← **Project review - APG1 Bootcamp. Day01**

📄 Type of project	Individual
🕐 Duration	30 min
☑ Passed Peer Reviews	0/3

Git project ⌃

Student ⌃

👤	**tbirdper@student.21-school.ru**	level 10

About ⌃

Introduction

The methodology of School 21 makes sense only if peer-to-peer reviews are done seriously. Please read all guidelines carefully before starting the review.
- Please, stay courteous, polite, respectful and constructive in all communications during this review.

- Highlight possible malfunctions of the work done by the person and take the time to discuss and debate it. The pedagogy is valid only and only if peer-evaluation is conducted seriously.
- Keep in mind that sometimes there can be differences in interpretation of the tasks and the scope of features. Please, stay open-minded to the vision of the other.
- If you have not finished the project yet, it is compulsory to read the entire instruction before starting the review.

Guidelines

- Evaluate only the files that are in src folder on the GIT repository of the student or group.
- Ensure to start reviewing a group project only when the team is present in full.
- Use special flags in the checklist to report, for example, an "empty work" if repository does not contain the work of the student (or group) in the src folder of the develop branch, or "cheat" in case of cheating or if the student (or group) are unable to explain their work at any time during review as well as if one of the points below is not met. However, except for cheating cases, you are encouraged to continue reviewing the project to identify the problems that caused the situation in order to avoid them at the next review.
- Doublecheck that the GIT repository is the one corresponding to the student or the group.
- Meticulously check that nothing malicious has been used to mislead you.
- In controversial cases, remember that the checklist determines only the general order of the check. The final decision on project evaluation remains with the reviewer.

MAIN PART   ⌄

Exercise 00 - Reading

Check that application can read provided example XML file without errors
Check that application can read provided example JSON file without errors
Try passing JSON/XML files with broken format and check that program prints a meaningful error
Check that application prints a corresponding JSON when being passed an XML file
Check that application prints a corresponding XML when being passed an JSON file
Check that in both cases 4-space indent is used for pretty-printing
Check that application code contains a `DBReader` interface and two structs that implement this interface - one for XML and one for JSON
Check that application decides which implementation to use based on specified file extension

No	✓ Yes

Exercise 01 - Assessing Damage

Check that changing units and counts for ingredients is correctly detected and printed
Check that removal or addition of a unit field is correctly detected and printed

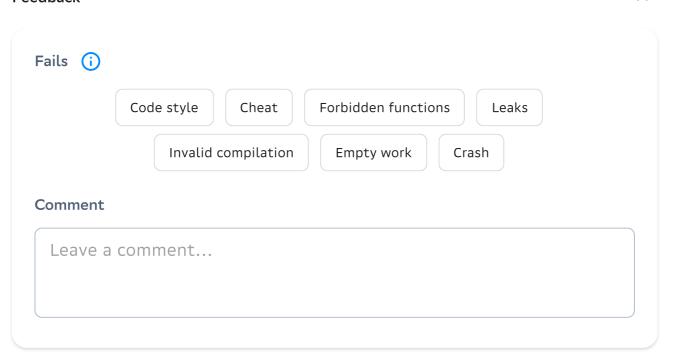| No | ✓ Yes |

Exercise 02 - Afterparty

Check that at no time both files are fully loaded into memory
Check that adding or removing new filepath is correctly detected and printed
Check that the program works correctly with empty files (both old and new)

| No | ✓ Yes |

Feedback ⌃

Fails ⓘ

| Code style | Cheat | Forbidden functions | Leaks |

| Invalid compilation | Empty work | Crash |

Comment

Leave a comment...

✓ Review