



Software Development for Robotics

ENPM 808X Fall/2022

Course Overview

As the robotics industry continues to grow and evolve, software's role in these products and systems is also becoming more critical. From embedded controls to advanced perception and learning, software permeates today's robots. Building off domain expertise developed in other robotics courses, this course teaches the tools and processes to develop professional quality software for deployed systems and products. Students will learn the best practices of taking new ideas or prototypes, and understanding what it takes to build the complex software that is so important to today's commercialized robotic systems. The course is split into two parts: the first will review the C++ programming language, object-oriented programming (OOP) concepts, version control, testing, and agile software development processes; the second will introduce the popular Robot Operating System (ROS) framework with intensive programming assignments and projects. Students should be proficient in using Linux, programming with C/C++ and understand the concepts of object-oriented programming.

Learning Outcomes

After successfully completing this course you will be able to:

- Meet the rising expectations of robotics software engineers in industry and government.
- Design appropriate software architecture and debugging techniques for real-time robotic systems.
- Build a simulated robotic system and demo using ROS and Gazebo.
- Utilize Linux shell effectively and write competent bash scripts – two important and desirable skills for robotics software and system engineers.
- Apply industrial best practices to code, build, test, and maintain large scale C++ software.
- Design software projects using UML diagrams and utilize good Object Oriented Programming (OOP) practices and design patterns.
- Apply agile development, pair programming, and test-driven development (TDD) approaches to software implementation.

Dr. Tommy Chang
(He/Him)

Your email: TBD

Class Meets

Wednesday
7:00-9:40 p.m.

Location: TBD

Office Hours

Location: Virtual

Date and times: TBD

and by appointment

Teaching Assistants

Ameya Konkar
akonkar@umd.edu

Prerequisites

ENPM809Y or equivalent
Proficient with Linux OS
and command-line.
Proficient with version
control (Git or Subversion).

Course Communication

Time-sensitive information
sent via ELMS
announcements.

For assignment questions,
use the "Discussions"
section in ELMS.

Contact instructor via email
for other inquiries. Refer to
ter.ps/email for helpful
guidance on writing
professional emails.

Required Resources

Course website: <https://elms.umd.edu>

Readings:

- Software Engineering: The Current Practice, Vaclav Rajlich. CRC Press
- Accelerated C++: Practical Programming by Example, Andrew Koenig, Barbara E. Moo
- A Gentle Introduction to ROS by Jason M. O'Kane (available for free as PDF, or in print on Amazon)

Hardware/Software:

- Hardware: desktop computer or laptop
- Software: Ubuntu Linux 20.04, ROS1, ROS2, clangd, any C++ IDE / editor that supports Language Server Protocol (LSP) and runs on Linux. Popular editors include Visual Studio Code, Eclipse, Sublime Text, Emacs, Vim, etc. See a list of LSP clients at <https://langserver.org/#implementations-client>.

Supplemental Resources

Readings:

- C++ Primer (5th Edition), Stanley B. Lippman, Josée Lajoie, Barbara E. Moo, Addison-Wesley Professional. *Note: older editions do not cover C++11 features so go with 5th Ed. or alternative reference books.*
- C++ Cookbook, Jeff Cogswell, Jonathan Turkanis, Christopher Diggins, D. Ryan Stephens, O'Reilly Media, Inc., ISBN: 0596007612 (*available online on Safari, check if UMD has access*)

Hardware/Software:

- VMware virtual machine manager from Terpstore.
- VirtualBox (version 6 or greater).
- Microsoft Windows Subsystem for Linux (WSL version 2).

Campus Policies

It is our shared responsibility to know and abide by the University of Maryland's policies that relate to all courses. Please visit <https://academiccatalog.umd.edu/graduate/policies/academic-record/> for the Office of Graduate Studies' list of campus-wide policies.

Activities, Learning Assessments, and Expectations

Assignments (50%)

This course is focused on learning sound software engineering principles, **not** learning a programming language. You must have proficient programming skills to execute programming assignments while spending the majority of your time on course topics.

Mid-term project (20%)

This is a 3-week project. In the first-part, a project proposal, I will review it and provide comments before you begin working on it. Once I provide feedback and give you permission to proceed, you will have approximately two weeks to complete the rest of the project. You must work in a group of two or three of your choosing. The groups must implement the project as pair programming, while a group of three adds a third role: the design keeper. The design keeper extends what a navigator does in the traditional pair and makes sure that the implementation is following the project design (e.g. UML diagrams) along with good OOP practices and appropriate design patterns.

Final project (30%)

This is a 4-week project, beginning with a proposal stage, followed by 3 weeks of development and completing with an in-class presentation. I will review the proposal and provide comments and approval. Final submission includes presentation materials. Teams will present their projects in class. Late submissions will not be accepted. Ungraded submissions will be due each week to provide feedback. You must work in a group of two or three of your choosing. The groups must implement the project as pair programming and follow the same approach taken in the mid-term project.

Expectations:

Every effort has been made to evenly distribute the course requirements, and to support your understanding of the course material. However, it is likely that some weeks will require more effort on your part, and some material will require additional help beyond what is immediately available. Please reach out to me or your appropriate TA for these course-related questions, and please be prepared to put in the additional effort.

Course Specific Policies

Assignments are due by the beginning of the next class lecture (unless stated otherwise) and must be submitted following the assignment instructions. Some will be submitted through Canvas while others will be submitted through GitHub. No points for late submissions.

Accessibility and Reasonable Accommodations

The University of Maryland is committed to creating and maintaining a welcoming and inclusive educational, working, and living environment for people of all abilities. The University of Maryland is also committed to the principle that no qualified individual with a disability shall, on the basis of disability, be excluded from participation in or be denied the benefits of the services, programs, or activities of the University, or be subjected to discrimination. The University of Maryland provides reasonable accommodations to qualified

individuals. Reasonable accommodations shall be made in a timely manner and on an individualized and flexible basis.

Discrimination against individuals on the grounds of disability is prohibited. The University also strictly prohibits retaliation against persons arising in connection with the assertion of rights under this Policy.

Accessibility & Disability Service (ADS) facilitates reasonable accommodations to qualified individuals. For assistance in obtaining an accommodation, contact Accessibility and Disability Service at [301.314.7682](tel:301.314.7682), or adsfrontdesk@umd.edu. More information is available from the [Counseling Center](#).

Get Some Help!

You are expected to take personal responsibility for your 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> (there are specific resources for graduate students under handouts, but please explore to find what you need). Sharpen your communication skills (and improve your grade) by visiting <https://gradschool.umd.edu/graduate-school-writing-center> and schedule an appointment with the campus Graduate 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.

Names/Pronouns and Self Identifications

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.

Grades

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 me at the beginning of the semester so that I 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 me to schedule a time for us to meet in my office.

Late work will not be accepted for course credit so please plan to have it submitted well before the scheduled deadline. I am happy to discuss any of your grades with you, and if I have made a mistake I will immediately correct it. Any formal grade disputes must be submitted in writing and within one week of receiving the grade.

Learning Assessments	Weight
Assignments	50%
Mid-term project	20%
Final project	30%

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 as making the cut ($89.99 \neq 90.00$). It would be unethical to make exceptions for some and not others.

Final Grade Cutoffs									
+	97.00%	+	87.00%	+	77.00%	+	67.00%		
A	94.00%	B	84.00%	C	74.00%	D	64.00%	F	<60.0%
-	90.00%	-	80.00%	-	70.00%	-	60.00%		

Course Schedule

Week	Class Date	Topic
1	08/31	Create and share GitHub user account Linux command-line tools Compiling and Running Install Linux (Native or VM) Reading Assignment - Accelerated C++ Ch. 0-4 Reading Assignment - Commercial Systems in Practice Reading Assignment - Intro to Scripting
2	09/07	Compiling and running in a C++ IDE (any LSP client of your choice) Reading Assignment - Accelerated C++ Ch. 5-9 Reading Assignment - Software Engineering: The Current Practice ch. 1-4
3	09/14	Reading Assignment - Accelerated C++ Ch. 8, 10, 13 Reading Assignment - Software Engineering: The Current Practice ch. 5,6, 8, 9, 12, 13
4	09/21	Git Tutorial Reading Assignment - Git Reading Assignment - Software Engineering: The Current Practice ch. 10, 11,14, 15, 16 Test-Driven Development exercise (with groups!)
5	09/28	Mid-term proposal Reading Assignment - CMake and Valgrind Valgrind exercise
6	10/05	Mid-term project - Phase I

7	10/12	Mid-term project - Phase II
8	10/19	Programming Assignment: ROS Publisher/Subscriber Reading Assignment - Intro to ROS Ch. 1-3 Reading Assignment - ROS Wiki Intro ROSCon Presentation Summary
9	10/26	Programming Assignment - ROS Services, Logging and Launch files Reading Assignment - Intro to ROS Ch. 4-8
10	11/02	Programming Assignment - ROS TF, unit testing, bag files Reading Assignment - Intro to ROS Ch. 9 - Recording and Playback
11	11/09	Final Project Proposal (Phase 0) Gazebo Tutorials Programming Assignment - Working with Gazebo Rviz Assignment
12	11/16	Extra Credit: Google Mock Final Project (Phase I) Reading Assignment - Actionlib and SLAM
13	11/23	No class (Thanksgiving Recess)
14	11/30	Final Project (continued) Reading Assignment - Autonomous Navigation
15	12/07	Final Project (last sprint, presentation) (Optional) Programming Practice
16	12/14	ROS 2.0

Note: This is a tentative schedule, and subject to change as necessary – monitor the course ELMS page for current deadlines. In the unlikely event of a prolonged university closing, or an extended absence from the university, adjustments to the course schedule, deadlines, and assignments will be made based on the duration of the closing and the specific dates missed.