Introductory Robot Programming
ENPM809Y

Instructor Info

Zeid Kootbally

Office Hrs: Mon & Fri 4-5 pm

Office: N/A

Website: N/A

zeidk@umd.edu

Course Info

Prereq: None

Thurs

7-9:40 pm

JMP2121 & Zoom

Grader Info

Sahana Anbazhagan

Office Hrs: TBD

Overview

This hands-on course will introduce students to the C++ programming language and is specifically designed for students who have had little to no programming experience in their previous studies.

With C++ still being one of the main languages for robot programming, I strongly believe that this course will prepare students for other ENPM robotics courses that require programming experience. This course will mainly focus on C++ programming. Towards the end of the course you will learn about the Robot Operating System (ROS) along with small exercises. Small projects will be assigned almost weekly to allow students to apply what they have learned in class.

Material

Recommended Text

Stroustrup, B. *The C++ Programming Language*. 4th Edition. Addison-Wesley Professional; (May 19, 2013). ISBN 978-0321563842.

Lippman, .S.B. and Lajoie, J. and Moo, B.E. *C++ Primer*. 5th Edition. Addison-Wesley Professional; (August 16, 2012). ISBN 978-0321714114.

O'Kane, J.M. *A Gentle Introduction to ROS*. CreateSpace Independent Publishing Platform. ISBN 978-1492143239 Online: https://www.cse.sc.edu/~jokane/agitr/

Other

Any required standards, ebooks, journal articles, and book chapters will be provided on Canvas.

Grading Scheme

Grades are not given, but earned. Your grade is determined by your performance on the learning assessments in the course and is assigned individually (not curved). If earning a particular grade is important to you, please speak with us at the beginning of the semester so that we can offer some helpful suggestions for achieving your goal.All assessment scores will be posted on the course ELMS page. If you would like to review any of your grades (including the exams), or have questions about how something was scored, please email us.Late work will not be accepted for course credit so please plan to have it submitted well before the scheduled deadline. We are happy to discuss any of your grades with you, and if we have made a mistake we will immediately correct it. Any formal grade disputes must be submitted in writing and within one week of receiving the grade.

22.22%	Quizzes $\times$ 5
26.67%	Real-world Applications $\times$ 4
51.11%	Final Project $\times$ 1

Final letter grades are assigned based on the percentage of total assessment points earned. To be fair to everyone I have to establish clear standards and apply them consistently, so please understand that being close to a cutoff is not the same this as making the cut ($89.99 \neq 90.00$). It would be unethical to make exceptions for some and not others.

Grades will follow the standard scale:

A+: $\geq$ 96 %; A: [93,96); A-: [90,93);

B+: [87,90) %; B: [83,87); B-: [80,83);

C+: [77,80) %; C: [73,77); C-: [70,73);

D+: [67,70) %; D: [63,67); D-: [60,63);

F: < 60 %

FAQs

Is this a ROS course?

Although you will learn about ROS in this course, this course does not focus on ROS. The main focus of this course is on modern C++ and its applications in robotics.

Do I need programming experience to take this course?

Prior programming knowledge is recommended but not required. You will be given guidance on how to approach programming and on how to write good code, good documentation, and on how to follow a given programming guideline.

What about the other ROS courses you are teaching?

809B is a C++ robotics course which focuses on the ARIAC competition. In this course you will have to write a control system for a Gazebo simulated industrial robot. 809E is a Python robotics course which focuses on mobile robotics. This course teaches Python and its applications with ROS and Gazebo with mobile robots.

Do I need Linux for this course?

Yes and no. 90 % of the course does not require a Linux environment. However, we will need to switch to Linux when we get to the ROS lectures. I would recommend that you have Linux installed from day 1 so you can get familiar with the OS. You will definitely use Linux if you plan to work in the robotics field and you should take advantage of this course to learn about Linux.

Learning Objectives

- Learn about basic C++ concepts.
- Apply coding convention guidelines and best practice.
- Write code documentation using Doxygen.
- Learn about ROS core components.
- Write C++ application to drive a mobile robot in ROS and Gazebo.
- Learn to write a search algorithm using object oriented programming.
- Apply search algorithm to solve a "rat in a maze" problem.

Activities, Learning Assessments, & Expectations

- Before class meetings, students need to understand the previous lessons as they need to put this knowledge into practice in the next class. At the beginning of the course, we expect the students to spend around 3 h on RWAs. As we approach the end of the course, many hours of coding are expected, and homework may take between 4 to 6 h. Quizzes will be performed solely in class and it is not allowed for absent students to take quizzes. Reports are written during students' free time.
- Class meetings are usually split between teaching a subject followed by exercises on the subject. It is crucial that all students bring a laptop since this course is hands on and most of the exercises are performed on the laptop. Attendance and participation are key components to perform the hands-on exercises and to present their homework. Attendance and participation are not graded but highly encouraged.
- Students who do not meet the deadlines will occur a penalty during the grading of their work. Late submissions should be avoided but exceptions can be made if more than one group needs an extension on their work submission.
- No quizzes or assignments will be dropped. The only document required in this course is an individual final report and there is no alternative to the grade of the report.
- The final exam consists of testing the students' code on a maze. The students will run their code in class and the performance of their work will be graded.
- Students are allowed to collaborate with each other on their group RWAs, final report, and final presentation. Students are not allowed to collaborate with each other nor use the slides when taking quizzes.
- Each student in a group needs to bring his knowledge and experience on the table. Meetings between students of the same group are crucial. To prevent students from not contributing to group projects, a peer review system will be put in place where each student grades the other students in her/his group based on their contributions.

Teaching Structure

Due to the current pandemic situation, only one on-campus lecture will be held in person in JMP2121. The remaining lectures will be provided live through zoom meetings.

Course-Specific Policies

Phones, tablets, and laptops can be used to take quizzes, which are hosted on the ELMS website. After the quizzes are taken, only laptops are allowed.

Campus Policies

It is our shared responsibility to know and abide by the University of Maryland's policies that relate to all courses, which include topics like:

- Academic integrity
- Student and instructor conduct
- Accessibility and accommodations
- Attendance and excused absences
- Grades and appeals
- Copyright and intellectual property

Please visit www.ugst.umd.edu/courserelatedpolicies.html for the Office of Undergraduate Studies' full list of campus-wide policies and follow up with me if you have questions.

Make-up Policy

Make-up exams or assignments will only be allowed for students who have a substantiated excuse approved by the instructor *before the due date* and the Office. Leaving a phone message or sending an e-mail without confirmation is not acceptable. Assignments, quizzes, and final project are mandatory.

Get Some Help!

You are expected to take personal responsibility for you own learning. This includes acknowledging when your performance does not match your goals and doing something about it. Everyone can benefit from some expert guidance on time management, note taking, and exam preparation, so I encourage you to consider visiting http://ter.ps/learn and schedule an appointment with an academic coach. Sharpen your communication skills (and improve your grade) by visiting http://ter.ps/writing and schedule an appointment with the campus Writing Center. Finally, if you just need someone to talk to, visit http://www.counseling.umd.edu. Everything is free because you have already paid for it, and everyone needs help... all you have to do is ask for it.

Diversity and Inclusivity Statement

I consider this classroom to be a place where you will be treated with respect, and I welcome individuals of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability - and other visible and non-visible differences. All members of this class are expected to contribute to a respectful, welcoming and inclusive environment for every other member of the class.

The University of Maryland recognizes the importance of a diverse student body, and we are committed to fostering equitable classroom environments. I invite you, if you wish, to tell us how you want to be referred to both in terms of your name and your pronouns (he/him, she/her, they/them, etc.). The pronouns someone indicates are not necessarily indicative of their gender identity. Visit trans.umd.edu to learn more.

Additionally, how you identify in terms of your gender, race, class, sexuality, religion, and dis/ability, among all aspects of your identity, is your choice whether to disclose (e.g., should it come up in classroom conversation about our experiences and perspectives) and should be self-identified, not presumed or imposed. I will do my best to address and refer to all students accordingly, and I ask you to do the same for all of your fellow Terps.

Class Schedule

Week	Due before class	During class meetings	Due after class
Week 1		Course Presentation and Introduction to C++	
Week 2		Structure of a C++ program	
Week 3		Program flow control	
Week 4	RWA01	Quiz01 Functions	
Week 5		Quiz02 Raw pointers and references	
Week 6	RWA02	Smart pointers and references	
Week 7		Quiz03 Object Oriented Programming – Part I	
Week 8		Object Oriented Programming – Part II	
Week 9		Quiz04 Process and Thread	
Week 10	RWA03	Path planning algorithm	Peer reviews
Week 11		The Robot Operating System (ROS) – Part I	
Week 12		Quiz05 The Robot Operating System (ROS) – Part II	
Week 13	RWA04	Class and Function Templates	Peer reviews
Week 14	Source code Slides	Final project review sessions	Peer reviews Final reports