# CS 362: Computer Graphics

In the assignments, you will design a small graphics application by implementing a set of procedures and algorithms that correspond to the different stages of the pipeline. The application allows the user to *define* object(s), which are then displayed in a *specified* way. **You should create a library of the procedures you are implementing, which can be called to implement graphics applications.**

The implementation will be done in stages, in sync with the progress in the course. Along with the implementation of the basic algorithms, you will also be required to learn the corresponding OpenGL functions and use those as specified in the assignments.

**Instructions:**

**a) Form two-member groups (inform me in case of any difficulties/confusion).**
**b) Any copying is strictly prohibited.**

## Part 1

1. **TAs will check your assignment (AL3 lab slot)**
2. **Date of submission: 2/2/11**

At the very beginning of the graphics pipeline, we need to represent/model 3D objects. We have discussed about different object representation techniques in the class. In part 1 of the assignment, you are expected to do the following.

1. Learn about different object representation techniques supported by OpenGL.
2. Implement the procedures to represent *curved surfaces* using (a) Hermite cubic and (b) Bezier cubic splines. A curved surface can be thought of as defined by a *mesh* (i.e. grid) of control points that represent set of orthogonal splines. You should use the De Casteljau algorithm to determine the points on the surfaces. That means, you have to transform the Hermite cubic representation to Bezier for the first case, as discussed in the class. Figure out ways to avoid conflicts, if any.
3. There should be an interface to specify the control points.

**You should show the wireframe model of at least two objects, demonstrating the working of your implementation. Also, show the same objects (with same mesh of control points) using the OpenGL functions for comparison. I leave it to your imagination to decide which objects to show and how to design the interface for providing inputs. More complex the surfaces are and user-friendly the interface is, more marks you will fetch.**

## Part 2

3. **TAs will check your assignment (AL3 lab slot)**
4. **Submission due by 19/2/11**

The 3D objects we modeled in the previous assignment needs to be combined together to create a scene. This requires modeling transformations that we discussed in the class. In part 2, you are expected to do the following.

1. Learn about different transformations supported by OpenGL.
2. Implement the procedures to perform the following 3D transformations w.r.t. any arbitrary point/axis.
    a. Scale
    b. Translate
    c. Shear
    d. Rotate

**You should create a scene (of your choice) where all the four transformations are used, to demonstrate the working of your implementation. Note that the objects you transform should be modeled using the procedure you developed in part 1. Also, construct the same using the OpenGL functions for comparison.**

## Part 3

1. **TAs will check your assignment (AL3 lab slot)**
2. **Submission due by 16/3/11**

Once a scene is created, it is illuminated to get realistic effects. Afterwards, we will project it on the view plane. In part 3, you are expected to do the following.

1. Learn about the following functions supported by OpenGL.
    a. The different shading models (does OpenGL supports all the shading models we discussed in the class?)
    b. Texture mapping functions.
    c. Projection functions.
2. Design and implement the procedures to perform the following tasks.
    a. Perspective projection.

**Create a scene with simple-shaped (say, cubical) objects (should use the transformation function from previous assignment to construct the scene). Apply the shading and projection functions to illuminate and project the scene. Also do the same using OpenGL functions for comparison (you may ignore normalization transformations for perspective projection).**

## Part 4

1. **TAs will check your assignment (AL3 lab slot)**
2. **Submission due by 13/4/11**

In this part, you will write library functions for performing the remaining stages of the graphics pipeline, namely clipping, HSR and scan conversion. Do the following.

1. Learn the different OpenGL functions to perform the tasks. Try to find out which of the various algorithms we discussed in the class are implemented by OpenGL.
2. Implement the procedures to perform the following tasks.
    a. 3D polygon clipping against an input view volume (all the planes are provided as input). Implement Sutherland-Hodgeman algorithm.
    b. Scan conversion using Breshenham's line drawing.
    c. Back-face culling and scan-line method for hidden surface removal (Note that the Bresenham's method in conjunction of the scan-line method of HSR can be used to scan convert the surfaces**)**
    d. Gouraud shading.

**You should create a scene with at least two objects (of your choice). Each object should consist of multiple surfaces. Note the following.**

**a) For simplicity, you may assume that the objects are made up of polygonal surfaces.**
**b) The surfaces can be convex or concave. In case of concave surfaces, you should split it up into convex polygons using the vector method we discussed.**
**c) All polygons should be converted to triangle meshes before applying Sutherland-Hodgeman.**
**d) Apply Gouraud shading after scan conversion.**
**e) Assume perspective projection for scan conversion.**

**You MUST use the methods you defined earlier for transformation, illumination and projection. Also, construct the same using the OpenGL functions for comparison.**