CSIS 4466: Computer Graphics
Fall 2015 -- Syllabus

Instructor: Dr. Vincent Cicirello **Office:** G-116
E-mail: Please use the course Blackboard mail **Phone (office):** 609-626-3526

Office Hours: Tuesday/Thursday: 1:15-2:15pm (available other times by appointment).
Or, feel free to drop-in any time I'm in my office (if I'm there, I'd be happy to talk to you).

Course Time and Location: Tuesday/Thursday: 10:30-12:20pm (G108)

Course Description: This course provides an introduction to 2D and 3D computer graphics including the study of hardware, software, algorithms, modeling, rendering, and applications. Topics will be introduced using the Java 2D and 3D APIs.

Prerequisites: MATH 2216 (Calculus II) and CSIS 2102 (Programming & Problem Solving II).

Required Textbook: *Computer Graphics Using Java 2D and 3D*, by Zhang and Liang, (ISBN 0-13-035118-0).

Strongly Suggested: Any Calculus textbook as a reference (or a good online calculus reference).

Quantitative Reasoning (Q2): This course is a **Q2** (*Quantitative Reasoning Across the Disciplines*) course. All computer science courses on Computer Graphics are necessarily quantitative reasoning intensive. It is not possible to learn the fundamentals of graphics rendering engines and modern graphics APIs without having a firm grasp on mathematics, especially calculus, geometry, and several topics of linear algebra such as vector and affine spaces, linear transformations, matrix operations, etc. The required math topics that are not covered in pre-requisite courses will be covered within the course itself. In this course, you will be applying your prior mathematics skills from Calculus I and II; and additionally will be building upon that with new knowledge in a sampling of topics usually covered in Calculus III and Linear Algebra within the context of the application to 2D and 3D graphics modeling and rendering. *If you don't like math, this course is not for you.*

CSIS Student Learning Outcomes: This course supports the following CSIS student learning outcomes:

- Outcome CSIS.a: An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline.
  - CSIS.a.3: Students will apply discrete mathematics concepts and algorithms.
- Outcome CSIS.c: An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
  - CSIS.c.2: Students will implement a computer-based system, process, component, or program from a given specification.
- Outcome CSIS.i: An ability to use current techniques, skills, and tools necessary for computing practice.
  - CSIS.i.1: Students will use a professional integrated development environment (IDE) for implementing programming projects.
  - CSIS.i.2: Students will research online resources to learn and utilize new techniques, skills, and tools.
- Outcome CS.j: An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
  - CS.j.2: Students will evaluate the effects of alternative data representations and algorithms on the performance of computer based systems.
- Outcome CS.k: An ability to apply design and development principles in the construction of software systems of varying complexity.
  - CS.k.1: Students will construct a simple software system using basic design and development principles.
  - CS.k.2: Students will construct a more sophisticated software system using advanced design and development principles.

IDEA Learning Objectives: The course also meets the following IDEA learning objectives.

- IDEA objective 1: Gaining knowledge of the terminology, methods, and algorithms of 2D and 3D computer graphics;
- IDEA objective 2: Learning the fundamental mathematics and other underlying theory of computer graphics;
- IDEA objective 3: Learning to apply course material to 2D and 3D modeling and rendering;
- IDEA objective 4: Developing skills and competencies in the Java 2D and Java 3D APIs necessary to develop professional software for applications requiring 2D and 3D modeling and graphics rendering, as well as developing skills in using the Eclipse integrated development environment (IDE), an IDE that is widely used in industrial settings.

Grading:	Exam 1 / Exam 2	17.5% / 17.5%
	Programming assignments & other written problem sets	60%
	Participation	5%

Grading Scale:

A: Overall average at least 90.00 **AND** Exam average at least 80.00.	A-: Overall average at least 89.00 **AND** Exam average at least 80.00.	B+: Overall average at least 88.00 **AND** Exam average at least 80.00.
B: Overall average at least 80.00 **AND** Exam average at least 70.00.	B-: Overall average at least 79.00 **AND** Exam average at least 70.00.	C+: Overall average at least 78.00 **AND** Exam average at least 70.00.
C: Overall average at least 70.00 **AND** Exam average at least 60.00.	C-: Overall average at least 69.00 **AND** Exam average at least 60.00.	D+: Overall average at least 68.00 **AND** Exam average at least 60.00.
D: Overall average at least 60.00 **AND** Exam average at least 50.00.	D-: Overall average at least 59.00 **AND** Exam average at least 50.00.	F: Overall average less than 59.00 **OR** Exam average less than 50.00.

Academic Honesty: Please familiarize yourself with Stockton's policy on academic honesty. Any first violation will be penalized by a 0 on the relevant assignment/exam/etc, plus a 10 point penalty on your overall course grade. Subsequent violations will result in a course grade of F. Examples of violations include, but are not limited to: (a) any form of cheating on an exam or assignment, (b) passing off the work of another as your own, (c) assisting someone in violating the academic honesty policy, (d) asking someone to assist you in cheating or other academic honesty violations (even if they refuse to help you cheat), etc.

Incomplete Policy: In general, no grades of incomplete will be given. The only exception to this rule is an institutionally documented medical emergency that necessitates your complete absence from Stockton for at least two continuous semester weeks. Additionally, you must be caught up on all work up to the point where your medical emergency began and currently in the "C" range or better overall at the point where the emergency began.

Exam 1, Exam 2: The exams are not cumulative. You are allowed up to 2 sheets of notes on standard letter-sized paper (8.5" by 11"), which can be double-sided. You are also allowed, and strongly advised, to use a calculator during the exams. However, "other aids" that are not allowed include cell phones, pagers, PDAs, and other communications devices, not even for calculator purposes (i.e., your calculator must be an actual calculator). The computers in the classroom are also not allowed during exams (the monitors will be lowered into the desks via the "elevators").

Make-Up Exams: Make-up exams in general will not be given (i.e., missed exam = 0), with the following exceptions:

1. Documented medical excuse: please provide a doctor's note on the first class you return after the missed exam
2. Other institutional excuses: There may arise situations related specifically to Stockton that prevent you from being able to attend an exam (e.g., an away game for a Stockton team that you are on, a fieldtrip for a Stockton course, etc). In most such cases, you should be aware of the conflict beforehand. Thus, I must be notified of this conflict one week prior to the missed exam, with written documentation (e.g., letter from Stockton coach or Stockton sponsor of field trip, etc).

Programming Assignments & Problem Sets: The most significant portion of your grade is based on programming assignments and other homework such as problem sets.

- **Problem Sets:** Several homework assignments will be problem sets, including but not limited to the mathematical foundations of the computer graphics topics we cover in class. Such problem sets are to be done individually.
- **Programming Assignments:** You have a choice of either working independently or with one other student on the programming assignments. If you work as a team of two for an assignment, then both receive the same grade on that assignment. You can choose to work as a pair for some assignments and individually for others (your choice).

Due Dates & Late Policy: Programming assignments will generally be due electronically via Blackboard and will be due by 11:59pm on the dates due. Problem sets will be due by the start of class (10:30am) and can be submitted either on paper or electronically in Blackboard. Late programming assignments and late problem sets will be penalized as follows: 25% off if less than 24 hours late, 50% off if less than 48 hours late, 75% off if less than 72 hours late, and a grade of 0 for assignments submitted at least 72 hours late. *The first time an assignment is late (within 72 hours), the late penalty will be waived.*

Participation: The participation part of your overall grade is not free points. You are expected to fully participate in all course activities. You will not be directly penalized for an occasional absence (e.g., a small number of absences will not alone reduce your participation grade). **Things that can positively impact your participation grade include:** asking/answering questions during class and other general class participation, assisting others stuck on any part of lab activities, or any other similar things that positively benefit the class. **Things that can negatively impact your participation grade:** frequently skipping class, frequently arriving late or leaving early, sleeping during class, any disruptive behavior (e.g., ringing phones, answering or making phone calls during class, arriving late or leaving early in a noisy or otherwise disruptive way, etc).

Tentative Schedule: This schedule is subject to change, and is likely to change. Changes will be announced via Blackboard (and in class). If tentative exam dates change, they will be announced at least one week prior.

Date	Topic
September 3	Syllabus; Course Overview; Intro to Computer Graphics
8	Overview of Computer Graphics; Brief review of some key elements of Java plus some examples using Java 2D
10	Basic 2D Graphics: the rendering process, 2D geometry and coordinate systems
15	Basic 2D Graphics: The Graphics2D class; graphing equations
17	Basic 2D Graphics: Geometric models; constructive area geometry, General Path
22	2D Rendering: Colors, Paints, Strokes
24	Math Background: Homogeneous coordinate systems
29	2D Rendering: Affine Transformations, Composition of transformations
October 1	2D Rendering: Transparency and compositing rules
6	2D Rendering: Clipping; Text and Font
8	Advanced 2D: B-spline curves
13	Advanced 2D: Fractals, Image Processing
15	Advanced 2D: Custom Primitives, Animation
20	Slack and/or Review for Exam 1
22	**Exam 1**
27	**NO CLASSES: Preceptorial Advising Day**
29	Math Background for 3D graphics: Graph Theory
November 3	Basic 3D Graphics: 3D rendering process and general overview of 3D graphics, Overview of Java 3D API
5	Basic 3D Graphics: Scene Graphs
10	Basic 3D Graphics: Structure of a Java 3D program
12	Basic 3D Graphics: Backgrounds and bounds
17	Graphics Contents
19	Graphics Contents
24	Geometric Transformation: 3D Affine Transformations
26	**NO CLASSES: Thanksgiving**
December 1	Geometric Transformation: Transformations in Scene Graphs
3	Independent Assignment TBD
8	Ideally, we will hopefully get to Views, Lighting, Texturing
10	Slack and/or Review for Exam 2
15	**10:30am classes do not meet**
17	**Exam 2: 10:30-12:30**