Report on
A comparative analysis of standard data structures in Java and C++

Submitted to
Professor Martha Wall
for
English 316
Brigham Young University
Provo, Utah
March 29th, 2013

by
Andrew Jensen
Joey Cozza

Brigham Young University

March 29th, 2013

Martha Wall, Ph.D.
4048 JKB
Brigham Young University

Dear Professor Wall,

Our technical report, *A comparative analysis of standard data structures in Java and C++*, details original research that was conducted in order to quantify certain differences in Java and C++. We consider the use of abstract data structures in terms of runtime performance, algorithmic complexity, general usage. We also provide recommendations for programmers that are considering which programming language would be most appropriate for their applications.

While our report was written specifically for programmers, the information is presented in a way that is accessible to non-programmers as well. Several graphs and diagrams have been included to help explain the purpose and implementation of abstract data structures.

We learned several interesting facts about the two languages while conducting our research. C++ code performed better than Java code, as we predicted; however, the difference was not as dramatic as we expected. Java code fluctuated quite a bit in its runtime performance, but the data structures from the built-in Java library were more consistent than our own implementations.

Thank you for the opportunity to perform this research. The process of preparing this report has been beneficial to the two of us and our findings will be useful for programmers in the future.

Respectfully,

Andrew Jensen
Joey Cozza

Table of Contents

List of Tables

List of Figures

Abstract

The choice of programming languages can have large effects on the end result of a program. Unfortunately, there has not been much research to compare individual programming languages with another, so at times it is difficult to know which language would be more effective to use. We decided it would be beneficial to programmers of all expertise levels to have a side-by-side comparison of Java and C++, two prolific programming languages. In order to narrow our research, we focused our study on the differences in Data Structures between the two languages. We began by making our own programs to time how long each language took to run some repetitive tasks. We also made our own implementations of some common data structures including Sets, Sorted Sets, Linked Lists, and Array Lists. We were expecting C++ to run more quickly, but Java to be easier to write for the programmer. In addition, we also assumed that our handwritten structures would always run slower than the standardized data structures. In general, the results were as we expected, but there were a few outliers that we hadn't anticipated. The most interesting output we saw was when our own implemented data structure effectively took no time to run, while the standard one took a full second to perform. We analyzed that output, and came to the conclusion that C++ was able to automatically optimize certain code that was handwritten, but not code that was included from a static library. This means that handwritten C++ data structures, on average, performed better than all of the other configurations we tested. We suggest that novice programmers create their programs in Java, using the Standard Java Library data structures for simplicity's sake. We suggest that advanced programmers focusing on speed implement their own data structures in C++ to tailor their structures to their programs' needs and be as efficient as possible. Our research is available online and can be extended easily by other programmers in the future.