# Overview

ME EN 497R is a research course that may be applied toward the technical elective requirement for the Mechanical Engineering program (and occasionally other programs, e.g. Applied Physics). The overall goal of the 497R course is to give undergraduates research experiences typically reserved for graduate students, albeit in an format accessible to the abilities and time constraints of an undergraduate student. In the FLOW Lab, under the direction of Dr. Ning, we have somewhat formalized this course while also allowing a great deal of student-lead learning. This syllabus document is the beginning of that process, wherein the undergraduate and graduate mentor work together to develop a customized syllabus for the undergraduate. This document is a template to be filled out and submitted to Dr. Ning for final approval and an add code for ME EN 497R. This paragraph should be replaced by an overview paragraph summarizing the rest of this document. Note that items in square brackets below should be replaced with the relevant information based on each student's custom course design. Also note, that there are several auxiliary items that every student will be expected to accomplish during their 497R experience. These are the items not within square brackets.

# Prerequisites

It is assumed that the student has completed the FLOW Lab Introductory Project, and have a basic familiarity with the following concepts:

- The Julia language, including using packages, functions, loops, doctstrings, comments, etc.
- LaTeX, including basic document creation, section headers, including figures, etc.
- Creating a github repository and using git for basic add, commit, pull, and push commands.

# Learning Outcomes

## Technical

1. [Technical Outcomes Here]
    - [futher details as needed]

## Auxiliary

The following auxiliary learning outcomes are based on those tools and methods of communication that are *universal* in the FLOW Lab. Note that there are other auxiliary skills that may be gained depending on your approach to your chosen course design.

1. Coding: The student, utilizing the Julia language, will learn the follwing:

- Docstring usage
- Good commenting practice

2. Version Control: Using git, the student will gain familiarity with the following concepts:
   - Using github Projects
   - Submitting and responding to github Issues
   - Opening and reviewing github Pull Requests

3. Technical Writing: The student will gain introductory experience in the following areas:
   - Reading and reviewing technical papers and creating a bibliography
   - Writing technical memos, a research proposal, and a technical report
   - Presenting technical information in well designed figures
   - Receiving and applying critical feedback

# Semester Schedule

As mentioned above, the semester will be broken down into 8, 2-week mini-projects. For each project, there will be a manditory auxiliary component along with your chosen technical component.

---

## [**Assignment 1 Title**]

*[date start - date end (Weeks 1-2)]*

### Technical Component

[Technical Assignment Description]

### Auxiliary Component

Since you will need a method to submit your assignments for the semester, we'll set that up here. As part of your first assignment, complete the following:

1. Create a repository on github called something along the lines of "497R Project" or something similar.

2. Add your graduate mentor as a collaborator so they can view your submissions throughout the semester

3. Create at least one project in your repository. Either for the semester, or perhaps one for the first term and another for the second. You decide what makes sense based on your course design.

4. Organize your repository in a way that makes sense for your course design. For example, you may consider adding a directory for memos/reports, a directory for code, etc. Keep in mind that you need files in those directories before git will recognize them.

5. Initialize a README.md file in your repository

6. In the README, write a summary of your project. You can use this syllabus for ideas of what to include. Make sure to include section headers and any other organizational elements to make it easy to read. (Think about how you could use this project as part of a portfolio.) You may also consider including a description of what your various directories include for easy reference later.

TIP: Remember that you can change any of this later, so don't worry about getting things perfect. More often than not, you'll update as you go and figure out better ways to organize, etc.

---

## [Assignment 2 Title]

*[date start - date end (Weeks 3-4)]*

### Technical Component

[Technical Assignment Description]

### Auxiliary Component

*Version Control:* For this, and every subsequent assignment, you will submit your deliverables using github. Thus, for every assignment from now on, you should do the following: (note that this seems involved, but it's really very simple in practice)

1. BEFORE you start coding, etc. create a github issue with a comment describing the problem description and any other information you want to include.

2. Make sure to add any assignments (assign yourself...) and labels (you may need to create custom labels) that make sense for your issue. Also add the issue to the relevant project. You may consider using multiple issues in a single assignment, depending on how you've designed your course. The Project Kanban boards can be helpful for organization.

3. Create a branch for what you're working on for this assignment.

4. Throughout the assignment period, feel free to add comments and questions in your issue (this is a good way to write things down that you want to ask your graduate mentor in your weekly meetings).

5. When you are ready to submit your assignment deliverables, create a pull request for your branch (this assumes you've been adding, committing, and pushing along the way as well). Request your graduate mentor as a reviewer.
6. Upon approval of your pull request, merge with a comment that closes the issue for that assignment, and delete your branch (unless you need it for something else).

REMEMBER: Do steps 1-3 before you start *this* assignment.

*LaTeX:* As you will be required to submit a memo, proposal, or report for this and each of the following assignments, so you will need to put together a LaTeXtemplate for this assignment. Your template should include at a minimum:

1. A well commented preamble with any packages you might need (for things like math, figures, hyperlinks, etc.)
2. A header with the assignment name/number, your name, and the date completed.
3. A footer with the page number.
4. Sections (with labels) for Introduction, Methods, Results, and Discussion/Conclusion.
5. Templates for basic figures and tables.

NOTE: This is a minimal template for this assignment. You will likely add to it later as your needs increase. If you do this well, you should be able to use your template for your other course assignments, especially final reports. Anecdotal evidence indicates that well formatted final reports such as those you can produce with LaTeXoften lead to significant grade increases.

You can obviously use this document as a reference for your template, but make sure you understand what everything does, and feel free to add your own personal flair.

## [Assignment 3 Title]

*[date start - date end (Weeks 5-6)]*

### Technical Component

[Technical Assignment Description]

### Auxiliary Component

## Prospectus

*[date start - date end (Weeks 7-8)]*

Each student will be required to submit a brief research prospectus including a simple literature review (at least 3 sources). See the prospectus in the undergraduate-onboarding repository template for additional information. Note that this can be moved around as needed to another bi-week block.

## [Assignment 4 Title]

*[date start - date end (Weeks 9-10)]*

### Technical Component

[Technical Assignment Description]

### Auxiliary Component

## [Assignment 5 Title]

*[date start - date end (Weeks 11-12)]*

### Technical Component

[Technical Assignment Description]

### Auxiliary Component

## [Assignment 6 Title]

*[date start - date end (Weeks 13-14)]*

### Technical Component

[Technical Assignment Description]

### Auxiliary Component

## Final Report and Code Submission

*[date start - date end (Weeks 15-16)]*

The Final Report is a culmination of the bi-weekly memos produced thus far, revised according to mentor feedback, as well as any additional material necessary to produce a technical report that is both complete and well executed. A final report template is available in the undergraduate-onboarding repository

The Final Code Submission is similarly the final, cumulative state of the code produced as part of the bi-weekly assignments, revised according to mentor feedback, as well as any additional material necessary to understand and use the code produced.

Both submissions will be due to the graduate mentor for review/grading on the last day of Finals.

## Grade Rubric

Table 1 includes the proprosed grading rubric for this course.

Table 1: Proposed grade division for ME 497R.

| **Points** | **Deliverable** |
|---|---|
| 10 | [Assignment 1 Title] |
| 10 | [Assignment 2 Title] |
| 10 | [Assignment 3 Title] |
| 10 | Prospectus |
| 10 | [Assignment 4 Title] |
| 10 | [Assignment 5 Title] |
| 10 | [Assignment 6 Title] |
| 10 | Final Report |
| 80 | Time Log |
| | |
| **160** | **Total** |

## Student/Mentor Acknowledgement

I, [the student], have helped write, and have read, and I understand the above proposed syllabus and accept it as my desired 497R experience.

______________________________ ______________________________

Student Signature Date

I, [the mentor], have read the above proposed syllabus and accept resposibility for providing guidance, feedback, and grades as required by this course plan.

______________________________ ______________________________

Mentor Signature Date