

Project 1 - Vector 3D

Due Jan 25, 2024 by 11:59pm

Points 100

Submitting an external tool

Project 1: Writing and testing a class named Vector3D

1. Instructions

This project should be done individually.

Since this assignment is to get you familiar with Java, most of it will need to be done by hand so you understand the syntax. Out of problems 4, 5, 6, 7, 8, 9, and 10 in Section 2.2 (What to do), you may choose up to two problems to be solved by a language model. At the top of your file, you must explain how you used the language model. For example, you can say "I asked ChatGPT to write a Java method that returns the product of a vector and a constant."

Pay attention to naming! Our test files expect your submissions to be named a certain way. Therefore, whenever the assignment specifies:

- the names of classes,
- the names and types of the fields within classes,
- the names, types and order of the arguments to the constructor, or
- filenames (directories, packages)

...be sure that your submission uses exactly those names.

2. Vectors

2.1. Introduction

Vectors are commonly used in Math and physics. A vector (in 3D) is signified by a direction and a length (magnitude). It is commonly represented as three components: x, y and z. Some operations on vectors include:

- Magnitude: for vector v this is denoted as $|v| = \sqrt{x^2 + y^2 + z^2}$.
- Normalizing a vector: this is a vector that is obtained by dividing each component of a vector by its magnitude.
- Addition of two vectors: a vector sum is a vector that is obtained by adding respective components of the two vectors.
- Multiplying a vector by a constant: this produces a vector obtained by multiplying each component of the current vector by the provided constant.
- Dot product: The dot product of two vectors $v = (v_x, v_y, v_z)$ and $w = (w_x, w_y, w_z)$ is defined as a number $v \cdot w = v_x * w_x + v_y * w_y + v_z * w_z$
- Angle between two vectors: This is defined as $\cos(\theta) = \frac{v \cdot w}{|v||w|}$. Technically, given two vectors, there are two angles between them. This returns the smaller of the two angles. For example, if the two vectors align with clock hands at 3PM, this would return 90 degrees.
- Cross product: Without going into too many details, the cross product of two vectors $v = (v_x, v_y, v_z)$ and $w = (w_x, w_y, w_z)$ is defined as another vector $u = (u_x, u_y, u_z)$ that is at right angles to both. See [here](#) for more details. This resulting vector u can be calculated this way:
 - $u_x = v_y w_z - v_z w_y$
 - $u_y = v_z w_x - v_x w_z$
 - $u_z = v_x w_y - v_y w_x$

2.2. What to do

Write a class `Vector3D` that represents a 3D vector. This class should contain the following:

- A constructor that takes in `x`, `y`, `z` components of the vector. Each parameter is of type `double`.
- Methods to get the values of individual components (e.g. `getX`, `getY`, `getZ`).
- A `toString` method that returns a string that describes this vector. This string should be of the form "`(x, y, z)`" replacing the letters with their values. Each component should be formatted to round to exactly two decimal places (look at the `String.format` method to see how to do this).
- A method `getMagnitude` that returns its magnitude.
- A method `normalize` that returns a normalized version of this vector. It should throw an `IllegalStateException` object if this operation cannot be completed.
- A method `add` that returns the result of adding this vector to another vector. It should not change the vectors that are being added.
- A method `multiply` that returns the result of multiplying this vector by a constant. It should not change the vector that is being multiplied.
- A method `dotProduct` that returns the dot product of this vector and another vector. It should not change the two vectors.
- A method `angleBetween` that returns the angle between two vectors in degrees. It should not change the two vectors. It should throw an `IllegalStateException` if this operation cannot be completed.
- A method `crossProduct` that returns the cross product of this vector and another vector. It should not change the two vectors.

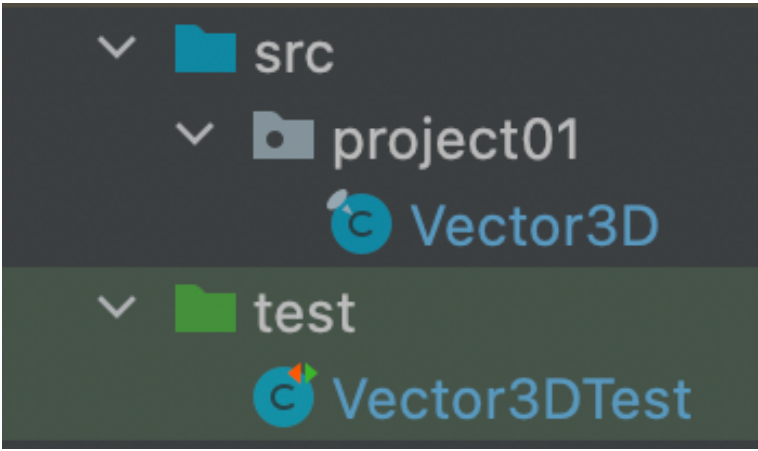
For each method, work in this order:

- Design the signature of the method.
- Write Javadoc-style comments for that method.
- Write tests that check that the method works as specified in all cases.
- Write the body of the method.

You may find the Math class useful for this assignment. Look at the [documentation for the Math class](#) and try to understand how to use it.

3. How to submit

Make sure your code is structured as follows (name of the package `project01` must be used in `src`):



[Here you can read](#) about the different ways you can upload your programming assignments on Gradescope.

You have three options for submitting your work:

- Drag and drop your zip file of your code into Gradescope (only include `src` and `test` folders)
- Submit a GitHub repository
- Submit a Bitbucket repository

If you do "drag and drop", your submission should look something like this:

Submit Programming Assignment

Upload all files for your submission

SUBMISSION METHOD

☒ Upload ☐ GitHub ☐ Bitbucket

Add files via Drag & Drop or Browse Files.

NAME	SIZE	PROGRESS	x
src/project01/Vector3D.java	4.9 KB	<div></div>	
test/Vector3DTest.java	5.4 KB	<div></div>	

Upload

Cancel

Autograder Results

ResultsCode

Test for add (5/5)

Test 2 for angleBetween, throwing exception (2.5/2.5)

Exception thrown

Test for toString (5/5)

Test for getMagnitude (5/5)

Test for dotProduct (5/5)

Select a question.

More

Request Regrade

Download Submission

Next Question

Project 1 - Vector 3D

19 Days, 21 Hours Late

Select each question to review feedback and grading details.

Student Yunyu Guo

Total Points 100 / 100 pts

Autograder Score 50.0 / 50.0

Passed Tests