Introduction

Each part of this lab builds on the one before it. Write your code on a computer. At each step, make sure that the code builds without errors, and be sure to get you professor to check each step.

Note: This lab does not explicitly tell you what to write in `main`. Get used to this! You should use `main` as a sandbox space to try out the classes and functions that you are writing outside of `main`. Test them to see if they work like you expect! When you move on the next part, then feel free to clear out `main` again.

This lab is **due as a miniassignment** on canvas. Submit your final code by 11:59pm on Tuesday November 22nd.

---

1. **Point3D class.** In class on Monday we learned how to write the simplest type of class: one that only contains public data members. Our in-class example was 2D points.

   $\implies$ Write the class `Point3D`. This will be almost the same as 2D points, but each point with need x, y, and z coordinates.

2. **Midpoint function.** The other part of Monday's demo was a midpoint function, which took two points and returned a point halfway between them. Now do this with 3D points. Your prototype should look like this:

   ```
   Point3D midpoint(Point3D p1, Point3D p2);
   ```

3. **Distance function.** Write a function to compute the distance between two 3D points. Use this prototype:

   ```
   double distance(Point3D, Point3D);
   ```

   and use this formula:
   $$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

4. **Sphere class.** Write a new class called `Sphere`. A sphere has a center (of type `Point3D`) and a radius (of type `double`).

5. **Volume function.** Write a function that computes the volume of a sphere. Use this prototype:

   ```
   double volume(Sphere s);
   ```

   and this formula:
   $$V = \frac{4}{3}\pi r^3$$

6. **Intersecting function.** Write a function that decides if two spheres intersect each other at all. If they do, return `true`, and if not then return `false`. Use this prototype:

   ```
   bool areIntersecting(Sphere s1, Sphere s2);
   ```

   And this procedure: if the distance between the sphere centers is greater than the sum of their radii, then they do not intersect. Otherwise, they do.