appendix E	Install C++ Development Environment
2	1/28	Introduction to Classes Pointers and strings	Ch. 3 Ch. 8.1 - 8.9 Ch. 9.1 - 9.10	Ch. 3 Ch. 8.1 - 8.9 Ch. 9.1 - 9.9	Ch. 3 Ch. 8.1 - 8.9 Ch. 9	Try some sample C++ code and debugging
3	2/4	More on Classes Static and const members Containers Pointer based string processing Using a debugger	Ch. 8.10 Ch 9.11 - 9.16 Appendix H	Ch. 8.10 Ch 9.10 - 9.15 Appendix H	Ch. 8.10 Ch. 10 Appendix H	Assignment #1 due Sunday 2/10
4	2/11	Function overloading, Scope and visibility Call Stack and Activation Records Inline functions, default arguments Arrays, STL vector class Namespaces	Ch. 6.10 - 6.12, 6.14 - 6.16 Ch. 7.1 - 7.5, 7.7, 7.10 Ch. 23.4 - 23.5	Ch. 6.10 - 6.14, 6.16 - 6.18 Ch. 7.1 - 7.5, 7.7, 7.10 Ch. 23.4 - 23.5	Ch. 6.9 - 6.13, 6.15 - 6.17 Ch. 7.1 - 7.5, 7.7, 7.11 Ch. 24.4 - 24.5	Assignment #2 due Sunday 2/17
5	2/18	Operator Overloading	Ch. 10	Ch. 10	Ch. 11	Assignment #3 due Sunday 2/24
6	2/25	Inheritance, virtual functions, static/dynamic binding Multiple inheritance (optional)	Ch. 11 Ch. 12 Ch. 23.7 - 23.8	Ch. 11 Ch. 12 Ch. 23.7 - 23.8	Ch. 12 Ch. 13 Ch. 24.7 - 24.8	Assignment #4 due Sunday 3/3
7	3/4	Templates - functions and classes Stream Input/Output	Ch. 18 Ch. 13	Ch. 18 Ch. 13	Ch. 14 Ch. 15	Assignment #5 due Sunday 3/10

8	3/18	File processing Class string and string stream processing Exception Handling (optional)	Ch. 14.1 - 14.5 Ch. 21 Ch. 17	Ch. 14.1 - 14.5 Ch. 21 Ch. 17	Ch. 17.1 - 17.5 Ch. 18 Ch. 16	Work on project
9	3/25	Standard Template Library (STL) Review for Final	Ch. 15 Ch. 16	Ch. 15 Ch. 16	Ch. 22	Project due Sunday 3/31
10	4/1	FINAL EXAM - Due Sunday 4/7 Available Mon 4/1				Final Exam due Sunday 4/7

Course Outline

Getting Started

- Welcome to *C++ Programming*
- What to expect for this course
- Optional books

Week 1 - Introduction, History, and Overview of C++

- History of C++
- Software in Crisis
- Points to Remember
- C++ compilers
- Introduction to C
- C++/C Small Stuff
- Coding Standard
- Exercise: Hello World in C++
- Background Topic: Pointers, Dynamic Memory, Linked Lists
- Paper: An Overview of C++
- Paper: As Close as Possible to C - But No Closer
- Paper: What is Object Oriented Programming?

Week 2 - Introduction to Classes, Pointers and Strings

- Why do we need something new?
- Classes with C++
- The development of the C++ class
- Starting with constructors
- Notes about destructors
- Highlights
- C++ style hints

Week 3 - More on Classes, Static and Const

- More on classes
- Special constructors
- Static and Const members
- Container class example
- List, Containers, and Friends

- Default constructors
- Highlights

Week 4 - Function overloading, Scope, Visibility, and Namespaces

- Scopes and visibility
- Namespaces
- Static
- More examples of the use of static
- Summary of this week

Week 5 - Operator Overloading

- Operator Overloading
- Rules
- More on Operator Overloading
- Another Example
- Creating a subtype
- Casting operators

Week 6 - Inheritance, Virtual Functions, Static and Dynamic Binding

- Inheritance
- Definition
- Terms
- Inheritance Examples
- Virtual Functions
- Optional: Multiple Inheritance, Multiple Inheritance Considerations

Week 7 - Templates - Functions and Classes, Standard Library IOSTream Classes

- Function Templates
- Class Templates
- More examples
- Standard IOSTream Library
- Notes about IOSTream
- Examples of IOSTreams
- Examples of FStream usage

Week 8 - Exception Handling, Standard Library string class

- File Processing
- Reading and writing sequential files
- std::string class
- String stream processing
- Optional: Exception Handling

Week 9 - Standard Template Library (STL)

- Standard Template Library
- Examples of STL usage

Week 10 - Final Exam

- Final exam
- Course survey