- Standup link: https://osu.zoom.us/rec/share/ktKVqPQRO3cdCTcmbJ_3G3_oEje5qRBe6Dbk2CKob8qR_JB2pwbgEH_yU3E9yt4C.7eY7WackCB2jXhwR
- A detailed report detailing the state of the source code including quality of the source code and implementation, documentation, testing, and team meetings.
    - Sam Cubberly - Right now, we have a calculator that is able to have numbers be put into the screen. Operators can also be viewed on the screen. The first sprint wasn't evenly allocated, and some people wrote parts of the code that wasn't in their allocation. Right now we have a lot of repeated code, but by the next sprint everything should be ironed out. The computing of the equation still needs to be fixed to accommodate functions, and other operations, and be implemented correctly. Also, memory and history need to be added. Testing needs done also.
    - Oliver Shen - Everyone is doing their best to advance and improve this project. The part I'm contributing is related to compute functions only considered the logic of the keys themselves before. Now, it needs to be further improved. For example, when I click on exp, it should display exp[on the screen, which means we also need to have the button]. Therefore, the logic of the arr.splice corresponding to functions also needs to be adjusted. In addition to these, we have a lot of code that can actually be merged into one file. For instance, we can move the contents of the corresponding funs.js, ops.js and other keys to the existing interacting.js to ensure to reduce the overlap of the code.
    - Yunfeng Wang - at this moment we have a calculator version that is functioning partially, as it now has all the components it should have, we're doing to make it fully functioning. Also, for this project, the quality of code and documentation is pretty good as everyone has contributed their stuff as arranged in sprint 1. Based on the team meeting held this morning, we have a sprint 2 that aims to fix all the problems we had on this project. Also, we need to do more testing on this project.
    - Anshuman Ranjan - Currently, the physical design of the calculator app is mostly completed- only the history/memory containers will be redesigned. All the buttons are active and working, but their corresponding javascript functions still need to be finished. We also need to implement all the methods for the history and memory functionality of the calculator. The back-end compute function (which is our version of the eval function) can currently parse and compute all the PEMDAS operations, but still needs a proper implementation of including the function calls (sqrt, mod, etc)

- Are the contributions and workload balanced across all team members? Why or why not?
    - Sam Cubberly - Yes, I believe the contributions are fair now.
    - Oliver Shen - Yes, everyone’s work contributes evenly.
    - Yunfeng Wang - Yes, everyone’s workload and contributions are balanced.
    - Anshuman Ranjan - Yes, everyone’s workload and contributions are balanced.
- Did every team member read the submitted update report and look at all of the source code (not just their own code) in the Team's repo on Github before the due date for this part of the assignment?
    - Sam Cubberly - Yes, I did.
    - Oliver Shen - Yes, I've read all.
    - Yunfeng Wang - Yes, read all.
    - Anshuman Ranjan - Yes, read all.
- For each team member, provide what JavaScript language elements are found in the source code contribution from that team member. What JavaScript language elements (from the lecture notes) are used in the solution from below?
    - Sam Cubberly
        - Used array functions in JS
            - Splice, indexOf, slice
        - Function calling in JS
        - Function definition in JS
        - Loops in JS
        - Used addEventListener
        - document.getElementById
        - Document.write

    - Oliver Shen
        - Used array functions in JS
            - indexOf, splice
        - Math library function calls
            - Math.pow, Math.sqrt, Math.abs, Math.exp, etc.
        - Used object method calls
            - expression. add_op, expression.to_s
        - Used variable declaration const
        - Used addEventListener
        - Used HTML custom data attributes data-value

    - Yunfeng Wang

  - Used const and let to declare variables,
  - Used addEventListener
  - Used DOM manipulation
  - Used array functions
  - Used if, else if, else conditionals
  - Used static methods
  - Used String.join and Math.pow
- Anshuman Ranjan
  - Used const and let to declare variables
  - Used addEventListener and DOM manipulation
  - Used a Switch-Case statement for individual button classes
  - Used custom HTML data attributes that are specific to JS (data-value, id, etc)
  - Utilized console.log for debugging
  - getAttribute, querySelectorAll, .textContent