

Objectives:

The primary objective of this lab assignment is to implement pointN class along with base class Element. Using these class develop a single-threaded merge sort algorithm in Java, tailored for n-dimensional points and Extend the merge sort implementation to a multithreaded version in Java.

Learnings:

- **Language Transition Skills:** This task enhances proficiency in transitioning code between different programming languages, particularly from C++ to Java.
- **Class Translation Understanding:** Understanding how to translate classes, especially those with inheritance relationships (base Element class), helps in grasping the nuances of object-oriented programming across languages.
- **Algorithm Design and Implementation:** Implementing a single-threaded merge sort algorithm for n-dimensional points fosters skills in designing and implementing complex algorithms.
- **Multithreading Proficiency:** Implementing a multithreaded merge sort extends knowledge to concurrent programming, emphasizing the importance of managing parallel execution and synchronization.
- **Polymorphic Base Class Usage:** Utilizing polymorphic base class references demonstrates an understanding of polymorphism, a key concept in object-oriented design.

Challenges:

- **Syntax Differences:** Identifying and adapting to syntax differences between C++ and Java, such as memory management, access modifiers, and class structure.
- **Standard Library Variances:** Dealing with differences in standard libraries and their functionalities between C++ and Java.
- **Algorithm Complexity:** Ensuring a deep understanding of the merge sort algorithm, especially when adapting it for n-dimensional points.
- **Concurrency Issues:** Managing and synchronizing concurrent threads to avoid race conditions and data inconsistency.
- **Thread Safety:** Ensuring that the multithreaded implementation is free from deadlock and other common concurrency pitfalls.

Conclusion:

the combined program, spanning the porting of the PointN class to Java, the implementation of merge sort algorithms, and the introduction of multithreading, provides a rich learning experience. It hones language transition skills, algorithmic proficiency, and concurrency management.