

Team Project

Each team will be using the agile method scrum to develop a calculator. The instructor will act as the Product Owner. Each team should attempt to follow the agile manifesto and/or scrum as closely as possible. Sprints will start and end on Mondays. We will use class time to do sprint reviews, sprint retrospectives, sprint planning and daily scrum. Code will be submitted via github. The calculator can be developed in the team's desired programming language (i.e. java, python, C#). Teams will use ZenHub for issue/story tracking. In class time will be used to perform sprint meetings. A majority of sprint execution (i.e. coding) will be done outside of class but feel free to do this work in class if time persists. Class time will be used as follows:

Monday

- "daily scrum"
- Sprint review
 - each team will demo any progress they have made and talk about what they accomplished
- Sprint Retrospective
 - Teams will answer the following questions
 - What went well?
 - What could have gone better?
 - What do we want to try next?
 - What puzzles us?
- Sprint planning
 - The Scrum team will review what items will be completed with the product owner

Tuesday

- "daily scrum"
- Backlog grooming
 - Switch estimation games each week.
- Sprint execution
- Product Owner questions

Sprint Schedule:

Sprint 1	17 th – 24 th
Sprint 2 (defect fix)	24 th – 1 st
Sprint 3	1 st – 8 th

Creating an Issue (yellow sections are required)

Create a new Issue Implement Example

Template None

Issue title

Implement Example

Write

Preview

As a < role>
I need <requirement or feature>
So that <goal / value>

Acceptance criteria / Definition of done (DOD)

Attach files by dragging and dropping, selecting them here, or pasting from the clipboard.

Styling with Markdown is supported

Create an Epic

Submit new Issue

Pipelines

CS-271
New Issues

Labels

No Labels yet

Assignees

ajputnam

Milestone

No Milestone

Estimate

1

Epics

Not inside an Epic

Releases

Sprint 1

Git commit example.

Each git commit should be associated with a task in [github/zenhub](#). The commit message should start with [Issue #] followed by a description of the change. If the task requires multiple code changes/pushes reuse the same issue.

```

MINGW64:/c:/Users/Aj-Pu/Documents/git/HU-CS-271-Team-X
- □ ×

[Issue 3] Adding an example of a code change.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Tue Jun 18 04:26:19 2019 -0600
#
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
# (use "git push" to publish your local commits)
#
# Changes to be committed:
#   new file:   example.txt
#
~
~
~
~
~
cuments/git/HU-CS-271-Team-X/.git/COMMIT_EDITMSG [unix] (04:27 18/06/2019)1,1 All

```

Project Backlog:

- Get the results from a mathematical expression (i.e. $1+5/4$)
- Support: add, subtract, multiply, divide, exponential, factorial
- PEMDAS support
- Support the following functions
 - Factor(n)
 - Sqrt(n)
- Invalid user input (error handling)
- Matrix Math (add, subtract, multiply)
- Fractional support ($1/3 + 1/3 = 2/3$)

Project requirements

- Each story/feature should have at least 1 test associated with it
- Each team member will be required to create and complete at least 1 **task** and code change per sprint
-

Sprint Rubric

Creating stories from the “backlog”. Breaking stories down into tasks and assigning them to a team member and “committing” to them	25
Submitting code with proper git commit message and tagging.	25
Attending and participating in team meetings	25
Overall team performance, code quality, tests and “product”	25