

## CECS 342 - Section 1

### Assignment 1 - Observation

#### Group 28

Having explored both C and Haskell, our group has discovered significant differences in their programming methodologies. C employs a procedural approach, executing code step-by-step, while Haskell emphasizes defining relationships between values as a functional language. This contrast is apparent in their data operation priorities, as C focuses on performing operations while Haskell prioritizes defining relationships. The type systems of C and Haskell also exhibit notable dissimilarities. C uses dynamic typing, determining a value's type at runtime, while Haskell requires static typing with the type declared before compilation. For example, the quicksort function in Haskell uses "`Ord a => [a]`," indicating that the elements must be orderable.

Despite their distinctions, C and Haskell share some similarities, such as support for recursion and built-in arrays. Control structures like if-else statements and for loops exist in both languages, albeit with differing syntax and implementation. In our group's experience, working with Haskell is more challenging and time-consuming than working with C. Haskell demands a distinct approach to programming. Nevertheless, comparing the similarities and differences between these two languages has helped us appreciate the importance of weighing trade-offs between performance, readability, and maintainability when selecting a language.

The comparison between C and Haskell demonstrates that different programming languages may adopt unique approaches to problem-solving and data handling. Understanding their differences and similarities can assist programmers in broadening their skills and knowledge in the field, which is especially relevant when working on group projects that require choosing the most suitable programming language for the task at hand.