

ENPM808X: Software Development for Robotics, Fall 2018

Instructor: David Pietrocola, dpiet@umd.edu

Office Hours: Tuesdays, 7:30-9:30 pm, Canvas Chat/email/Google Hangouts (subject to change). Or by appointment.

Schedule, textbooks, assignments, office hours are subject to change.

Consult course website for announcements and adjustments.

Prerequisite skills:

- ENEE 150 or equivalent
- Proficient with C++
- Comfortable with object-oriented programming (OOP) concepts and principles
- Comfortable with Linux OS and command-line
- Comfortable with version control (Git or subversion)

Objectives:

- Prepare students for the rising expectations of robotics software engineers in industry and government
- Strengthen students' portfolio of software projects

Textbook:

- 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)
- Optional references:
 - 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)

Learning Assessments:

Assignments	50%
Mid-term	20%
Final	30%
Total	100%

Lecture Availability and Assignments:

Unless otherwise stated or announced, lectures will be posted by Wednesday morning each week. Assignments are due no later than the following Tuesday at 11:59 pm, 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.

Week	Topic	Reading	Assignment
1	<p>Course Overview</p> <p>Recent history of software development in robotics industry</p> <p>Tools of the trade and product example walk-through (OS, language, build systems, debugging, scripting, IDE, version control, simulation)</p>		
2	<p>Review of C++, object-oriented programming, design patterns</p>		
3	<p>The robotics software development cycle</p> <p>Agile development</p>		
4	<p>Quality Control: Version control, unit tests, continuous integration with build farms, debugging with Eclipse IDE and gdb, profiling with valgrind</p>		
5	<p>Review testing and continuous integration, Build systems (cmake)</p> <p>Mid-term project discussion: focused on a small demo project that uses version control, documentation, OOP</p>		
6	<p>Overview of open-source</p>		

	software, licenses, how it affects robotics software development Mid-term project discussion		
7	Important libraries in robotics: Eigen, OpenCV, PCL, Boost Mid-term project discussion		
8	Mid-term project due Introduction to ROS		
9	No Class		
10	Review mid-term grades Additional ROS overview: launch files, logging, rqt_console, services, parameters, catkin build system		
11	ROS TF, URDF, rostopic, state publisher, recording and playback		
12	ROS for simulation and visualization (Gazebo and Rviz) Final project discussion		
13	Review of other useful ROS packages: Actionlib Navigation stack: mapping		
14	ROS Navigation Stack: autonomous navigation		
15	Discussion of final projects		
16	Project demos and presentations		

