UNIVERSITY OF MARYLAND

SYLLABUS

# ENPM702 INTRODUCTORY ROBOT PROGRAMMING

FALL 2025

| | | | |
|---|---|---|---|
| **Section:** | 0101 & RO01 | **Professors:** | Z. Kootbally |
| **Credits:** | 3 | **Pronouns:** | He/Him |
| **Course Dates:** | 09/02/2025 - 12/12/2025 | **Emails:** | zeidk@umd.edu |
| **Course Times:** | Tue (7:00 pm-9:40 pm) | **Office Hours:** | By appointment |
| **Classroom:** | JMP 3201 | | |

## Course Description

ENPM702 is an advanced course focused on the practical application of C++ programming within the field of robotics. Through a combination of theoretical lectures and hands-on projects, students will gain a comprehensive understanding of C++ programming principles and techniques tailored specifically for robotics applications.

The course is divided into two main parts. In the first half, students will learn the fundamental concepts of C++ programming, including syntax, data types, control structures, functions, and object-oriented programming (OOP) principles. Special emphasis will be placed on understanding the intricacies of memory management using smart pointers and leveraging the power of the Standard Template Library (STL) for efficient coding practices.

During the second part of the course, students will transition into applying their C++ skills within advanced programming concepts and design patterns. Through hands-on exercises and projects, students will learn to develop robust, scalable software systems suitable for robotics applications. The course concludes with an introduction to the Robot Operating System (ROS) framework, providing students with foundational knowledge for future robotics coursework.

By the end of the course, students will have acquired the necessary skills to proficiently develop and maintain C++ codebases for robotics applications, with a strong foundation in software engineering principles essential for real-world deployment. Whether pursuing careers in robotics research, industry, or academia, ENPM702 equips students with the foundational knowledge and practical experience essential for success in the rapidly evolving field of robotics.

## Prerequisites

This course provides a thorough exploration of C++, starting from foundational principles and progressing to advanced concepts. While no prerequisites are necessary, students with previous programming experience will find the course content more approachable.

## Learning Outcomes

After successfully completing this course you will be able to:

- *Demonstrate Proficiency in C++ Programming*: Students will master C++ programming fundamentals, including syntax, data types, control structures, functions, and object-oriented programming (OOP) principles, enabling them to develop robust and efficient software solutions for robotics applications.
- *Apply Advanced Memory Management Techniques*: Students will utilize advanced memory management techniques, such as smart pointers, to effectively manage memory resources and mitigate common pitfalls associated with dynamic memory allocation, ensuring the reliability and stability of their C++ codebases in resource-constrained robotic environments.
- *Use STL for Efficient Coding Practices*: Students will employ the Standard Template Library (STL) to implement efficient and reusable algorithms and data structures, enhancing code readability, maintainability, and scalability in robotics projects. They will learn to leverage STL containers, iterators, and algorithms to streamline their C++ code and optimize performance in diverse robotic applications.
- *Implement Object-Oriented Design Patterns*: Students will implement object-oriented design patterns, such as inheritance, polymorphism, and encapsulation, to architect modular and extensible C++ codebases for robotics projects. They will analyze and apply design patterns tailored to specific robotic scenarios, fostering code reuse, flexibility, and scalability in software development.
- *Understand Robotics Framework Foundations*: Students will gain foundational knowledge of robotics frameworks and middleware, with an introduction to ROS concepts that prepares them for advanced robotics coursework and professional development in robotics software engineering.

## Required Resources

- Ubuntu Desktop 24.04 LTS: Noble Numbat (Recommended) or Ubuntu 22.04: Jammy Jellyfish
  - Please prioritize the installation of Ubuntu 24.04 LTS if possible.
- The first lecture will guide you through the installation and the setup of the following tools and software:
  - Visual Studio Code
  - Code documentation: Doxygen
  - Version Control:
    * Git
    * GitHub
- The following tools will be installed in subsequent lectures:

- Valgrind
- Visual Studio Code extensions.
- ROS 2 (for introductory coverage in final weeks):
  * Jazzy Jalisco (for Ubuntu 24.04)
  * Iron Irwini (for Ubuntu 22.04)

## Course Structure

This course includes both on-campus and online sections. To attend synchronously online, log into ELMS-Canvas at the time of the Section CP01 class (Wednesday at 5:15 pm) and select "Video Conference" from the left side menu. This will open a Zoom link to the live classroom.

For asynchronous online students, all lectures will be recorded and made available on ELMS-Canvas under "Panopto Recordings/Video Lectures" within 24 hours of the class me. Be sure to review the recorded lecture in a timely manner.

On-campus students come to class prepared to engage with the lecture and materials. Online students, be sure to log into Canvas regularly and participate in discussions and activities. Regardless of the section you are enrolled in, parcipation is expected.

**Please note that F1 students enrolled in the on-campus section are required to attend in person**. If you have a conflict on a particular day, please reach out to me in advance to discuss.

## Tips for Success in this Course

1. **Participate**: We 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 professors. Participation can also help you articulate your thoughts and develop critical thinking skills.
2. **Manage your time**: Students are often very busy, and We 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.
3. **Login regularly**: We 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.
4. **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.
5. **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.
6. **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 see the University's website for graduate course-related policies at: https://gradschool.umd.edu/course-related-policies

## 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. Visit lgbtq.umd.edu to learn more.

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.

### Communication with Instructor

- **Email**: If you need to reach out and communicate with us, 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 Thursdays from 8:00am-5:00pm EST.
  - Ensure that you format your email subject properly, following the example provided below. A clear and specific subject line is essential because I am teaching three courses this semester, and some students are enrolled in multiple courses I am teaching. A well-formatted subject line helps me quickly identify the course, section, and topic of your email, ensuring a faster and more accurate response. For example, a subject line could be: **"ENPM702 (Section CP01): Question about Assignment 2"**

- **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 us 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

- **Individual Assignments**: Four individual assignments are designed to help you reinforce and apply the concepts covered in class. These assignments will challenge your understanding, problem-solving skills, and ability to work through tasks independently. Detailed instructions and submission guidelines will be provided for each assignment. **Late submissions will incur a 10% deduction per day, with a maximum allowance of 3 days late. Assignments not submitted by the fourth day will receive a grade of zero.**
  1. **Assignment #1** (Released Week 2, Due Week 3): Modern C++ console-based user interface implementation. Students will develop interactive command-line applications using modern C++ features including input/output streams, string manipulation, and basic program structure to create user-friendly console interfaces.
  2. **Assignment #2** (Released Week 3, Due Week 4): Pointer fundamentals and memory management using traditional C++ pointers. This assignment focuses on understanding memory allocation and dynamic memory management using new and delete operators.
  3. **Assignment #3** (Released Week 4, Due Week 5): Standard Template Library (STL) implementation and utilization. Students will work with STL containers (vector, map, set, queue), iterators, and algorithms to solve complex data manipulation problems efficiently and demonstrate mastery of modern C++ programming practices.
  4. **Assignment #4** (Released Week 6, Due Week 8): Function design and implementation covering function overloading, default parameters, pass-by-reference, pass-by-value, lambda expressions, and function pointers. Students will create modular, reusable function libraries that demonstrate advanced C++ function capabilities.
- **Quizzes**: Quizzes provide an effective means to assess your understanding of the course material. They are administered at the beginning of class, either on Canvas or on paper, and typically last between 10 and 20 minutes. All quizzes are conducted under closed-notes conditions.
  - Students registered for in-person learning must take the quizzes in class.

    - Any quiz taken outside of class will not be counted, and a different quiz will need to be taken.
- **Participation & Engagement**: Active participation and engagement are essential components of this course. You are expected to contribute to class discussions, ask questions, and collaborate with peers during activities. Your involvement will not only enhance your understanding of the material but also create a dynamic and interactive learning environment.
- **Team Projects** (**Assignment #5** and **Final Project**)
  - **Components and Timeline**
    - **Assignment #5** – *C++ Implementation* (Released Week 10, Due Week 12): **Micromouse Maze Solver** - Teams implement a breadth-first search algorithm in pure C++ to navigate a micromouse through a maze to reach the center. This phase demonstrates integration of all course concepts including STL containers, smart pointers, and object-oriented design.
    - **Final Project** – *ROS Conversion* (Released Week 12, Due Week 15): Teams convert their C++ implementation to a ROS-based system with publishers, subscribers, and node architecture.
    - *Progress Check*: Week 14
  - **Project Details**
    - **Team Formation**: Week 9
    - **Team Composition**: Teams of 3–4 students.
    - **Objective**: Develop a complete micromouse maze-solving system, first in pure C++, then convert it to ROS. This two-phase approach allows teams to master the algorithm collaboratively before tackling ROS integration.
    - **Project Scope**: Teams will design and implement the breadth-first search algorithm using advanced C++ programming concepts, then refactor their code into ROS nodes implementing publishers and subscribers for maze communication, robot control, and path visualization.
    - **Grading**: Part 1 (C++ Implementation) - 35%, Part 2 (ROS Conversion) - 40%, Project Report - 25%
    - **Guidelines and Rubrics**: Detailed instructions are available on ELMS-Canvas under the "Team Project" section.

## Grading Structure

The grading structure is illustrated in Table 1.

| Assignment | Percentage % |
|---|---|
| Assignments | 30% |
| Quizzes | 20% |
| Team Project Package/Report | 50% |
| **Total** | **100%** |

Table 1: Grading Distribution

## 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, or AI-generated content 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 or biased 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.

| |  Open Notes |  Use Book |  Learn Online |  Gather Content with AI |  Ask Friends |  Work in Groups |
|---|---|---|---|---|---|---|
| **Assignments** | ✔ | ✔ | ✔ | — | ✔ | — |
| **Quizzes** | — | — | — | — | — | — |
| **Team Project** | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

Table 2: Guidelines for Task Completion

## Grades

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 and discuss.

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.

Final letter grades are assigned based on the percentage of total assessment points earned. To be fair to everyone we 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 ≠ 90.00). It would be unethical to make exceptions for some and not others.

| Final Grade Cutoffs | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **A+** | 97.00% | **B+** | 86.00% | **C+** | 76.00% | **D+** | 66.00% | | |
| **A** | 93.00% | **B** | 83.00% | **C** | 73.00% | **D** | 63.00% | **F** | < 60.00% |
| **A-** | 90.00% | **B-** | 80.00% | **C-** | 70.00% | **D-** | 60.00% | | |

Table 3: Final Grade Cutoffs

## Course Outline

**Note**: A tentative schedule is presented in Table 4. This schedule subject to change, so ensure you 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.

| Week # (Date) | Topic | Deliverables |
|---|---|---|
| #1 – Sep 2 | Course Overview + Tools Setup | |
| #2 – Sep 9 | Introduction to C++ | • **Assignment #1 Released** |
| #3 – Sep 16 | Normal Pointers | • **Quiz #1**<br>• **Assignment #1 Due**<br>• **Assignment #2 Released** |
| #4 – Sep 23 | Standard Library | • **Quiz #2**<br>• **Assignment #2 Due**<br>• **Assignment #3 Released** |
| #5 – Sep 30 | Functions (Part I) | • **Assignment #3 Due** |
| #6 – Oct 7 | Functions (Part II) | • **Quiz #3**<br>• **Assignment #4 Released** |
| #7 – Oct 14 | **Fall Break – No Class** | |
| #8 – Oct 21 | Smart Pointers | • **Assignment #4 Due** |
| #9 – Oct 28 | Object-Oriented Programming (Part I) | • **Team Formation** |
| #10 – Nov 4 | Object-Oriented Programming (Part II) | • **Quiz #4**<br>• **Assignment #5 Released (Micromouse Maze Solver)** |
| #11 – Nov 11 | Object-Oriented Programming (Part III) | |
| #12 – Nov 18 | ROS (Part I) | • **Assignment #5 Due**<br>• **Final Project Released** |
| #13 – Nov 25 | ROS (Part II) | |
| #14 – Dec 2 | ROS (Part III) | • **Quiz #5**<br>• **Project Progress Check** |
| #15 – Dec 9 | ROS (Part IV) & Course Wrap-up | • **Final Project Due + Report** |

Table 4: Tentative Course Outline – Fall 2025

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

## 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 or one of the many other mental health resources on campus.

### 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**'s website at ocrsm.umd.edu.

### 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 supports 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.

## Netiquette Policy [Optional]

Netiquette is the social code of online classes. Students share a responsibility for the course's learning environment. Creating a cohesive online learning environment requires learners to support and assist each other. To craft an open and interactive online learning environment, communication has to be conducted in a professional and courteous manner at all times, guided by common sense, collegiality and basic rules of etiquette.

## Participation

Given the interactive style of this class, attendance will be crucial to note-taking and thus your performance in this class. Attendance is particularly important also because class discussion will be a critical component for your learning. Each student is expected to make substantive contributions to the learning experience, and attendance is expected for every session. Students with a legitimate reason to miss a live session should communicate in advance with the instructor, except in the case of an emergency. Students who miss a live session are responsible for learning what they miss from that session. Additionally, students must complete all readings and assignments in a timely manner in order to fully participate in class.

## 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. By completing all of your evaluations each semester, you will have the privilege of accessing through Testudo the evaluation reports for the thousands of courses for which 70% or more students submitted their evaluations.

## Copyright Notice