CS301

Programming Assignment #2

Thomas Crow

**Source code:**

Matrix\_multiplication.cpp

**Text

Description automatically generated**

Matrix\_Multiplication.java

Text

Description automatically generated

Matrix\_Multiplication.py

Text

Description automatically generated

**Runtime Screenshots:**

250x250 Matrices

|  |  |  |
| --- | --- | --- |
| C++  Text  Description automatically generated | Java  Text  Description automatically generated | Python  Text  Description automatically generated |

500x500 Matrices

|  |  |  |
| --- | --- | --- |
| C++ Text  Description automatically generated | Java  Text  Description automatically generated | PythonText  Description automatically generated |

1000x1000 Matrices

|  |  |  |
| --- | --- | --- |
| C++Text  Description automatically generated | JavaText  Description automatically generated | Python Text  Description automatically generated |

1500x1500

|  |  |  |
| --- | --- | --- |
| C++Text  Description automatically generated | JavaText  Description automatically generated | Python Text  Description automatically generated |

2000x2000 Matrices

|  |  |  |
| --- | --- | --- |
| C++ Text  Description automatically generated | Java Text  Description automatically generated | Python Text  Description automatically generated |

**Runtime graph:**

**Conclusion:**

Java and C++ were by far faster in these tests than Python. As you can see in the graph, Python poor performance makes the other two languages virtually indistinguishable. This is likely due to Java and C++ being compiler-based languages versus the interpreter-based Python. None of the languages when a library wasn’t in use took advantage of parallel processing. Utilizing parallel processing, either built into the language or via a library, would greatly increase performance. As would optimized code that using a library like numpy would provide.