Assignment Problem:

Problem: Use a selection of OpenGL functions to recreate several provided 3d scenes which match the images in the HW2 folder screenshot 1-3. The second part of the assignment is to create a nontrivial image with your own imagination using mainly teapot and cube functions.

Methods:

For the first problem, there are 10 units revolved around a circle so theta was calculated by dividing 360 by the number of units then using a for loop with 10 iterations to use **glPushMatrix()**, which creates a plane for the object to be drawn on. Next, we rotate the plane around the origin (0, 0, 0) using the function **glRotatef(theta, 0, 0, 1)** to rotate around the z axis to give a 2d rotate image instead of a 3d one. Then, we translate the plane in order to make a circle by using the command **glTranslatef(1, 0, 0)** to translate the matrix across the x axis by 1 unit. Finally, you draw the teapot using **glutSolidTeapot(0.25)** with a size of 0.25 to draw a solid teapot which will be taken care of in the loop until it makes a circle with 10 teapots.

The next problem is making a staircase which is a bunch of cubes bunched together over 4 units. The function made in the cpp creates 16 steps over 4 units that increments nonlinearly. 3 variables are initialized to set the initial points for the shape like size, initial scale (the height of the steps), and the initial point named move. Since there are 16 steps, a for loop is initiated with 16 iterations and the same process with matrices being pushed each loop. The scale is increased with the formula **scale = scale + 3*(iteration number/128)** which was calculated using a quadratic formula to reach a specific point. Next, the scale is applied with **glScalef(1, scale, 3)** which increases the height of the steps by a certain amount, and the z axis (wideness) by 3. Then the plane is translated by the size of the cube lengthwise on the x axis starting at **x = -2** in order for the steps to be touching each other each iteration. The cube is then drawn with **glutSolidCube(size)** and results in a rectangle, and the matrix is popped at each iteration which results in a staircase.

For problem 3, in order to make a pyramid with even number rows, a bool matrix was used with a defined area of 11 row x 6 columns. The matrix would be filled out with a for loop that counts the set amount of spaces calculated by **spaces = num_rows - row - 1** with -1 for the array to be centered since the starting index is 0. Once it reaches the right amount of spaces, then it adds a true to the next column followed by a false in a pattern for a set amount of times (row number). Then another for loop is made for iterating through the matrix and drawing the planes with defined bounds in order for the matrix to be properly used. The matrix is translated with **glTranslatef(left_bound[-2.5] + col_number/2.0, top_bound[1] - row/2.0, 0)** by translating to the right by ½ for each column and down by ½ for each row. If the spot in the matrix is true, then the teapot is drawn, and the process repeats until there is a pyramid with gaps in between each teapot.

The creative one models a hand that has 2 extremely long fingers which uses a for loop to create each finger. The base of the hand is made first with a solid cube scaled by 2 on the z axis, then the thumb is made with 2 different rectangles with one being scaled 2x on the x axis and

translated by 0.85 in order to line up with the other part of the thumb which is scaled 2x toward the z axis and translated to x = 0.25 and z = 0.25 and rotated by -15 degrees around the y-axis in order to be diagonal. The fingers are made in a for loop that generate 3 solid cubes that are scaled specifically and translated to a specific spot if they are the first or second finger (toward the left). If they aren't, then 2 very long solid cubes are made by scaling the z axis by 12. A triangle is drawn and colored with immediate mode and colored in order to have it drawn quickly which is at the top of the function. The result is the 4th screenshot in the directory.