### **Point3D Class Unit Tests Documentation**

#### 1. **DefaultConstructor**

* Description: Verifies the correct initialization of a Point3D object using the default constructor.
* Test Method: TEST(Point3DTest, DefaultConstructor)

#### 2. **ParameterizedConstructor**

* Description: Ensures that the Point3D object is correctly initialized with specified values.
* Test Method: TEST(Point3DTest, ParameterizedConstructor)

#### 3. **SettersAndGetters**

* Description: Validates the functionality of the setter and getter methods for the Point3D class.
* Test Method: TEST(Point3DTest, SettersAndGetters)

#### 4. **OperatorOverloadingAddition**

* Description: Checks the correctness of the addition operator overloading for two Point3D objects.
* Test Method: TEST(Point3DTest, OperatorOverloadingAddition)

#### 5. **OperatorOverloadingSubtraction**

* Description: Tests the subtraction operator overloading for Point3D objects.
* Test Method: TEST(Point3DTest, OperatorOverloadingSubtraction)

#### 6. **OperatorOverloadingMultiplication**

* Description: Validates the multiplication operator overloading for a Point3D object and a scalar.
* Test Method: TEST(Point3DTest, OperatorOverloadingMultiplication)

#### 7. **OperatorOverloadingDivision**

* Description: Tests the division operator overloading for a Point3D object and a scalar.
* Test Method: TEST(Point3DTest, OperatorOverloadingDivision)

#### 8. **OperatorOverloadingCrossProduct**

* Description: Checks the correctness of the cross product operator overloading for two Point3D objects.
* Test Method: TEST(Point3DTest, OperatorOverloadingCrossProduct)

#### 9. **OperatorOverloadingDotProduct**

* Description: Validates the dot product operator overloading for two Point3D objects.
* Test Method: TEST(Point3DTest, OperatorOverloadingDotProduct)

#### 10. **OperatorOverloadingCompoundAssignment**

* Description: Tests the compound assignment operators (+=, -=, \*=, /=) for a Point3D object.
* Test Method: TEST(Point3DTest, OperatorOverloadingCompoundAssignment)

#### 11. **NormalMethod**

* Description: Ensures the correct computation of the magnitude (normal) of a Point3D object.
* Test Method: TEST(Point3DTest, NormalMethod)

**Triangle Class Unit Tests Documentation**

**DefaultConstructor**

Objective: Verify the correct initialization of a Triangle object using the default constructor.

Test Method: TEST(TriangleTest, DefaultConstructor)

Test Steps:

* Create a Triangle object using the default constructor.
* Ensure that the vertices (p1, p2, p3) are initialized to (0, 0, 0).
* Ensure that the normal vector is initialized to (0, 0, 0).

**ParameterizedConstructorPoints**

Objective: Ensure that the parameterized constructor with vertices initializes a Triangle object correctly with the provided vertices and a zero normal vector.

Test Method: TEST(TriangleTest, ParameterizedConstructorPoints)

Test Steps:

* Create a Triangle object using the parameterized constructor with three vertices (p1, p2, p3).
* Verify that the vertices are set to the provided values.
* Verify that the normal vector is set to (0, 0, 0).

**ParameterizedConstructorNormalPoints**

Objective: Ensure that the parameterized constructor with normal vector and vertices initializes a Triangle object correctly with the provided normal vector and vertices.

Test Method: TEST(TriangleTest, ParameterizedConstructorNormalPoints)

Test Steps:

* Create a Triangle object using the parameterized constructor with a normal vector and three vertices (normal, p1, p2, p3).
* Verify that the vertices are set to the provided values.
* Verify that the normal vector is set to the provided normal vector.

**SettersAndGetters**

Objective: Ensure that the setters and getters for the normal vector in the Triangle class work as expected.

Test Method: TEST(TriangleTest, SettersAndGetters)

Test Steps:

* Create a Triangle object using the default constructor.
* Set a new normal vector using the setNormal method.
* Verify that the normal vector is updated correctly.
* Set the normal vector to (0, 0, 0) using the setNormal method.
* Verify that the normal vector is set to (0, 0, 0).
* Set the normal vector to a new value using the setNormal method.
* Verify that the normal vector is updated correctly.