# Computing the dot product of two vectors

This project involves implementing and comparing two programs that calculate the dot product of vectors - one in C and another in Java. The implementations will support various numeric data types including integers (int, short, signed char) and floating-point numbers (float, double).

## Requirements

1. Develop separate programs in C and Java that compute the dot product of two vectors.
2. Implement execution time measurement for the computation. The measured time should demonstrate a positive correlation with increasing vector lengths.
3. Conduct a performance comparison between the C and Java implementations. Analyze and explain any observed performance differences, considering various data types (`float`, `double`, `int`, `short`, `signed char`). Investigate whether the C implementation demonstrates significant speed advantages over Java.
4. Extend the analysis to identify noteworthy patterns or insights, such as performance variations across different data types, impact of language-specific optimizations, memory usage characteristics.

## Tips:

1. As efficicomputational efficiency is the primary focus, you can use randomly generated vector values.
2. Use native data types directly. No custom arbitrary-precision types are required.
3. Employ precise timing mechanisms:
    - In C: Use `<time.h>` functions. But be careful that function `clock()` may not provide correct time.
    - In Java: Utilize `System.nanoTime()`

## Rules:

1. The project report and the source code must be submitted before the deadline. Any submission after the deadline (even by 1 second) will result in **a score of 0**. The deadline is 23:59 on March 30.
2. The files should be submitted as report.pdf, dotproduct.c, and Dotproduct.java. Use exact filename capitalization and extensions. The files should **NOT** be compressed into one.
3. The score will depend on the quality of both the source code and the report. The report should be easy to understand and provide a clear description of the project, especially the highlights.

4. Attention should be paid to code style. Adequate time is given for code to be written correctly and with good style. Deductions will be made for poor code style. Code style guides, such as the Google C++ Style Guide (http://google.github.io/styleguide/cppguide.html), can be used as a reference.