Instructor: Dr. K.R.Subramanian, Rm. 435E, Woodward Hall.
Phone: (704) 687-8579
Email:krs@uncc.edu
WWW: http://www.cs.uncc.edu/~krs

Course Website: http://www.cs.uncc.edu/~krs/courses/4120-5120

Office Hours: Mon. 5-6pm, Thurs. 3-4 pm, or by appointment.

Meeting Times: Mon. 12.30-1.45pm, Woodward 155

Prerequisites: Data Structures (ITCS 2214 or equivalent), linear/vector algebra (MATH 2164),
good programming skills(C++ or Java).

Lab Facilities: Linux PCs, Woodward Hall 335 Student Lab

Text: Donald Hearn and Pauline Baker, Computer Graphics - with OpenGL,

Reference: Dave Shreiner, Mason Woo, Jackie Neider, Tom Davis, OpenGL Programming

Documentation: OpenGL:
OpenGL Resources: http://www.opengl.org

FLTK: http://www.fltk.org

Grading: Programming projects: 50%, Midterms(2) 15% each, Final: 20%.
No other graded homework. All projects evaluated by interactive demonstrations
on campus.

Grading roughly on 90-80-70-60 scale, with letter grades separated by
suitable breakpoints.
Passing grade on projects required to pass the course.
Incompletes, only under exceptional circumstances and at the discretion of instructor.

Course Materials
Lecture slides, projects and other materials will be available at the website below.
www.cs.uncc.edu/~krs/courses/4120-5120

Attendance Policy
Attendance of all scheduled classes is required, as the material covered in the lectures will not necessarily be
restricted to that in the prescribed text. You are responsible for all material covered in class. Attendance for all
exams is mandatory. Makeup Exams will not be given except under special circumstances.

Academic Integrity
Cheating in any form is subject to disciplinary action(UNCC Code, under http://www.legal.uncc.edu/policies/ps-105.html). As far as programming projects are concerned, you are allowed to discuss general concepts and
strategies for solving problems. No sharing of modules or parts of programs will be allowed.
COURSE SCHEDULE

<table>
<thead>
<tr>
<th>Topic</th>
<th>Chapters</th>
<th>Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphics Applications, Hardware, Math Background</td>
<td>1, 2, A-1</td>
<td>8/24, 8/26</td>
</tr>
<tr>
<td>Graphics Packages, OpenGL</td>
<td>2, 3</td>
<td>8/26, 8/31, 9/9, 9/14</td>
</tr>
<tr>
<td>Raster Graphics</td>
<td>3, 4</td>
<td>9/16, 9/21, 9/23, 9/27</td>
</tr>
<tr>
<td>Mid Term 1 - 10/5 (Roughly)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Geometric Transformations</td>
<td>5</td>
<td>10/7, 10/14, 10/19</td>
</tr>
<tr>
<td>3D Viewing and Projections</td>
<td>7</td>
<td>10/21, 10/28, 11/2</td>
</tr>
<tr>
<td>Mid Term 2 - 11/4 (Roughly)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shading Models, Rendering</td>
<td>10</td>
<td>11/9, 11/16</td>
</tr>
<tr>
<td>Visible Surface Detection</td>
<td>9</td>
<td>11/18, 11/30</td>
</tr>
<tr>
<td>Texture Mapping, Modeling</td>
<td>8, 10</td>
<td>12/2, 12/7</td>
</tr>
<tr>
<td>Final Exam, May 4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

COURSE OUTLINE

- Introduction, Applications, Math Background
  - What is Computer Graphics?
  - Applications of computer graphics.
  - Graphics pipeline, display hardware
  - Math background: linear algebra - coordinate systems, points, vectors, matrices

- Graphics Packages and OpenGL

- Raster Graphics (2 weeks)
  - Graphics primitives, attributes
  - Raster algorithms: Filling, Scan-conversion.
  - Aliasing and Anti-aliasing.
  - Clipping algorithms: Line and Polygon Clipping.

- Geometric Transformations
  - 2D/3D Affine Transformations
  - Matrix representation and homogeneous coordinates
  - Window - Viewport transformation

- Modeling (1 week)
  - Hierarchical Modeling.
  - Surface and Scene Representations (CSG, Scene Graphs)

- 3D Viewing and Projections
  - Parallel and perspective projections
  - The camera transformation
  - Mathematics and implementation of geometric projections.
  - 3D Clipping

- Shading Models and Rendering Algorithms
○ Local Reflectance Models, Flat, Gouraud and Phong shading.
○ Global Illumination Algorithms - Ray Tracing, Radiosity
○ The Rendering Equation

■ Visible Surface Detection
○ Image vs. Object precision algorithms
○ Z-buffer, painter’s, Warnock’s algorithms.
○ Scanline algorithm, BSP trees, Ray Tracing.

■ Texture Mapping, Cubic Curves/Surfaces
○ Surface Detail
○ Mipmaps
○ Bump mapping, displacement maps, cube maps
○ Cubic curves (Bezier, B-Spline)