

UMBC UGC New Course Request: CMSC 340 – Advanced C++

Date Submitted: 3/2/2020

Proposed Effective Date: 8/23/2020

	Name	Email	Phone	Dept
Dept Chair or UPD	Jeremy Dixon	jdixon@umbc.edu	5-8866	CSEE
Other Contact	Jeannette Kartchner	jkartch@umbc.edu	5-1826	CSEE
Other Contact	Mohamed Younis	younis@umbc.edu	5-3968	CSEE

COURSE INFORMATION:

Course Number(s)	CMSC340
Formal Title	Advanced C++
Transcript Title (≤30c)	Advanced C++
Recommended Course Preparation	
Prerequisite	
Unless otherwise indicated, a prerequisite is assumed to be passed with a "D" or better.	Corequisite with CMSC 341
# of Credits	3
Must adhere to the <u>UMBC Credit Hour Policy</u>	
Repeatable for additional credit?	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No
Max. Total Credits	3 <small>This should be equal to the number of credits for courses that cannot be repeated for credit. For courses that may be repeated for credit, enter the maximum total number of credits a student can receive from this course. E.g., enter 6 credits for a 3 credit course that may be taken a second time for credit, but not for a third time. Please note that this does NOT refer to how many times a class may be retaken for a higher grade.</small>
Grading Method(s)	<input checked="" type="checkbox"/> Reg (A-F) <input checked="" type="checkbox"/> Audit <input checked="" type="checkbox"/> Pass-Fail

PROPOSED CATALOG DESCRIPTION (Approximately 75 words in length. Please use full sentences.):

This course will cover advanced programming topics in C++ to include topics including pointers, memory allocation, object-oriented class design, inheritance, polymorphism, operator overloading, templates, and exceptions. In-class programming practice, projects, and design assignments will be used to improve programming skills. This class may be especially helpful for transfer students who have taken Computer Science I & II in Java and are ready to take CMSC 341. This class must be taken concurrently with CMSC 341 Data Structures.

RATIONALE FOR NEW COURSE:

a) Why is there a need for this course at this time?

After a thorough analysis related to our retention rates for CMSC students, we have identified that transfer students are more than twice as likely to be unsuccessful (DFW) in our traditional CMSC 341 course than our UMBC first-time freshman population. As such, we are proposing a new course that students can optionally take with CMSC 341. The goal is to provide additional support to students who have a programming background in Java instead of C++. Most of the transfer students (especially those from neighboring community colleges) have a significant background in Java. CMSC 340 will leverage the transfer students' knowledge in Java to improve their success rate in CMSC 341. At the end of the semester, after taking both CMSC 340 and CMSC 341, students will have achieved proficiency in the C++ language and all of the learning outcomes of CMSC 341. We believe this will improve the retention rate of transfer students in the computer science major.

- b) How often is the course likely to be taught?
We hope to offer it every semester at both UMBC and USG.
- c) How does this course fit into your department's curriculum?
This course will be an optional accompaniment with CMSC 341. Every CMSC student will optionally take it.
- d) What primary student population will the course serve?
The course is designed for sophomore and juniors who are CMSC majors and minors.
- e) Why is the course offered at the level (ie. 100, 200, 300, or 400 level) chosen?
Since CMSC 341 (which has CMSC 202 and CMSC 203 as prerequisites) is a corequisite, this course should be offered at the 300 level as well. has several prerequisites and it the prerequisite for almost all 400 level CMSC courses.
- f) Explain the appropriateness of the recommended course preparation(s) and prerequisite(s).
This course has a corequisite of CMSC 341 (CMSC 341 has prerequisites of CMSC 202 and CMSC 203). Therefore, students should be familiar with programming in either Java or C++ but most likely Java.
- g) Explain the reasoning behind the P/F or regular grading method.
Students can take this course for a letter grade, P/F, or audit although almost all students take it as A-F.
- h) Provide a justification for the repeatability of the course.
This course cannot be repeated.

ATTACH COURSE SYLLABUS (mandatory):

CMSC 340: Advanced C++

Prerequisites:

Corequisite of CMSC 341.

Instructor:

Name: TBD

Office: TBD

Office Hours: TBD

Phone: TBD

Email: TBD

Course Description:

This course will cover advanced programming topics in C++ to include topics including pointers, memory allocation, object-oriented class design, inheritance, polymorphism, operator overloading, templates, and exceptions. In-class programming practice, projects, and design assignments will be used to improve programming skills. This class may be especially helpful for transfer students who have taken Computer Science I & II in Java and are ready to take CMSC 341. This class must be taken concurrently with CMSC 341 Data Structures.

Credits:

Three credits: not repeatable

Books:

Downey, A. (2016). Think Data Structures Algorithms and Information Retrieval in Java. Green Tea Press.
(<http://greenteapress.com/thinkdast/thinkdast.pdf>). (free – required)

Learning Outcomes:

At the end of the course, the student will:

1. Improve problem solving abilities, program design skills and coding skills.
2. Understand the fundamental programming concepts of an object-oriented programming paradigm, including data abstraction, encapsulation, classes, inheritance, composition, polymorphism, exception handling, and recursion.
3. Apply C++ programming concepts such as templates, pointers, virtual functions, dynamic allocation of memory, dynamic binding, memory management and operator overloading.
4. Analyze a problem and construct a C++ program that solves it.
5. Choose and apply the required commands to develop, test and debug C++ programs in a command-line environment.

Course Format and Procedures:

CMSC 340 is an optional course to act as an accompaniment to the required CMSC 341 – Data Structures designed for students who have a significant background in Java instead of C++. Students are required to take one of the following: CMSC 341 or CMSC 341H. CMSC 340 is not required for the CMSC major.

Students with a background programming in Java may find CMSC 341 too challenging as they are expected to learn a new programming language (C++) and new data structures concurrently. This course is designed to provide additional language specific instruction for students new to C++. For example, the instructor may introduce a concept in C++ during lecture and then reinforce it with an in-class activity. Some data structures will be presented first in Java and then again in C++ in order for students to be able to compare and contrast the structures in the two programming languages.

The course itself will be a combination of introducing new structures and concepts and then doing related coding examples. The course will be project heavy requiring students to program throughout the semester.

Readings:

Evans, B. J., & Flanagan, D. (2019). Java in a nutshell. Sebastopol, CA: O’Reilly Media, Inc.

Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). Data Structures & Algorithms in Java. New Jersey, John Wiley & Sons, Inc.

Savitch, W. (2015). Absolute C++. 6th Edition. Pearson. ISBN: 978-0133970784

Course Topics:

Students in data structures will participate by:

- Introduction: Introduction to data structures and setting up your environment.
- Object Oriented Programming (Java and C++)
 - Classes and objects
 - Encapsulation and data hiding
 - Inheritance/Aggregation
 - Polymorphism
 - Pointers
 - Constructors, Copy Constructors, and Destructors
 - Operator overloading
 - Generics
 - Exceptions
- Supporting Data Structures Taught in CMSC 341

Grading:

Assignment	Total
Projects	40%
Homework/Labs	15%
Exams	45%
Total:	100%

Grading is on a standard 10-point scale, so you will get an A for 90.0 or more total points, a B for 80.0 or more but less than 90.0 points, and so on.

The homework assignments will be a blend of practical exercises and questions that cement conceptual knowledge. The projects will be assigned both in Java and C++ and in some cases will be the same project assigned in both languages.

Academic Integrity

By enrolling in this course, each student assumes the responsibilities of an active participant in UMBC's scholarly community in which everyone's academic work and behavior are held to the highest standards of honesty. Cheating, fabrication, plagiarism, and helping others to commit these acts are all forms of academic dishonesty, and they are wrong. Academic misconduct could result in disciplinary action that may include, but is not limited to, suspension or dismissal. To read the full Student Academic Conduct Policy, consult the Academic Integrity Resources for Students page (<https://aetp.umbc.edu/ai/resources-for-students/>) or the Faculty Handbook (<http://provost.umbc.edu/faculty-handbook/>), specifically Sections 14.2-14.3.

If you need help with a project, see your instructor, your TA, or tutors provided by the Learning Resource Center. We also encourage you to consult textbooks and code examples provided on Blackboard. Consult Blackboard for additional Academic Integrity policies for projects.

Any act of dishonesty will be reported to the University's Academic Conduct Committee for further action, which may include, but is not limited to, academic suspension or dismissal from the University.

We will be using special software to check for cheating. The software is quite sophisticated and has surprised many students in the past. There is no difficulty in comparing every pair of assignments – even assignments submitted to other sections of this course, or from previous semesters.

This is a *non-exhaustive* list of restrictions for completing your assignments in this course.

- **If you have questions about what is acceptable, please contact a professor or TA.**

You may not look at, access, download, or obtain anyone else's work.

- You should think carefully about the assignment, and the assignment you turn in should be entirely a product of your own understanding of the material.
- You may not use any online resources to request additional help. Please contact a professor or TA for additional help.
- You may not post any part of a course document online. Posting any slides, projects, or labs will be considered a violation of this course policy and will result in an "F" for the course.
- You may not look at someone else's code "for reference," even if you put it aside before programming, and even if that person is not a CMSC student.
- You may not Google or search for the solution to an assignment, even if it's "only for reference."
- You may not copy code other than that provided in the course materials (slides, book, labs, etc.).

- You may not let someone else explain a solution to you in such detail that they are effectively dictating the code to you line by line. It does not matter if this person has never taken this course, or if they are not looking at their own code while doing so!

Student Disability Services:

UMBC is committed to eliminating discriminatory obstacles that may disadvantage students based on disability. Services for students with disabilities are provided for all students qualified under the Americans with Disabilities Act (ADA) of 1990, the ADAAA of 2009, and Section 504 of the Rehabilitation Act who request and are eligible for accommodations. The Office of Student Disability Services (SDS) is the UMBC department designated to coordinate accommodations that would allow students to have equal access and inclusion in all courses, programs, and activities at the University.

If you have a documented disability and need to request academic accommodations for access to your courses, please refer to the SDS website at sds.umbc.edu for registration information and to begin the process, or alternatively you may visit the SDS office in the Math/Psychology Building, Room 212. For questions or concerns, you may contact us through email at disAbility@umbc.edu or phone (410) 455-2459.

If you require accommodations for this class, make an appointment to meet with your instructor to discuss your SDS-approved accommodations.

Disclosures of Sexual Misconduct and Child Abuse or Neglect

As an instructor, I am considered a Responsible Employee, per UMBC's Policy on Prohibited Sexual Misconduct, Interpersonal Violence, and Other Related Misconduct (located at <http://humanrelations.umbc.edu/sexual-misconduct/umbc-resource-page-for-sexual-misconduct-and-other-related-misconduct/>). While my goal is for you to be able to share information related to your life experiences through discussion and written work, I want to be transparent that as a Responsible Employee I am required to report disclosures of sexual assault, domestic violence, relationship violence, stalking, and/or gender-based harassment to the University's Title IX Coordinator. As an instructor, I also have a mandatory obligation to report disclosures of or suspected instances of child abuse or neglect (www.usmh.usmd.edu/regents/bylaws/SectionVI/VI150.pdf).

The purpose of these reporting requirements is for the University to inform you of options, supports and resources; you will not be forced to file a report with the police. Further, you can receive support and resources, even if you choose to not want any action taken. Please note that in certain situations, based on the nature of the disclosure, the University may need to act.

If you need to speak with someone in confidence about an incident, UMBC has the following Confidential Resources available to support you:

The Counseling Center: 410-455-2472

University Health Services: 410-455-2542

(After-hours counseling and care available by calling campus police at 410-455-5555)

Other on-campus supports and resources:

The Women's Center, 410-455-2714

Title IX Coordinator, 410-455-1606

Additional on and off campus supports and resources can be found at:
<http://humanrelations.umbc.edu/sexual-misconduct/gender-equitytitle-ix/>

Tentative Schedule:

Week	Topic	Assignments
1	Introduction	
1	C++ Primer and Compilation	Lab 1
2	Functions and Overloaded Functions	
2	Pointers	Project 1
3	Dynamic Memory	Lab 2
3	Classes and Objects in C++	
4	Classes and Objects in C++	
4	Inheritance	
5	Inheritance	Project 2
5	Polymorphism	
6	Polymorphism	Lab 3
6	Makefiles	
7	Exam 1 Review	
7	Midterm Exam 1	
8	Linked Lists	Project 3
8	Asymptotic Analysis	
9	Running Times	Lab 4
9	STL Containers	
10	STL Containers	
10	Operator Overloading	Project 4
11	Copy and Assignment	
11	Range Average Query	Lab 5
12	Exam 2 Review	
12	Exam 2	
13	Function Templates	Project 5
13	Class Templates	Lab 6
14	<u>Case Study: The Pokegraph</u>	
14	C++ 11 Features	
15	C++ 11 Features	Lab 7
15	Final Exam Review	

This schedule is subject to change without notification from the professor.