This assignment is designed to introduce you to OpenGL by experimenting with 3D geometric primitives and hierarchical affine transformations. An executable version of your “Jumping Brain” assignment (discussed below), as well as the source code for a related “Running Dude” program (pictured above), are linked to the course website (http://www.cs.siue.edu/~wwhite/CS482/Syllabus.htm).

Note that the “Running Dude” program displays a primitive model of a person, composed of various spheres and cylinders that have been arranged in a hierarchical manner to simulate a running action. Keyboard commands allow the user to speed up or slow down the runner’s pace, and to reorient the viewing angle.

This code employs two classes to implement the model’s hierarchy. The Joint class, which consists of a sphere and a cylinder emerging from one side of the sphere, is limited to rotation about the x-axis (i.e., pitch). This class is used to implement the head/torso, humerus, forearm, hand, thigh, shin, heel, and toe. The Crossjoint class, consisting of a sphere and a cylinder emerging from both sides of the sphere, permits rotation about the z-axis (roll) and about the y-axis (yaw). This class is used to implement the pelvis and the shoulders, which are the top of their respective hierarchies for the runner’s lower and upper body.

The individual limbs are constructed one piece at a time, using OpenGL’s stack approach. For example, the left toe section is positioned, rotated by an appropriate amount, and then translated so it will be connected properly to the heel. (Due to the symmetry of the sphere, the rotation occurs first; if the sphere was scaled to be less symmetric, then the sequence of transformations might have to be modified.) Next, the left heel section is positioned, rotated and then translated so it will be connected to the shin; then the same is done for the shin and the thigh.

Your assignment is to modify the “Running Dude” program to produce an animated version of the Emilio Garcia “Jumping Brain” figure pictured at right. In this first version of the figure, the emphasis will be upon using the Joint and Crossjoint classes to produce a primitive model composed of spheres and cylinders, with a keyboard-based ability to produce an animated “jumping” motion. Later assignments will involve refining the brain model and placing the figure in an environmental setting.

In your assignment, you will modify the “Running Dude” program in several significant ways:

1. The current program uses unscaled spheres to model the joints and crossjoints, but the Jumping Brain model requires that the spheres have the ability to be scaled in each of their three dimensions. This will require new data members, accessors, and mutators, as well as alteration to the constructor, initialization, and draw member functions in both the Joint and Crossjoint classes.

2. All references to the Running Dude class need to be replaced with updated references to the new Jumping Brain class that you’ll be developing. In addition, new global constants will need to be developed to implement the scale factors for the various body parts. Of course, you should accurately comment your Jumping Brain code, excising outdated references to the Running Dude code.
3. Due to the reduced symmetry of the model components, significant changes will be needed in the Jumping Brain’s draw function. Specifically, when rendering the model’s left leg, the top of the stack should contain the toe section, centered at the origin. That section should then be translated to allow an appropriate toe rotation around one side of the toe. That rotated toe should then be translated to a position at the end of the heel section of the leg, which is positioned at the origin. The heel/toe combination is then rotated according to the heel’s current orientation, and then translated to the end of the shin section, which is positioned at the origin. This process is continued to orient and position the shin/heel/toe section at the end of the thigh, and then the thigh/shin/heel/toe section at one end of the pelvis. Similarly, a hierarchical approach will be used to attach the humerus/forearm/hand sections to the shoulders.

4. Add keyboard controls to increase and decrease the speed of the arms (A/a), the speed of the legs (L/l), as well as the speed of all limbs (+/-). Graduate students in this course must also provide keyboard controls for toggling the visibility of each type of body part. (B=Body; S=Shoulders; H=Humerus; F=Forearm; h=hand; P=Pelvis; T=Thigh; s=shin; f=foot; t=toe.) The title bar of the display window should be altered to reflect all available keyboard controls.

Keep your code modular and readable, with an extensive explanation (including your name) at the top of each program file, explanatory sentences preceding each function, and in-line comments everywhere within the code where your logic is particularly complicated. Modify all comments in the “Running Dude” code to reflect the changes that you make in your “Jumping Brain” version. Avoid code redundancy by foregoing cut-and-paste in favor of placing any code that is needed repeatedly into its own module (function, class, structure, etc.).

Place all of your program files (just the .cpp and .h files, not your project file) into a single folder named with your last name. When the instructor creates a new Visual C++ project and properly attaches the OpenGL libraries and your code, it must compile and execute in order to be graded. Zip-compress this folder and copy it to your Moodle drop-box by Thursday, September 7, 2017, at 12:00 Noon. Late assignments are not accepted without verifiable medical documentation. You must write your own code (with the exception of clearly annotated code that you receive from the instructor or as part of the material available from the textbook publishers), and no one but the instructor may see your code.