- Make matrix class
- README file

Class

- use no STL containers (vectors)
- make use of dynamic arrays, new, and delete
- matrix with i rows and j columns
- using two dimensional arrays (double ** data)

start by writting a constructor that takes two unsigned ints: a row count and column count, and a double fill value.

- If either row or column is zero should return 0x0 matrix
- All starting values start at fill value

Suport

- equality operator "==" → all indicies must be equal
- inequality operator "!="

Other implementations

- num_rows()
- num_cols()
- clear() → makes matrix an empty matrix
  - → makes rows and columns zero
  - → deallocates any memory currently held by the Matrix
- get() → takes a row, column, and a double
  - ↳ if row and column are in boundry
    - ↳ the value at a[row,col] should be stored in the double and return true
    - ↓ otherwise return false
- set() → takes in a row, column, and double value
  - ↳ in bounds return true and set to value
  - ↳ if not in bounds return false
- operator << → allows to print
  - ↳ watch video on this

Simple Matrix Operations

- multiply_by_coefficient()
  - takes a double coefficient
  - multiplies every element by coefficient

- swap_row()
  - takes two unsigned int
    - source row and target row
  - if both inside matrix switch values of two rows and return true
  - else return false

- transpose()
  - type void
  - turns $m \times n \rightarrow n \times m$
  - $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$

Binary Matrix Operations

- add()
- subtract()
  - needs to be same dimension
  - takes in one argument (another matrix)

Harder Matrix Operations

- get_row()
- get_col()
  - take in unsigned int
  - return double *

- quarter()
  - takes in no argument
  - splits matrix into quarters
  - returns matrix *