Assignment 0326
This assignment represents the next building block in your 3D graphics arsenal—a solid matrix library. It has a short turnaround because it is fairly plug-and-chug, plus the semester’s schedule makes it ideal to have this done before the Easter break.

Outcomes
This assignment will affect your proficiency measures for outcomes 2a, 2b, 3d, 3e, and 4a–4f.

For Submission
You will want to base this assignment on the vector bazaar sample. Structure your repository and write source code so that your pipeline scene uses your matrix code without needing redundant copies.

Enter the Matrix
Design and implement a computer graphics matrix library, matrix4x4.js. Include:

- A basic Matrix4x4 object that initializes, by default, to the identity matrix
- A multiply function which multiplies two Matrix4x4 objects and returns the result (as a Matrix4x4 object, of course)
- A translate function which takes three parameters dx, dy, and dz, returning a Matrix4x4 object that accurately represents this transformation
- A scale function which takes three parameters sx, sy, and sz, returning a Matrix4x4 object that accurately represents this transformation
- The rotate function given in the sample code, but refactored to fit your Matrix4x4 object
- The ortho projection function given in the sample code, but, as with rotate, refactored to fit your Matrix4x4 object
- A frustum projection function based on the matrix derived from the course handout
- Conversion/convenience functions to prepare the matrix data for direct consumption by WebGL and GLSL.

Demonstrate your library’s correctness with a unit test suite based on QUnit (as demonstrated by the vector example). If you like, you may also start using the library in your fledgling scene.

Commit and push your work to your git repository, under a location of your choosing but set up as described in the top paragraph.