

Coding Assessment

The problem domain is deliberately simple, and you could very easily write an extremely terse solution that satisfies the requirement. But our goal is not to see you implement a trivial sorting algorithm. Most importantly it is to understand how your code communicates its purpose clearly and with empathy to your potential team members. What do we mean by empathy? Empathy here is caring about how easy your code is to understand and navigate for the next engineer who touches it. Secondly it is to understand your ability to compose quality code that adheres to SOLID ([https://en.wikipedia.org/wiki/SOLID_\(object-oriented_design\)\)](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design))) principles.

So, give us your best effort, a solution that you are proud of.

The Goal: Name Sorter

Build a name sorter. Given a set of names, order that set first by last name, then by any given names the person may have. A name must have at least 1 given name and may have up to 3 given names. Example Usage Given a file called unsorted-names-list.txt containing the following list of names:

```
Janet Parsons
Vaughn Lewis
Adonis Julius Archer
Shelby Nathan Yoder
Marin Alvarez
London Lindsey
Beau Tristan Bentley
Leo Gardner
Hunter Uriah Mathew Clarke
Mikayla Lopez
Frankie Conner Ritter
```

Program should read names from a text file.

Should print the sorted names to screen

```
Marin Alvarez
Adonis Julius Archer
Beau Tristan Bentley
Hunter Uriah Mathew Clarke
Leo Gardner
Vaughn Lewis
London Lindsey
Mikayla Lopez
Janet Parsons
Frankie Conner Ritter
Shelby Nathan Yoder
```

Assessment Criteria

We will execute your submission against a list with a thousand names.

Your submission must meet the following criteria.

- The solution should be available for review on GitHub.
- The names should be sorted correctly.
- It should print the sorted list of names to screen.
- It should write/overwrite the sorted list of names to a file called sorted-names-list.txt.