Cross-Language Expense Tracker C++ and Python

Presented by Samrat Baral

Project Overview

- Purpose: Build and Compare Implementations
  - Expense Tracker Application
  - Add, Filter, Summarize Expenses
  - Compare C++ vs Python approaches

C++ Implementation

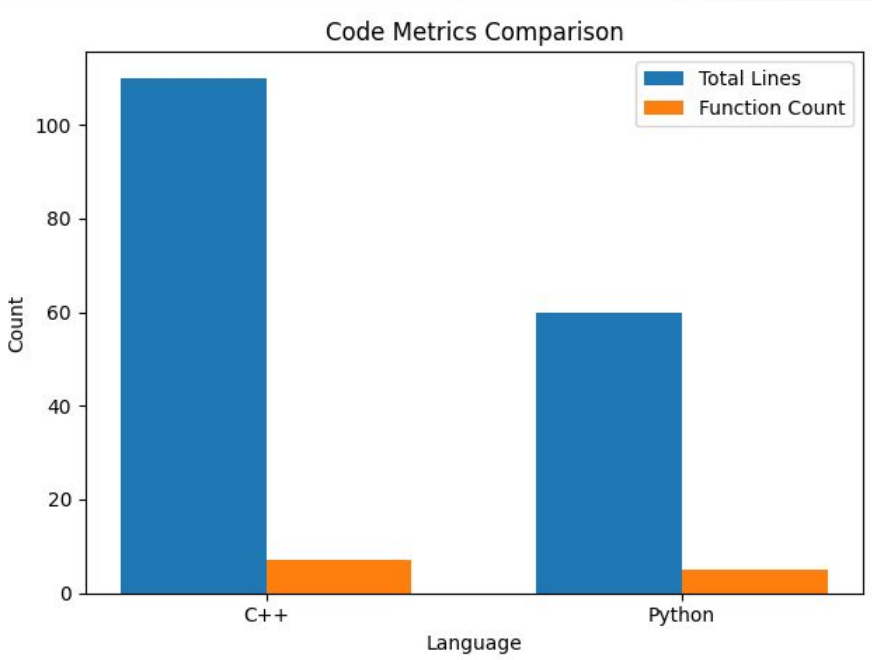
- Features:
  - Structs and Vectors for Data
  - Smart Pointers for Memory Safety
  - Manual Formatting and Iteration

Python Implementation

- Features:
  - Dictionaries and Lists for Data
  - Dynamic Typing and Garbage Collection
  - List Comprehensions for Filtering

Demonstration Results

- Observations:
  - Both versions produced identical outputs
  - C++: Manual memory and fixed structure
  - Python: Rapid development and flexible handling



Key Differences and Insights

- C++: Strong Typing, Performance, Verbose
- Python: Rapid Development, Flexibility, Readable
- Trade-offs: Speed vs Productivity

Conclusion

Final Thoughts:

- Both languages succeeded
- Language choice depends on project needs
- Practical learning from implementation differences