

(3 points) 1. Why do the adjectives and nouns files contain the total number of words on the first line? What would happen if we tried to load a file that does not contain this number? What would need to change in the code to avoid this problem?

The first line tells the program how many words to expect, so we can size the array up front and read exactly that many lines. It keeps the loader simple and fast without using extra data structures. If a file didn't have that number, our current loadTxt() would try to parse the first word as an integer and crash (NumberFormatException), or if the count didn't match the real lines, we'd end up with nulls or miss items. To avoid this, I'd change the reading logic to not rely on the header at all: read line-by-line into a dynamic list (e.g., ArrayList<String>), trim/skip empties, and then convert the list to a String[] at the end. Another option is two passes (first count, then read), but the ArrayList approach is cleaner. I'd also add a small validation step: if the header exists, compare it with the actual number of lines and warn if they differ.

(2 points) 2. As you may have noticed, it is quite easy to generate a band name that can be considered inappropriate. Since you would like to tell your grandparents about your band too, you prefer an appropriate name. How would you go about making sure that the generated name is appropriate? What are some of the challenges that come with doing so?

To keep names appropriate, I'd filter both the source words and the final combined phrase. Easiest start: a blacklist of bad words. I'd run a check on each randomly picked adjective/noun, and also on the full "Adjective Adjective Noun" string to catch combos that become weird only when placed together. If I wanted to go further, I could use categories (e.g., remove sexual, violent, slur, or drug terms). The hard parts: context and slang change fast, so you'll miss some or block harmless words; different languages have different thresholds; and maintaining the list is ongoing work. There's also a risk of false positives (blocking normal words) and false negatives (letting a new slang term slip through).