

MACALESTER COLLEGE

# COMP127: Project (Section 3)

[GitHub Classroom assignment URL](#) (updated S26)

[Course Website](#) (updated S26)

*Adapted from Amin Alhashim's project description*

## OVERVIEW

The course project offers you the opportunity to collaborate with peers to demonstrate mastery of course learning goals and develop software of your choosing that you would proudly present to others.

## GUIDELINES

Check the course website (linked at the top of the document) for details.

## TEAM SIZE

You are expected to work in a team of 3 students (or 2 if not possible).

## DOMAIN AREA

Each team is free to select any domain area for their project, ranging from image processing and data analysis to sound processing, simulation, gaming, art, and productivity. The sole requirement is that the software must be sufficiently sophisticated to engage in a meaningful discussion about it with others.

## DUE DATES

See the milestones section below.

## WHERE TO SUBMIT

Most deliverables are submitted via the project GitHub Classroom Assignment (this link is at the top of the document).

### Project Group GitHub Classroom Assignment

The project is a group GitHub Classroom Assignment, which means all team members share a single online repository. This setup is prone to merge conflicts when multiple people edit the same file and attempt to push changes simultaneously. To prevent conflicts, I strongly recommend that

each team member work in a separate file, eg, `amin.java`. The team must then coordinate to integrate the necessary pieces from each individual file into the final project submission.

The GitHub collaboration process can sometimes be challenging. If your team encounters any issues you cannot quickly resolve on your own, please notify the instructor immediately to prevent project delays.

## MILESTONES

### **Milestone 0: Team Formation** | due: Fri 4/17

Students will be assigned to teams based on their preferences. Students will meet to talk about project ideas.

### **Milestone 1: Proposal + Implementation Plan** | due: Wed 4/22

After creating their GitHub repository using the link at the top of this document, each team should submit a proposal for their project by creating a markdown file named `proposal.md` file in their project GitHub repository. Check the [proposal requirement \(html\)](#) on the course website for what should be submitted.

In addition, each team should include a rough implementation and responsibility plan, ie, what tasks need to be accomplished and who will do each when. Think about the list of deliverables when building your list of tasks. The plan should be added as GitHub Issues within the project repository—see the callout box below for details.

### **Milestone 2: Progress Presentations** | due: Mon 5/4

Teams should present their progress to class to solicit feedback.

- 5 minutes
- Demo
- Narrative (challenges / changes / insights)
- Revised plan (must be committed as `updated_plan.md` in your GitHub repo).

### **Milestone 3: Deliverables + Evaluation** | due: Sat 5/9

The list of deliverables for the project are as follows:

#### Code

Each team should push the code of their project to GitHub. Each repository should include a `README.md` file that answers the following questions:

- What is this GitHub repository all about?

- What software (with the version numbers) need to be installed to run the code contained in this GitHub repository, eg, openjdk 25.0.1 2025-10-21 LTS and VSCode 1.115.0
- What steps need to be taken to run the code contained in this GitHub repository? *Think about the steps you did at the beginning of the semester to prepare your machine for class.*
- What does the expected output look like? *You can use screenshots of the main windows of the software.*
- A presentation video as described below.
- A presentation slides as described below.
- What known limitations does the software currently suffer from, eg, known bugs or cases that the software can not currently handle?
- What resources were referenced while developing the software?

## README.md

The README.md file is the landing page for your GitHub repository which must be inviting and informative. The README.md file is a dynamic document that you need to keep up-to-date. See [the GitHub basic writing and formatting syntax page](#) for tips on how to format your README file. For inspirational README.md examples and a list of useful resources, see [the awesome README GitHub page](#)—the [Aimeos TYPO3 extension](#) and [SegwayJump-Game](#) are good examples.

## Presentation Video

After soliciting feedback from peers, teams are expected to adjust their software accordingly and then record a video presentation (5-10 minutes if possible) demoing the software and explaining its main technical features afterward, and describing the project narrative (what was challenging, interesting, and what changed along the way). The video should point the interested viewers to the code base if more information is needed. The presentation video should be uploaded to the project GitHub repository (if possible) or a video streaming service such as YouTube or Loom then embedded in the **README.md** file.

## Embedding Videos

See this webpage for [how-to embed videos](#).

## Presentation Report

The presentation report is a brief (one-page) document answering some specific questions about your project. The [report assignment details](#) are described on the course website.

## Presentation Slides

The presentation slides used in the presentation video (or an altered version) should be embedded in the report, preferably after uploading it to the project GitHub repository.

## Embedding Presentations

See this webpage for [how-to embed images](#)—this will also work on pdf files.

See this Google Help page for how-to [embed a Google presentation](#).

## Evaluation and Reflection

I will ask you to complete a peer and self reflection.

## IMPORTANT NOTES

### Teamwork

Working in a team is a chance to improve one's communication skills as well as know their teammates better. However, working in a team sometimes poses some challenges. To ensure successful project outcome, below are few expectations:

- Active participation—be present, attend classes, do your work, keep your team informed about any unexpected events
- Active listening—show interest in other team members' ideas
- Inclusive environment—invite teammates to participate
- Each member should be in all main aspects of the project, including coding, reporting, and presentation. It is not acceptable for a single member to solely handle one aspect, such as coding, while another focuses solely on the report, and another solely on the presentation.

### Code Backup

When working on your project, ensure that you commit your code frequently to GitHub, accompanied by meaningful commit messages, and push your changes regularly. This practice helps prevent any unforeseen issues or loss of progress.

### Code Styling

Teams should follow [COMP127: Code Styling Guidelines](#). This practice will ensure your code is more readable and enhance its maintainability.

### Software Robustness

The software must be resilient to user errors and should not encounter failures during execution, ie, it should be free from any bugs. Developing such software is a challenging task, requiring the team to adhere to the **incremental, test-driven development approach** discussed in class. It is important to prioritize robustness over features, as a software that is more robust with fewer features is preferable to one that is less robust with more features.