Project Requirements

Turn in your projects using the D2L Dropbox. You may turn in your projects multiple times before the project deadline. Only the last version will be used for grading.

Due Date

Late labs will be docked 10% per class day. Labs will be counted on time if they have a date on or before the assigned due date and time. Labs will not be accepted after graded labs have been returned.

Correctness (at least 50% of the total lab grade but possible up to 70%)

If your program produces syntax errors, it will not be graded. When syntax errors occur, I will inform you immediately via e-mail in hopes that problems occurred in transmission.

Most graphics labs do much more than produce a picture. They are usually interactive in nature and the user is allowed to make a variety of choices. When I grade your lab, I test the program thoroughly to make sure that it meets the specs and produces the correct images. You should thoroughly test your program before turning it in. Partial credit will be given for a partially working program.

Documentation (up to 20% of the lab grade)

Each of your source files should include a heading at the top of the program with the following information:

FILE NAME:	cube.html
PROGRAMMER:	Your name
CLASS:	CSCI 4250/5250
DUE DATE:	date in the form Tuesday, 9/12/2014
INSTRUCTOR:	Dr. Li

The heading should also include the following in **complete correct sentences:**

- a short description of the program
- a description of user input if any
- a description of expected results
- tips for using the program
- limitations – how much of the program is working and what limitations do you perceive in the program

In addition to this beginning documentation, each section of the program should be fully documented. A heading that contains the following should precede each function or method:

- the name of the function/method to follow
- the purpose of the function

All variables **must** be briefly described (this includes local variables as well as function parameters). Comments beside variables should be aligned as much as possible in a column.

Enough comments should be spread throughout the program so that the reader can easily follow the logic.

Maintainability (up to 30%)

- Structured programming techniques should be used. Each function should have a single well- defined purpose.
- Use proper indentation and be consistent with your indentation Blank lines should be used for readability
- Use descriptive variable names
- Global variables are normally strongly discouraged. However, when using WebGL, it may be necessary to use some global variables. When it is absolutely necessary, you can use global variables. However, avoid them when possible!

Efficiency (up to 20%)

Care should be exercised to select efficient algorithms that do not have excessive space and time requirements.