

# CS300

## Course Syllabus | Fall 2017

### Course Title

CS300: Advanced Computer Graphics I

### Pre-requisite

CS250

### Course Description

This course introduces students to algorithms that are essential to creating photorealistic images in interactive simulations. Topics covered include an overview of modern GPU (graphics processor unit) architecture and the common graphics APIs used, including OpenGL and DirectX. Rendering techniques covered include texturing, illumination models, transparency, shading algorithms, mapping techniques (bump mapping, environment/reflection mapping, etc.), and shadows. Students will learn how to implement all algorithms by using vertex and pixel shaders.

### Objectives and Outcomes

After the completion of the course, the student will be able to implement the lighting, normal mapping, environment mapping (for reflection and refraction) and shadow mapping algorithm using shaders and will have a good understanding on various other computer graphics algorithms.

### Contact Information

Instructor: Pushpak Karnick  
Office Hours: Tue, Thu 2:00 pm – 3:00 pm, or by appointment.  
Email Address: [pushpak.karnick@digipen.edu](mailto:pushpak.karnick@digipen.edu) – Prefix the subject with [CS300]  
Class Web Page: [distance.digipen.edu](http://distance.digipen.edu)  
Contact: x-4413

### Section Day and Time Room

Section	Day and Time	Time	Room
CS300	Tuesday & Thursday	04:00 PM – 05:20 PM	VAN GOGH

## Textbooks & References

- OpenGL Programming Guide (8<sup>th</sup> Edition)  
[Dave Shreiner](#), Mason Woo, and Jackie Neider

Available through DigiPen Library at

<http://proquestcombo.safaribooksonline.com/9780132748445?uicode=digipen&cid=shareLink>

- OpenGL Shading Language (3<sup>rd</sup> Edition)  
[Randi J. Rost](#), Bill Licea-Kane, Dan Ginsburg, and John M. Kessenich

### Good-to-have references:

- Real-time Rendering (3<sup>rd</sup> Edition)  
Tomas Akenine-Möller, Eric Haines, and Naty Hoffman
- OpenGL Development Cookbook  
M. M. Mowbray

## Grading

Assignments	60%
Quizzes	20%
Midterm Exam	20%
<b>Total</b>	<b>100</b>

## Tentative Schedule (This is a guideline for the semester and is subject to change.)

Week	Topic
1	<ul style="list-style-type: none"> <li>OpenGL Introduction</li> <li>Shaders and Shading Languages</li> <li>Assignment 1: Draw objects with OpenGL – <b>Due Week 3</b></li> </ul>
2	<ul style="list-style-type: none"> <li>Shading Languages</li> <li>Introduction to Lighting Models</li> </ul>
3	<ul style="list-style-type: none"> <li>Lighting Techniques (Phong Illumination Model)</li> <li>Assignment 2: Lighting with Shaders – <b>Due Week 6</b></li> </ul>
4	<ul style="list-style-type: none"> <li>Texturing</li> </ul>
5	<ul style="list-style-type: none"> <li>Normal Mapping</li> </ul>
6	<ul style="list-style-type: none"> <li>Projective Texturing</li> <li>Assignment 3: Texturing &amp; Normal Mapping – <b>Due Week 10</b></li> </ul>
7	<ul style="list-style-type: none"> <li>Midterm</li> </ul>
8	<ul style="list-style-type: none"> <li>Shadow Volumes</li> </ul>
9	<ul style="list-style-type: none"> <li>Shadow Mapping</li> </ul>
10	<ul style="list-style-type: none"> <li>Environment Mapping</li> <li>Transparency and Refraction</li> </ul>

	<ul style="list-style-type: none"> <li>Assignment 4: Environment Mapping – <b>Due Week 13</b></li> </ul>
11	<ul style="list-style-type: none"> <li>Deferred Shading</li> </ul>
12	<ul style="list-style-type: none"> <li>Anti-aliasing</li> <li>Level-of-Detail methods</li> </ul>
13	<ul style="list-style-type: none"> <li>Advanced Topics: High Dynamic Range Imaging and Rendering</li> </ul>
14	<ul style="list-style-type: none"> <li>Advanced Topics: Global Illumination</li> <li>Advanced Topics: Ray Tracing</li> </ul>
15	<b>Finals Week</b>

### Late Policy

- Late Work:  
The following penalties will apply to work submitted late:
  - Less than one week: 5% deduction per day
  - More than one week: 0 on the assignment

### Classroom Policy

- Students are expected to behave professionally at all times with regards to classroom conduct and timely delivery of all assignments.
- Specific guidelines will accompany each assignment, along with a completion date.
- As this is a learning environment, it is critical that the noise level remains at a minimum. Cell phones must be turned off at the beginning of class; laptops are permitted if they are used for taking notes.
- All work in this class must be retained until the end of the semester. Subsequent assignments will build on the functionality implemented in previous assignments.**

### Academic Integrity Policy

Academic dishonesty in any form will not be tolerated in this course. Cheating, copying, plagiarizing, or any other form of academic dishonesty (including doing someone else's individual assignments) will result in, at the extreme minimum, a zero on the assignment in question, and could result in a failing grade in the course or even expulsion from DigiPen.

### Disabled Student Services

If students have disabilities and will need formal accommodations in order to fully participate or effectively demonstration learning in this class, they should contact the Disability Support Services Office at (425) 629-5015 or [dss@digipen.edu](mailto:dss@digipen.edu). The DSS Office welcomes the opportunity to meet with students to discuss how the accommodations will be implemented. Also, if you may need assistance in the event of an evacuation, please let the instructor know.

## Assignment Submission Guidelines

1. Your programming submission must be written using C++ under Visual Studio version installed on DigiPen computers. **Your assignments MUST compile and execute on Edison and Tesla machines. Failure to do so will automatically result in a zero grade for the assignment.**
2. The assignments are individual project. You may discuss and work with other student, but you MUST implement the assignment yourself, i.e. each project is a PERSONAL project. It is NOT a group project. An interactive forum will be provided for you to interact with other students and the TA (if any) for the course.
3. Your programming submission must contain a README text file. The README file MUST contains the following information:
  - a. How to use parts of your user interface that is NOT specified in the assignment description.
  - b. Any assumption that you make on how to use the application that, if violated, might cause the application to fail.
  - c. Which part of the assignment has been completed?
  - d. Which part of the assignment has NOT been completed (not done, not working, etc) and explanation on why those parts are not completed?
  - e. Where the relevant source codes (both C++ and shaders) for the assignment are located. Specify the file path (folder name), file name, function name (or line number).
  - f. Which machine (or lab) in DigiPen that you test your application on.
  - g. The number of hours you spent on the assignment, when you started working on it and when you completed it.
  - h. Any other useful information pertaining to the application
4. **Failing to include the required information in the README file will lose you 15 points on your grade for that assignment.**
5. The README, source, header, and data files must start with the header specified here:

```

/* Start Header -----
Copyright (C) 2017 DigiPen Institute of Technology.
Reproduction or disclosure of this file or its contents without the prior written
consent of DigiPen Institute of Technology is prohibited.
File Name: <put file name here>
Purpose: <in a concise fashion, explain the contents of this file>
Language: <specify language and compiler>
Platform: <specify compiler version, hardware requirements, operating systems>
Project: <specify student login, class, and assignment. For example: if foo is in
class CS300 and this file is a part of assignment 3, then write: foo_CS300_3>
Author: <provide your name, student login, and student id>
Creation date: <date on which you created this file>
End Header -----*/

```

6. To submit your programming assignment, organize a folder consisting of ALL relevant source code (including source files, header files, data files), documentation files, Visual C++ solution and project files, and any 3rd party library (GLUT/freeglut, GLEW, GLUI etc) that you are using (with the exception of OpenGL). In other words, your submission must be ready for compiling and linking by the grader. Name this folder using the following convention:
  - a. <student login name>\_<class>\_<assignment#>

7. For example, if your login is foo and assignment 2 is being submitted, your folder would be named foo\_CS300\_2. Failing to include and/or set all necessary files to compile 'cleanly' could cause you to lose points on your grade.
8. Your folders should contain all the files necessary for building a new solution.
9. Zip this folder and name the resulting file using the following convention:
  - a. <student login name>\_<class>\_<assignment#>.zip
10. For example, if your login is foo and you are submitting assignment 2, your zipped file would be named as: foo\_CS300\_2.zip
11. Next, upload your zip file after logging into the course web page using the link: <https://distance.digipen.edu>.
12. Finally, perform a sanity check to determine if your programming submission follows the guidelines by downloading the previously uploaded zip file, unzipping it, then compiling, linking, and executing your submission. If you have to resubmit because of problem(s) that could have been avoided if you did the sanity check, your resubmission date is your submission date.
- 13. Do the sanity check on DigiPen computer. If your application does NOT run on DigiPen computer, then it does NOT run. Regardless the fact that it could run on your desktop at home and/or laptop.**