Project Assignment 1:
Preparation

Due: Friday, 9/30/2022, 23:59h

Preamble

This project is intended to prepare you for the exam and to give you some more practical experience in software quality assurance. Over the course of three to four milestones, you will incrementally build a detailed software quality report on a substantial project.
Please complete it together with another homework team, i.e., **three or four people should be working on this project**. If you have no homework partner, please find a team of three to work with. If you are a team of three without a fourth person, that's fine.
As a rule, I assume that all members of the team contribute equally to the solution. In contrast to the homework assignments, divide-and-conquer is OK, provided that you meet as a team and explain, verify, and double-check each other's work. Pay particular attention to consistency of presentation as well as findings.

Task 1

Find a piece of software that you have written in the past, e.g., for a class, for fun, or for a company. The software should satisfy the following properties:

- Written in Java or Python (if possible), JavaScript (if you must)
- An independent software with no or only few dependencies or needed libraries
- Not a framework, API, Web Application, Mobile Application, compiler, or data structure
- You should have the source code, ideally some type of specification, and should be able to compile the code.
- The specification can be Javadoc, but ideally is some type of Requirements Specification or other human-readable (and understandable) description of what the code does.

Typically, a CSC380 or CSC365 class project works. Don't pick your CSC365 b-tree (or similar) project.

If you cannot find a piece of software, ask a friend, classmate, or me to share one with you. Be sure you have the rights to the software for academic use and distribution. This means that, if you use a project from a company, employer, or even a private team project, that you have explicit permission. It will be your responsibility to be certain and I will assume that you have said permissions.

Task 2: Introduce your Project

Give a brief introduction to your project. Write a short (and concise, about 250 words) description of what the purpose of the code is and what it is used for. Your description should at least contain the following:

- Name of the software you chose
- A brief description what it does (3-5 sentences)
- Some illustrations for easier understanding
- An overview and brief description (3-5 sentences) of the most important methods, functions, components, etc.
- Again, some illustrations

In the next couple of project assignments, we will try out all the techniques we talk about in class. Your description should allow judging whether or not it is suitable to apply the QA techniques, so please include any detail you feel are necessary in order to be able to make this judgment. Only if I can ascertain your project choice will I be able to help you along the project. Also, if you do too superficial a job, there is a chance that you will have more effort later on in the project and may have to redo things.

When you have this document, remember it well for what comes next.

Task 3: Start a Quality Management Plan (QMP)

Start a Quality Management Plan for the component you have decided. You may use the template provided on Brightspace, but you will probably want to tailor it to your specific needs (and the requirements outlined herein). You may of course also make your own document. However, if you do this, please don’t just submit a series of bullets. Instead, make a document with a suitable structure, subheadings, page numbers, and such, like in the provided template.

Your quality management plan should start with a meaningful title page and an outline/table of contents. It should then contain a section “Quality Management Approach,” which subsections for the following:

1. An identification of the system under test, incl. it's components, classes, methods, functions, attributes, and operations;
2. A brief synopsis what the purpose of the system is and how it fits into the context of its use (remember Principle 1 of QA);
3. A glossary of important terms that are important (this list may be short at first);
4. A list of all project partners, their contact details, and their responsibilities;
5. A brief synopsis how you plan on testing this product.

Obviously, you can use your discussions from Task 2 to address these points. Find the appropriate areas in the template to enter your solution for Task 2. In other words: I will need only the QMP to be submitted, no need for an extra document for Task 2.

Task 4: Define Quality Properties, Metrics, and Criteria

Now, extend your quality management plan with another section on Quality Properties, Metrics, and Criteria. From ISO 25010, find useful set of quality properties given the component under test. You may even want to define your own. But bear in mind that all project milestones will only deal with unit testing, so your quality property should be able to be determined through unit testing only. For each property, find a way to measure it, and select a suitable criterion (upon which you'd accept the property as 'satisfied'). Write down your definitions and be, if necessary, update your glossary. Example:

Quality Property	**Definition**	**Metric**	**Criterion**
correctness	Executed test cases yield the same result that anticipated during their derivation.	Systematic derivation and execution of test cases for specification-based, control flow based, and data flow based tests.	The component is considered correct, iff at least • 100% of test cases with high criticality pass • 80% of test cases with medium criticality pass • 40% of test cases with low criticality pass.

Don't overdo this. A choice of 3-5 quality criteria will be sufficient. "Correctness" should probably be certainly be one of them, but only you can make that determination depending on what system you picked.

Task 5: Conduct the Specification-based Test

Derive a set of test case that would allow to test all methods against their specification. The specification may include header files, but also natural limitations (e.g., there can never be a negative number of students in a class). Specifically, do the following:

a) For each parameter of each method of each class, generate a complete set of equivalence classes. Document them as precisely as you can (e.g., set-theoretic notation, etc.).

b) For each equivalence class of each parameter, find a representative.

c) Conduct Boundary Value Analysis and find one representative to test each boundary, along with its inpoint and outpoint. If your equivalence classes are all-encompassing, that means there's nothing to do here, but be sure that there isn't (i.e. state the reason why you can't conduct a Boundary Value Analysis for some equivalence class).

d) Aggregate all representatives to test cases, which test each equivalence class. Remember that for valid equivalence classes, you can aggregate multiple representatives of different parameters into one test case, but not for invalid equivalence classes!

e) For each valid and invalid test case, find the expected result. Again, refer to the available documentation, if necessary.

You may want to use the templates provided in the previous Assignment Sheets to help you document your results. **You may also want to create ONE table for all your test cases.**

Document your findings in a new section "Specification-Based Testing" in your Quality Management Plan. Create appropriate subsections for each class and method.

Task 6: Submit and Present

Submit the PDF of your finished Quality Management Report for Project 1. Prepare yourself to present your findings in class. Don't panic.