CS350
Assignment 1: Geometry Library

Submission:

Zip up and submit the entire project folder (the .sln and the CS350Framework folder). Make sure to not include any build artifacts! See the syllabus for submission specifications. Due January 28th (Sunday) by 11:55pm.

Topics:

The purpose of this assignment is to build up a geometry library needed for the rest of this class. This mostly consists of Point, Ray, Frustum, Plane, Triangle, Sphere, and Aabb tests.

Testing:

You will be given a txt file containing sample results for the assignment. To test your project run with the command line arguments of *"%AssignmentNumber%* 0" where *%AssignmentNumber%* is the current assignment's number (1 in this case) and diff them against the provided results file. The argument "0" runs all of the tests.
Do note that the graded tests will not be limited to the ones given to you!

Implementation Details:

For the most part, make sure to read the comments for each function (in the header). Listed here are some extra details to help avoid confusion.

BarycentricCoordinates:

For later tests it is necessary to expand the test shape just a bit due to numerical robustness. An epsilon is passed to this function for that reason. The epsilon should be used to expand the barycentric coordinate range test. For example, a line normally would validate that each coordinate is between [0, 1]. With the provided epsilon the range to check should instead be [-epsilon, 1+epsilon].

RayAabb:

Although you can conceptually think of this test as performing 6 RayPlane tests, do not call RayPlane in this test! This is quite inefficient and you should write each axis' version. Also make sure to handle when the ray is parallel to an axis correctly (no zero divisions)!

PlaneAabb:

You will lose points for testing all 8 points. You must either use the 2 point method or the radius method.

FrustumTests:

For FrustumSphere and FrustumAabb the variable "lastAxis" is passed in but is not required for this assignment. Later assignments will require this variable to be used. The lastAxis should be the first frustum plane you test. Also, lastAxis should be filled out with the first plane that results in non-intersection. Note that this should still result in 6 plane tests at maximum!

Code Quality:

Code quality is a percentage modifier applied to your grade up to a -20%. Common code quality penalties are: redundant code, re-computed values, unnecessary allocations, and very hard to read code. In particular, no function should allocate memory in this assignment! Any buffers/arrays needed are fixed sized so no allocations are necessary!

Grade Breakdown:

	Points	Percentage
Geometry.cpp:		
BarycentricCoordinates (line)	4	4%
BarycentricCoordinates (triangle)	7	6%
PointPlane	4	4%
PointSphere	4	4%
PointAabb	4	4%
RayPlane	7	6%
RayTriangle	7	6%
RaySphere	7	6%
RayAabb	7	6%
PlaneTriangle	7	6%
PlaneSphere	7	6%
PlaneAabb	7	6%
FrustumTriangle	7	6%
FrustumSphere	7	6%
FrustumAabb	7	6%
SphereSphere	7	6%
AabbAabb	7	6%
Shapes.cpp:		
Both Plane::Set functions	5	4%
Total	112	100%

Notes:

Avoid creating new files in the framework. I will be copying the relevant files for the assignment into a clean project for testing. Do feel free to add any helper functions/classes (only in the submitted files!). For this assignment that is the files:

1. Geometry.hpp/cpp
2. Shapes.hpp/cpp

The full framework should still be submitted though.

To make it easier to find missing functions they contain the comment "/******Student:Assignment1******/" that you can search. Also there is a warning that will pop-up in the console.