CS130 Project 2 - Design Document
==========

Please answer all questions in this design document. Note that the final feedback section is optional, and you are not required to answer it if you don't want to.

Unanswered or incompletely answered questions, or answers that don't actually match the code/repository, will result in deductions.

Answers don't have to be deeply detailed! We are mainly looking for an overview or summary description of how your project works, and your team's experiences working on this project.

Logistics (7 pts)

L1. [2pts] Enumerate all teammates here.
Kevin Do, Eli Kugelsky, Zack Dugue, Purvi Sehgal

L2. [2pts] What did each teammate focus on during this project?
Kevin - Reworking Project 1, Cell Notifications
Eli - Reworking Project 1, Performance tests
Zack - Update, Rename, Copy sheet, performance Tests
Purvi - Saving/Loading JSON files, Cell Notifications, Saving/Loading tests

All parties worked to write tests for project 1 to make sure our buggy code from before was working. And we had early design meetings for project 2 in which all team members had input about implementation of big features like cell notification.

L3. [3pts] Approximately how many hours did each teammate spend on the project?

Approximately 9 hours per person

Spreadsheet Engine Design (11 pts)

D1. [3pts] Briefly describe how your workbook-loading code operates. Does it do anything sophisticated to optimize the performance of loading a workbook, such as deferring cell-value update calculations, or analyzing the graph of cell dependencies?

The workbook-loading code iterates through the .json file, goes through every sheet, and updates the cell contents for each of the cells in the sheet. It does not do anything sophisticated yet, but this will be incorporated into the next project.

D2. [4pts] Sheet-level operations like copying a sheet, deleting a sheet, renaming a sheet, or even creating a new sheet, can cause cell values to be updated. How does your workbook identify such cells and ensure that they are updated properly?

For now, the workbook goes through every cell that has been created either by reference or by setting cell contents, and check if the cells sheet name corresponds to the new sheet name, and then we just call update cell

In a future approach, we would be only be searching for cells in the sheet, so that we don't have to go through every single cell in the workbook

D3. [4pts] When renaming a sheet, cells with formulas that explicitly reference the renamed sheet must be updated with the new sheet name. Give an overview of how your implementation updates these formulas.

Our workbook has a set that contains every cell in the workbook that is referenced by another cell. We then iterate through the cells in this set of referenced cells that are also in the sheet whose name is being changed. Then we go to the parents of those referenced cells and replace their formulas with a new formula that has the updated reference. We use the lark transformer to update the formula names (including removing unnecessary single quotes). We then use the lark reconstructor to reconstruct the string from the parse tree. Note that the new string is totally missing spaces. Donnie said this was okay, but in the future we'll probably want to make this reconstruction a little bit more readable for the user!

Informal Design Reviews (16 pts)

R1. [4pts] What insights did your team gain regarding the design of *your own* spreadsheet engine code? What parts of your design are you happy with? What parts might require further attention in the future?

We're not really happy with the graph implementation of our code for the most time. We have the dependency list, and we have parents and children of our code stored in cells themselves, and we have to iterate through every cell referenced and created for operations like new/rename/del/copy sheet.

After talking to Donnie, we also realized that we can reduce coupling in our workbook by having the parser be a global variable, and that way the workbook doesn't depend on the parser.

After talking to some other groups, we realized that there might just be better ways of doing the graph dependency, like just storing all the names of the cells in a graph-like structure.

We're pretty happy with our sheet class, as that doesn't really need to get touched at all

R2. [4pts] Did you feel like you were effective at helping other teams assess *their* software designs? Briefly discuss what went well, and what could have gone better, in your interview of another team.

The team that we interviewed sounded like they had everything together and they were really happy with their work. We don't think that we helped very much honestly, but we also don't think that anything could have gone better. However as a result of the interview, we have a some idea of how to integrate what they did into something that we can do

R3. [4pts] How closely did your team's design match the designs of the other teams you talked with? Briefly discuss significant similarities and differences between your team's approach and other teams' approaches.

Some parts, like making a cell class, sheet class, and the workbook class was all consistent. We all felt that it was pretty necessary to do this to have the cleanest code possible.

The team that we did interview had an enum type to represent formulas, numbers and strings, which was pretty interesting, as we were considering making whole cell type classes, but deemed it not necessary. We still don't think that the enum type isn't really necessary either way, because we can just kind of cast numbers and strings and have an is formula function.

Everyone represented their graphs differently, and we thought that was kind of interesting.

R4. [4pts] Which GRASP principles were the most pertinent in your discussions? How much of your discussions referenced the GRASP principles?

Part of our discussions referenced the GRASP principles, as we were saying how our design had some coupling issues. When we had a code review with Donnie, he mentioned how the parser work although was handled in a separate file, resulted in some coupling as the workbook references the parser, and the parser represents the workbook

Performance Analysis (16 pts)

In this project you must measure and analyze the performance of two central areas of your spreadsheet engine. Using pair programming, construct some performance tests to exercise these aspects of your engine, and use a profiler to identify where your program is spending the bulk of its time.

A1. [4pts] Briefly enumerate the performance tests you created to exercise your implementation, along with the teammates that collaborated to implement each of them.

Zack Dugue and Eli K. worked together on the performance tests.
We tested:

- Creating a bunch of sheets repeatedly
- The Speed of Workbook set contents for random strings.
- The Speed of writing a chain formula (IE each cell is just = the prior cell).
- The Speed of updating a chain formula (IE how fast does the chain update if the prior cell is set to a value).
- The speed of cycle checking on a chain formula (IE I make and break a cycle repeatedly by connecting the last cell of the chain to the first).
- The Speed of writing, updating, and cycle checking, with a formula pyramid (the formula is essentially a cumulative sum over prior cells).
- Writing a test which makes many small cycles that all depend on the same cell. Then make and break the cycle over and over again.
- Copying a large spreadsheet with many entries.

A2. [2pts] What profiler did you choose to run your performance tests with? Why? Give an example of how to invoke one of your tests with the profiler.

We chose to use cProfiler because it didn't seem like our code is complicated enough to justify having to use a sampling profiler (like i don't think any of the tests are compute heavy enough to have us worry about overhead).
The profiler is specifically integrated into the tests, so you call the test and then it returns a profiler object along with printing out the total time taken, computed with the profiler. In the future we want to make this profiler code more dev friendly, by integrating it with the pytest framework.

A3. [6pts] What are ~3 of the most significant hot-spots you identified in your performance testing? Did you expect these hot-spots, or were they surprising to you?

During our profiler test for updating the chain formula (IE updating the value at teh beginning of the chain). We found that the Lark Parser took up a majority of the time here. This is interesting because our profiler specifically only tracks the system time for after the first cell in the chain

has been changed. But that first cell is being changed from a decimal to a decimal, so the lark parser shouldn't even be getting called here. It's very interesting and kinda weird that we're spending so much time in a piece of our code that isn't even relevant to the task! We look forward to debugging this over the weekend.

Updating cell values in our pyramid shows we have a hot spot in our "Recompute cell value function". This makes sense given the nature of the pyramid being a cumulative sum. There are a lot of dependencies to go back through and so it makes sense that we would spend a lot of time in that function for this test.

For many of the profiler "get_cell_value" is a method we spend a lot of time on! This is a fun example of a "hot spot" we really can't do anything about. Get cell value is literally the lowest level possible field access, and so despite our profiler saying it is a large part of the cumulative time of our program, there's really not much to be done!

A4. [4pts] Reflect on the experience of pair-programming as you constructed these tests. What went well with it? What would you like to try to do better in the future?

Pair programming the tests resulted in better tests since more edge cases were considered, which otherwise may not have been tested. In the future, we would like to have people who have not worked on the code, write tests for the code. We could split into groups of two, have two people write half of the code, test it, and give it off to the other group of two to write more tests. This way the tests aren't influenced by bias. Specifically in the case of these profiler tests, it was really hard to have pair programming because me and him worked on different parts of the project, and had different ideas about what areas should be stress tested and where we thought hot spots might be.

Section F: CS130 Project 2 Feedback [OPTIONAL]

These questions are OPTIONAL, and you do not need to answer them. Your grade will not be affected by answering or not answering them. Also, your grade will not be affected by negative feedback - we want to know what went poorly so that we can improve future versions of the course.

F1. What parts of the assignment did you find highly enjoyable? Conversely, what parts of the assignment did you find unenjoyable?

I think the interview was fun!

F2. What parts of the assignment helped you learn more about software engineering best-practices, or other useful development skills? What parts were not helpful in learning these skills?

I think the profiler test thing should honestly be a separate little independent problem assignment. Seriously. We have 0 experience with profilers and so it would be nice to be handed a program that already runs well and works well (like obvs not a spreadsheet engine or whatever) and then use the profiler on that to see where the hot spots are and stuff. It's just a bit harder with your own code to tell if the code is the problem or your understanding of the profiler is the problem.

F3. Were there any parts of the assignment that seemed _unnecessarily_ tedious?
(Some parts of software development are always tedious, of course.)

We don't understand the part about removing redundant single quotes off of spreadsheet names. It feels very out of the blue and kinda unnecessary as a user feature. Like if we develop a spreadsheet engine to sell to users a spreadsheet there is zero universe I would implement this before implementing custom functions.

F4. Do you have any feedback and/or constructive criticism about how this project be made better in future iterations of CS130?

This project can be made better in future iterations by specifying how the file will be opened before loading/saving via with open or via StringIO or another technique. Also, one of the parameters to the load/save functions was fp, which was confusing since the input is not a file path, but rather, the actual file object. Other than that, the spec was pretty clear!