Spreadsheet Proposal Memo
To: Jason^2
From: Jason Hemann
Date: September 13, 2018
Subject: Specifications for Spreadsheet

Overview:

        We are looking for a module that will provide us with a class that is able to model the behaviour behind a spreadsheet. This module should be named `spreadsheet` and written using Python 3.6.3. This class should be able to easily be used to create an API for any number of relevant tasks to be called by a controller of any variety. The class should be able to accept a CSV file for the initial creation of the spreadsheet and use a hashmap for its data storage . Finally, this class should be easily extensible for all further addition of functionality.

Specifications:

- Cells will be denoted by a tuple, in the form of `(x,y)` representing its coordinates
- The class will be called `Spreadsheet`
    - This will contain a dictionary for the underlying data structure of the spreadsheet
        - Key will be the tuple representing the cell's coordinates
        - Value will be the value of the cell, unevaluated
    - Will have `create_spreadsheet(path)` which takes in the path of the input CSV
    - Will have `get_value(cell)` which takes a tuple referring to a cell
    - Will have `put_value(cell, value)` which takes *cellRef* and a string representing a *cellVal*.
- Initial data input will be in the form of a CSV
    - Each value must contain a valid *cellVal*

```
cellVal    :=    formula
           |     cellRef              int        :=    digit
           |     number                          |     onenine digits
           |     ''

                                      frac       :=    ''
formula    :=                                    |     '.' digits
     '(' cellVal op cellVal ')'
                                      digits     :=    digit
op         :=    '+'                              |     digit digits
           |     '*'
                                      digit      :=    '0'
cellRef    :=                                    |     onenine
     '(' int ',' int ')'
                                      onenine    :=    '1' . '9'
number     :=    int frac
```