MARYLAND APPLIED GRADUATE ENGINEERING

Introductory Robot Programming (ENPM702)

Term: *Fall/2026*
Instructor: Zeid Kootbally
Pronouns: He/Him
Office Phone: N/A
Email: zeidk@umd.edu
Office Hours: Fridays (6 pm - 7 pm) + by appointment

Credits: 3
Course Dates: From Sep 1, 2026 - Dec 8, 2026
Course Times: Tuesdays, 7 pm – 9:40 pm
Classroom: JMP 2217

Course Description

ENPM702 is the programming foundation of the robotics graduate curriculum. Its purpose is to build enough fluency in modern C++ that the robotics software encountered later in the program, and in professional practice, reads as something a student can understand and change rather than something to work around.

The semester has two halves. Weeks 1 to 9 build C++ from the ground up: how memory actually works, pointers and references, the standard library, functions and templates, smart pointers and move semantics, and object-oriented design. Weeks 10 to 15 apply that language in ROS 2, moving from publishers and subscribers through services, actions, coordinate transforms and lifecycle nodes to a complete sense-plan-act autonomy architecture running in simulation. The second half depends throughout on the first. No topic is taught for its own sake.

All course code targets the C++20 standard and ROS 2 Jazzy Jalisco on Ubuntu 24.04.

There is no formal prerequisite and no prior C++ is assumed. Students are expected to have programmed before in some language and to be comfortable working in a terminal. For students arriving from Python, the first few weeks are more likely to feel unfamiliar than difficult: C++ requires explicitness about matters that Python handles silently, and that explicitness is the point.

A note on pace. The course moves quickly and each week depends on the one before it. Students who fall behind in Week 3 usually feel it in Week 9. Anyone who begins to struggle should contact the instructor early rather than late.

Learning Outcomes

After successfully completing this course, students will be able to:

- Write, compile and debug modern C++20 programs that use references, pointers, dynamic memory and the standard library, and diagnose memory errors using gdb and Valgrind.
- Apply RAII, smart pointers and move semantics to manage object ownership and lifetime correctly, and justify the choice between unique, shared and non-owning access in a given design.
- Design and implement class hierarchies that use encapsulation, inheritance, polymorphism and abstraction to model robot components and behaviors.

- Build ROS 2 applications in C++ that communicate through topics, services and actions, and configure them using parameters, launch files and executors.
- Integrate coordinate transforms, lifecycle-managed nodes and a decision layer into a working autonomous robot application in simulation, and explain the architectural trade-offs involved.

Required Resources

- Course Website: elms.umd.edu
- **Course documentation:** enpm702-fall-2026-docs.readthedocs.io. Lecture notes, weekly exercises, setup guides, self-study reading material and the course glossary. Required. Free.
- **Textbook:** None required, and nothing to buy. Recommended free references are cppreference.com, the C++ Core Guidelines, and the ROS 2 Jazzy documentation at docs.ros.org/en/jazzy. Further references are listed in the Reading Material section of the course documentation.
- **Operating system:** Ubuntu 24.04 LTS (Noble Numbat). Required. A native or dual-boot install is strongly recommended. A virtual machine is workable for the first half of the course but Gazebo simulation in the second half is graphics-intensive and will suffer. WSL 2 may not supported because of its GUI and Gazebo limitations.
- **Toolchain:** GCC 13 or newer (build-essential), CMake, gdb, Valgrind, Doxygen and Git. All required, all free, all installed from the Ubuntu archive. Course code is compiled with the C++20 standard.
- **Robotics software:** ROS 2 Jazzy Jalisco and Gazebo Harmonic. Required starting in Week 10, though installing them at the start of the semester is recommended so that problems surface early. Free.
- **Editor:** Visual Studio Code with the recommended extensions listed in the course documentation. Required. Free.
- **Accounts:** A GitHub account. Required. Free.
- **Hardware:** A computer able to run Ubuntu 24.04 with at least 8 GB of RAM, two or more CPU cores and 50 GB of free disk space.
- **Total estimated cost of required course materials:** $0.00

Setting up this environment is not optional preparation, it is the first thing the course depends on. Students should follow the Setup section of the course documentation before the first class and contact the instructor if anything does not work.

Course Structure

The course meets in person once a week for a single long lecture on Tuesdays. That format places most of the learning outside the classroom, so how the week between lectures is spent matters more here than in a course that meets three times a week.

Each lecture has a companion page in the course documentation containing the lecture notes, shell exercises, C++ exercises, a self-check quiz and references. The weekly exercises are the mechanism by which a lecture becomes a skill. They are best attempted in the days immediately after class, while the material is fresh, with anything that does not work brought to the following week.

The semester splits into two phases. Weeks 1 to 9 are individual work in C++, assessed through three Real-World Applications and five quizzes. Weeks 10 to 15 are teamwork in ROS 2, assessed through three Group Projects. Both sequences are cumulative: each assignment extends the code written for the previous one, so falling behind compounds rather than resolving itself.

Some material is assigned as self-study rather than lectured, because it is better learned by doing than by watching. The Reading Material section of the course documentation covers the Linux shell, VSCode and CMake, and version control before the semester begins; flow control and operations before September 15; and exception handling before October 13.

Students should expect to spend roughly four to six hours per week outside class on reading, exercises and assignments, with more in the weeks an assignment is due. That estimate includes time for environment problems. Toolchain trouble is normal in systems programming, and it always takes longer than expected.

Tips for Success in this Course

- **Participate.** I invite you to engage deeply, ask questions, and talk about the course content with your classmates. You can learn a great deal from discussing ideas and perspectives with your peers and instructor. Participation can also help you articulate your thoughts and develop critical thinking skills.
- **Manage your time.** Students are often very busy, and I understand that you have obligations outside of this class. However, students do best when they plan adequate time that is devoted to course work. Block your schedule and set aside plenty of time to complete assignments including extra time to handle any technology related problems.
- **Login regularly.** I recommend that you log in to ELMS-Canvas several times a week to view announcements, discussion posts and replies to your posts. You may need to log in multiple times a day when group submissions are due.
- **Do not fall behind.** This class moves at a quick pace and each week builds on the previous content. If you feel you are starting to fall behind, check in with the instructor as soon as possible so we can troubleshoot together. It will be hard to keep up with the course content if you fall behind in the pre-work or post-work.
- **Use ELMS-Canvas notification settings.** Pro tip! Canvas ELMS-Canvas can ensure you receive timely notifications in your email or via text. Be sure to enable announcements to be sent instantly or daily.
- **Ask for help if needed.** If you need help with ELMS-Canvas or other technology, IT Support. If you are struggling with a course concept, reach out to me and your classmates for support.

Policies and Resources for Graduate Courses

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 reference and review the Graduate School's Course Related Policies and follow up with me if you have questions.

Course Guidelines

Names/Pronouns and Self-Identifications:

The University of Maryland recognizes the importance of a diverse student body, and we are committed to fostering inclusive and equitable classroom environments. I invite you, if you wish, to tell us how you want to be referred to in this class, both in terms of your name and your pronouns (he/him, she/her, they/them, etc.). Keep in mind that the pronouns someone uses are not necessarily indicative of their gender identity. For more information about names and gender markers, visit the LGBTQ+ Equity Center.

Additionally, it is your choice whether to disclose how you identify in terms of your gender, race, class, sexuality, religion, and dis/ability, among all aspects of your identity (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.

Video or Audio Recording:

Video or audio recording of class sessions, activities, or discussions is prohibited without advance permission from the instructor. Maryland is one of 11 U.S. states that require two-party, or all-party, consent before it is permissible to record or otherwise intercept and/or capture communications (such as phone calls, in-person conversations, video conferences). This means that everyone participating in a conversation must give their consent before any communication is recorded. Read this UMD article on Audiovisual Recording for more information.

Communication with Instructor:

Email: If you need to reach out and communicate with me, please email me at zeidk@umd.edu. Please DO NOT email me with questions that are easily found in the syllabus or on ELMS (i.e. When is this assignment due? How much is it worth? etc.) but please DO reach out about personal, academic, and intellectual concerns/questions. While I will do my best to respond to emails within 24 hours, you will more likely receive email responses from me on Mondays, Wednesdays and Fridays from 8:00am-10:00am EST

ELMS: I will send IMPORTANT announcements via ELMS messaging. You must make sure that your email & announcement notifications (including changes in assignments and/or due dates) are enabled in ELMS so you do not miss any messages. You are responsible for checking your email and Canvas/ELMS inbox with regular frequency.

Communication with Peers:

With a diversity of perspectives and experience, we may find ourselves in disagreement and/or debate with one another. As such, it is important that we agree to conduct ourselves in a professional manner and that we work together to foster and preserve a virtual classroom environment in which we can respectfully discuss and deliberate controversial questions. I encourage you to confidently exercise your right to free speech—bearing in mind, of course, that you will be expected to craft and defend arguments that support your position. Keep in mind, that free speech has its limit and this course is NOT the space for hate speech, harassment, and derogatory language. I will make every reasonable attempt to create an atmosphere in which each student feels comfortable voicing their argument without fear of being personally attacked, mocked, demeaned, or devalued.

Any behavior (including harassment, sexual harassment, and racially and/or culturally derogatory language) that threatens this atmosphere will not be tolerated. Please alert me immediately if you feel threatened, dismissed, or silenced at any point during our semester together and/or if your engagement in discussion has been in some way hindered by the learning environment.

Major Assignments

Real-World Applications (RWA1 to RWA3)

- Three individual assignments building one cumulative C++ robotics application. RWA1 covers C++ fundamentals, RWA2 adds STL containers and functions, and RWA3 adds smart pointers and object-oriented design. Each builds directly on the code submitted for the previous one.

- RWA1 is posted September 15 and due September 29 (44 points). RWA2 is posted September 29 and due October 13 (48 points). RWA3 is posted October 27 and due November 10 (48 points).
- These are individual assignments. Discussion of concepts with classmates is permitted, but submitted code must be the student's own work. See the AI policy below for what tool assistance is allowed.
- Full specifications, rubrics and submission instructions are posted on ELMS-Canvas and on ReadTheDocs.

Group Projects (GP1 to GP3)

- Three team projects building one cumulative ROS 2 application running in Gazebo. GP1 covers publishers, subscribers and simulation; GP2 adds services, actions and robot class hierarchies; GP3 adds coordinate frames, lifecycle nodes and an autonomy decision layer.
- GP1 is posted November 10 and due November 17 (44 points). GP2 is posted November 17 and due December 1 (48 points). GP3 is posted December 1 and due December 11 (48 points).
- Teams are assigned on ELMS-Canvas. Collaboration within a team is expected on every part of these projects. Each team member is responsible for understanding the whole submission, not only the portion they wrote.
- Full specifications, rubrics and team assignments are posted on ELMS-Canvas and on ReadTheDocs.

Quizzes

- Five quizzes of 20 points each, given in Weeks 4, 6, 8, 10 and 12 (September 22, October 6, October 20, November 3 and November 17).
- Quizzes check that the concepts from the preceding lectures have been absorbed before the next assignment depends on them. They are short and are not designed to be tricky. Students who have completed the weekly exercises are already prepared.
- AI assistance is not permitted on quizzes.

Participation and Engagement

- Worth 20 points, earned week by week rather than as a single total. In each of the 15 weeks, participation is credited either by actively engaging in class discussion or by submitting that week's exercise solutions on ELMS-Canvas.
- The exercise route exists deliberately. Students for whom speaking up in a large lecture is not the best way to learn are not disadvantaged: the exercises are an equivalent path to the same credit.
- The score is proportional to the number of weeks credited: 20 x (weeks credited / 15), so each week is worth approximately 1.33 points.
- **Weekly submission is the point of this component.** Exercise solutions are credited only when submitted before the start of the following lecture. Week 15 solutions are due by the last day of class, December 11. Solutions submitted in bulk after the fact earn no participation credit: a batch uploaded near the end of the semester scores zero, regardless of how many weeks it covers or how well the work is done. The purpose of this component is to keep practice continuous across the semester, and a late batch does not serve that purpose.

Final Exam

- There is no final exam in this course. GP3 is the capstone assessment.

Grading Structure

Assignment	Percentage %
Real-World Applications (RWA1 to RWA3, 140 points)	**35%**
Group Projects (GP1 to GP3, 140 points)	**35%**
Quizzes (5 x 20 points, 100 points)	**25%**
Participation and Engagement (20 points)	**5%**

Total	**100%**

Academic Integrity

For this course, some of your assignments will be collected via Turnitin on our course ELMS page. I have chosen to use this tool because it can help you improve your scholarly writing and help me verify the integrity of student work. For information about Turnitin, how it works, and the feedback reports you may have access to, visit Turnitin Originality Checker for Students.

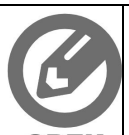
The University's Code of Academic Integrity is designed to ensure that the principles of academic honesty and integrity are upheld. In accordance with this code, the University of Maryland does not tolerate academic dishonesty. Please ensure that you fully understand this code and its implications because all acts of academic dishonesty will be dealt with in accordance with the provisions of this code. All students are expected to adhere to this Code. It is your responsibility to read it and know what it says, so you can start your professional life on the right path. **As future professionals, your commitment to high ethical standards and honesty begins with your time at the University of Maryland.**

It is important to note that course assistance websites, such as CourseHero, are not permitted sources unless the instructor explicitly gives permission. Material taken or copied from these sites can be deemed unauthorized material and a violation of academic integrity. These sites offer information that might be inaccurate, and most importantly, relying on restricted sources will hamper your learning process, particularly the critical thinking steps necessary for college-level assignments.

Additionally, students may naturally choose to use online forums for course-wide discussions (e.g., Group lists or chats) to discuss concepts in the course. However, collaboration on graded assignments is strictly prohibited unless otherwise stated. Examples of prohibited collaboration include: asking classmates for answers on quizzes or exams, asking for access codes to clicker polls, etc. Please visit the Office of Undergraduate Studies' full list of campus-wide policies and reach out if you have questions.

Finally, on each exam or assignment you must write out and sign the following pledge: ***"I pledge on my honor that I have not given or received any unauthorized assistance on this exam/assignment."*** If you ever feel pressured to comply with someone else's academic integrity violation, please reach out to me straight away. Also, ***if you are ever unclear*** about acceptable levels of collaboration, ***please ask***! To help you avoid unintentional violations, ***the following table*** lists levels of collaboration that are acceptable for each graded exercise. Each assignment will contain more specific information regarding acceptable levels of collaboration.

Permitted Resources and Collaboration by Assessment

Assessment						
Real-World Applications	✓	✓	✓	✓	---	---
Quizzes	✓	✓	✓	---	---	---
Group Projects	✓	✓	✓	✓	✓	✓

AI policy for this course

Generative AI tools (TerpAI, ChatGPT, Claude, Gemini, GitHub Copilot and similar) are part of professional robotics practice, and this course treats them that way. The policy below is written so that these tools can be used without undermining what the course exists to teach.

Permitted, with disclosure. On RWAs, Group Projects and the weekly exercises, AI tools may be used to explain concepts, interpret compiler and linker errors, review code the student has already written, suggest debugging strategies and generate small illustrative examples. When an AI tool has contributed to submitted code or design, the submission must carry a short disclosure at the top naming the tool and describing how it was used. Two or three sentences are enough. Disclosure carries no penalty. Undisclosed use is a violation of the Code of Academic Integrity.

Not permitted. AI tools may not be used during quizzes. Submitting AI-generated code that the student cannot read, explain and modify is not acceptable.

The standard applied. Each student is responsible for every line submitted. The instructor may ask, in class or in office hours, for a walkthrough of any part of a submission: why it works, what would break it, and what would be changed. Work that cannot be explained does not count, regardless of whether it compiles or how it was produced. This is the same standard a future employer will apply.

A caution worth taking seriously. These tools are strongest on the C++ and ROS 2 patterns, most common on the public internet, and that body of code skews old. Confident answers using ROS 1 idioms, deprecated rclcpp APIs and pre-C++11 memory management should be expected. Anything a tool produces must be verified against the course documentation and the official ROS 2 Jazzy documentation before it is relied upon. Learning to catch a plausible wrong answer is itself one of the skills this course is trying to build.

Students unsure whether a particular use is permitted, should ask the instructor before using it rather than after submitting it.

See the University of Maryland guidelines on Generative AI for Teaching & Learning for more information

Grades

This course is graded on a 400-point scale, weighted as shown in the Grading Structure table above: Real-World Applications 35 percent, Group Projects 35 percent, quizzes 25 percent, and participation 5 percent. Final letter grades include plus and minus descriptors and are assigned from the percentage of total points earned, using the cutoffs below.

All assessment scores are posted on the course ELMS-Canvas page as they are graded, so standing in the course can be tracked throughout the semester. Requests to review graded work, or questions about how something was scored, should be sent by email to the instructor, who will arrange a time to go through it.

Late work. RWAs and Group Projects are accepted up to 48 hours after the deadline with a 20 percent penalty, after which they are not accepted. Quizzes and participation cannot be made up. Extensions are granted for documented excused absences under University policy, and the instructor should be contacted before a deadline rather than after it. Because each assignment builds on the one before it, a late submission also costs time on the next assignment.

Regrade requests. A request to correct a score must be submitted in writing within one week of receiving the grade. Errors will be corrected immediately.

Final letter grades are assigned from the percentage of total points earned. Clear standards are applied consistently to every student, so being close to a cutoff is not the same as making the cut (89.99 is not 90.00). Exceptions are not made for some and not others.

Final Grade Cutoffs

Letter Grade	Cutoff
A+	97%
A	94%
A-	90%
B+	87%
B	84%
B-	80%
C+	77%
C	74%
C-	70%
D+	67%
D	64%
D-	60%
F	<60%

Course Outline

The schedule below maps each week to its lecture topic and the work due that week. Lectures meet on Tuesdays. The Thanksgiving recess (November 25 to 29) does not fall on a Tuesday, so no lecture is lost.

Tentative Schedule

Week #	Topic	Deliverable
Before Sep 1	Self-study reading: Linux shell, VSCode and CMake, version control	Reading material completed before the first lecture
1 (Sep 1)	L1: Course introduction and conventions	Lecture 1 exercises
2 (Sep 8)	L2: Introduction to C++ (variables, types, constants, namespaces)	Lecture 2 exercises
3 (Sep 15)	L3: Pointers and memory management (stack, heap, references)	Lecture 3 exercises; RWA1 posted
4 (Sep 22)	L4: STL containers (strings, arrays, vectors, iterators)	Lecture 4 exercises; Quiz 1
5 (Sep 29)	L5: Functions, basics (declarations, parameters, overloading, recursion)	Lecture 5 exercises; RWA1 due; RWA2 posted
6 (Oct 6)	L6: Functions, advanced (templates, lambdas, functors, std::function)	Lecture 6 exercises; Quiz 2
7 (Oct 13)	L7: Smart pointers and move semantics (unique_ptr, shared_ptr, weak_ptr)	Lecture 7 exercises; RWA2 due
8 (Oct 20)	L8: OOP basics (classes, objects, encapsulation, constructors)	Lecture 8 exercises; Quiz 3
9 (Oct 27)	L9: OOP advanced (inheritance, polymorphism, abstraction)	Lecture 9 exercises; RWA3 posted
10 (Nov 3)	L10: ROS 2 foundations (nodes, publishers, subscribers, custom interfaces)	Lecture 10 exercises; Quiz 4
11 (Nov 10)	L11: ROS 2 configuration and orchestration (parameters, launch files, executors)	Lecture 11 exercises; RWA3 due; GP1 posted
12 (Nov 17)	L12: ROS 2 communication patterns (services, actions)	Lecture 12 exercises; Quiz 5; GP1 due; GP2 posted
13 (Nov 24)	L13: ROS 2 coordinate frames and transforms	Lecture 13 exercises
14 (Dec 1)	L14: ROS 2 lifecycle nodes	Lecture 14 exercises; GP2 due; GP3 posted
15 (Dec 8)	L15: Robot autonomy architecture (sense-plan-act, where AI and ML fit)	Lecture 15 exercises; GP3 due Friday, December 11

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.

Resources & Accommodations

Accessibility and Disability Services

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 Accessibility & Disability Service (ADS) provides reasonable accommodations to qualified individuals to provide equal access to services, programs and activities. ADS cannot assist retroactively, so it is generally best to request accommodations several weeks before the semester begins or as soon as a disability becomes known. Any student who needs accommodations should contact me as soon as possible so that I have sufficient time to make arrangements.

For assistance in obtaining an accommodation, contact Accessibility and Disability Service at 301-314-7682, or email them at adsfrontdesk@umd.edu.

Emergency Preparedness

Emergencies on campus can happen at any time. To prepare, visit prepare.umd.edu or use the emergency symbol in the UMD App to review information. Resources for persons with disabilities are available on the emergency preparedness page of the ADA Coordinator's website.

Student Resources and Services

Taking personal responsibility for your own learning means acknowledging when your performance does not match your goals and doing something about it. I hope you will come talk to me so that I can help you find the right approach to success in this course, and I encourage you to visit UMD's Student Academic Support Services website to learn more about the wide range of campus resources available to you.

In particular, everyone can use some help sharpening their communication skills (and improving their grade) by visiting UMD's Writing Center and schedule an appointment with the campus Writing Center.

You should also know there are a wide range of resources to support you with whatever you might need (UMD's Student Resources and Services website may help). If you feel it would be helpful to have someone to talk to, visit UMD's Counseling Center.

Basic Needs Security

If you have difficulty affording groceries or accessing sufficient food to eat every day, or lack a safe and stable place to live, please visit UMD's Division of Student Affairs website for information about resources the campus offers you and let me know if I can help in any way.

Veteran Resources

UMD provides some additional support to our student veterans. You can access those resources at the office of Veteran Student life and the Counseling Center. Veterans and active duty military personnel with special circumstances (e.g., upcoming deployments, drill requirements, disabilities) are welcome and encouraged to communicate these, in advance if possible, to the instructor.

Notice of Mandatory Reporting

Notice of mandatory reporting of sexual assault, sexual harassment, interpersonal violence, and stalking: As a faculty member, I am designated as a "Responsible University Employee," and I must report all disclosures of sexual assault, sexual harassment, interpersonal violence, and stalking to UMD's Title IX Coordinator per University Policy on Sexual Harassment and Other Sexual Misconduct.

If you wish to speak with someone confidentially, please contact one of UMD's confidential resources, such as CARE to Stop Violence (located on the Ground Floor of the Health Center) at 301-741-3442 or the Counseling Center (located at the Shoemaker Building) at 301-314-7651.

You may also seek assistance or supportive measures from UMD's Title IX Coordinator, Angela Nastase, by calling 301-405-1142, or emailing titleIXcoordinator@umd.edu.

To view further information on the above, please visit the Office of Civil Rights and Sexual Misconduct.

Course Evaluation

Please submit a course evaluation through Student Feedback on Course Experiences in order to help faculty and administrators improve teaching and learning at Maryland. All information submitted to Course Experiences is confidential. Campus will notify you when Student Feedback on Course Experiences is open for you to complete your evaluations at the end of the semester. Please go directly to the Student Feedback on Course Experiences to complete your evaluations. You may access the evaluation reports for courses for which 70% or more students submitted their evaluations.

Copyright Notice