

# Junior Software Engineer C++ Programming Task: Address Book

## Summary

Using standard C++ and only its standard library, design and implement code that models an address book.

## Minimum Requirements

- Support from zero to a “large number” of entries (typically 100,000s).
- Entries should contain:
  - A first and/or last name (you can treat each character as a single byte, i.e. no need to support Unicode).
  - An optional phone number.
- Provide functionality to:
  - Add an entry.
  - Remove an entry.
  - Retrieve entries in alphabetical order, sorted by first name.
  - Retrieve entries in alphabetical order, sorted by last name.
  - Retrieve entries whose first or last name exactly or partially match a supplied search string, from the beginning of the name (e.g. searching for “Dan” would match any entries with “Daniel” as well as any “Dan”).
  - Make this matching case insensitive.
- Implement the tests to demonstrate all of the address book’s functionalities:
  - It is fine to hard-code example entries and expected test results
  - Add any additional tests as you see fit.
- Provide documentation for the API and to justify your design choices.
  - Inline comments in the code are fine.

## Additional Considerations

- Imagine that the address book code could be used by other developers, in various client applications.
- Assume that the retrieval operations are going to be called the most frequently in the majority of client applications.

## Notes

- Feel free to use C++ and standard library references, e.g. [cppreference.com](http://cppreference.com), but please do not ask for help with the specifics of this task. It should be your own work.

- Feel free to comment on any known shortcomings of your design and make suggestions about how these could be addressed given more time or if the library and version restrictions were removed.
- We've provided a sample framework/starter template **address-book-starter-template-main** to assist you so you can focus on your implementation. Feel free to modify it in any way as you see fit. More details in the **README.md** file in the project.