Overview:
Computer graphics is one of the most exciting ways that computing has made an impact on the world. From the simple charts to see data, to the ways graphics has enhanced entertainment, to the ways graphics and visualization helps understand scientific principles, computer graphics is present everywhere. Computer graphics is a very broad field involved in work that uses computation to create or modify images that are still, moving, interactive or fixed and on film, video, screen, or in print.

Graphics programming tasks typically include developing appropriate representations for the geometric objects that are to make up the images, to assemble these objects into an appropriate geometric space where they can have the proper relationships with each other as needed, to define and present the look of each of the objects as part of that scene, to specify how the scene is to be viewed, and to specify how the scene as viewed is to be displayed on the graphic device.

A greatly improved graphics hardware and software is now a part of most computing systems. Although, computer graphics is a highly technical field, requiring programmers to understand some mathematics and algorithms along with many hardware nuances, it has now become a field that allows the graphics programmer to think and work at a much higher level of modeling and to create effective images that communicate effectively with the user by employing software tools such as a graphics API.

An Application Programming Interface (API) is a set of tools that allow a programmer to write applications that include the use of interactive computer graphics without dealing with specific system details. Among the advantages of an API is that it hides the details of any one computer system and allows the programmer to develop applications that will work on any of a wide range of systems.

The course is intended to introduce the fundamentals of computer graphics programming to the students. The programming environment for the course is Python, a high level programming language and the OpenGL API. It is expected that using Python and OpenGL would make it easier to understand concepts and application so as to be able to cover a breadth of graphics topics in the duration of the course.
Course Sections (details TBA shortly):

1. Introduction to Computer graphics and Python
2. Introduction to OpenGL (PyOpenGL)
3. Principles of modeling, viewing and projections and transformations
4. Color, blending, lighting models and shading
5. Applications, basic image processing, animation, games

Course Policy:

1. The course consists of 2 lectures (1.5 hrs each) per week and a lab and tutorials session (3.0 hrs each) per week.
2. Each lab session is a tutorial and a class assignment to be completed individually.
3. There are about 10 homework assignments (minor) during the course to be completed individually.
4. The major course project is introduced midway during the course.
5. All exams are given and are mandatory.
6. Final grades are based on performance in class, lab assignments, homework, the project and the exams.

Readings:

3. Dive into Python by Mark Pilgrim
4. How to Think Like a Computer Scientist: Learning Python by Allen B. Downey, Jeffrey Elkner and Chris Meyers
5. Python Tutorial by Guido van Rossum (inventor of Python)
6. A Quick, Painless Tutorial on the Python Language by Norm Matloff