

CS304 Project: Project Proposal (Total: 20 points)

Deadline: 12 March 2021, 11.59pm

The class project (Invitation links: <https://classroom.github.com/g/PNJJP9gt>) is a group-based project where students will be divided into groups of 4-6 (Min: 4, Max: 6). The objective of the project is to allow student to apply the knowledge that they learned in the class to contribute to real-world Java applications. Each group need to choose 1-2 Java projects from the list of open-source Java projects at:

<https://github.com/dkorobtsov/automation-arsenal/blob/master/java/> or other projects listed in “Other projects”.

All group members should carefully discuss and agree upon 1 (or 2) Java projects. Once the project has been selected, your group cannot change to select other projects. The project needs to be selected based on the criteria listed below:

- **Ease of use:** Each member of the team should be able to compile the Java project successfully in his or her computer without errors. The app could be executed in a device. *All selected projects should fulfill this criterion:*
- **Existing Tests:** The project contains some test cases for checking for regression errors. *All selected projects should fulfill this criterion.*
- **Popularity:** The number of stars in GitHub should be greater than 100.
- **Actively Maintained:** There are recent commits (within a year) to the projects.
- **Number of open GitHub issues:** The project should have at least 15 open (unresolved) GitHub issues that are bugs-related/feature-related (You need to check if the 15 GitHub issues are bugs/features instead of questions or documentation). *All selected projects should fulfill this criterion.*

For the selected Java projects, you need to select 5-20 issues based on the following **guidelines for selecting issues:**

- **Importance of the issues:** The selected issue should be important. For example, important issues have tags like “up for grab” or “help wanted”. For most open source projects (e.g, INRIA/spoon), you could find the issues that are suitable for beginners under the “contribute” link (e.g, for INRIA/spoon, the “contribute” link is <https://github.com/aaa/spoon/contribute>).
- **No fixing commit:** The issue should not have any fixing commit. **Do not select any issue if a contributor has volunteered to fix it. This is a compulsory criterion.**
- **Reproducible:** For each selected issue, you need to make sure that the issue is reproducible and you will need to write a test case for reproducing the issue (this test case could be a system-level test or test included in the issue). Do not select an issue

if you do not understand the issue and cannot reproduce it. Do not select an issue if someone mentioned in the comment that it is not reproducible. **This is a compulsory criterion.**

- **Estimated lines of code to resolve issue:** The selected issue should be non-trivial to resolve. The estimated lines of code needed to resolve issue should be greater than adding/modifying 10 lines of code.
- **Estimated time to resolve issue:** The estimated time taken to resolve one selected issue should be at least 1-2 weeks instead of 1-2 days.

1. List the names and student ids of your group members. (1 point)
2. Each group should select a unique group name. The group name will be used for the setting up the GitHub repository. List the group name that your group has selected. (2 points for successfully setting up your group GitHub repository)
3. Each group needs to select 1-2 projects. To ensure that each project gets to test different projects, each group needs to post on Tencent Doc:

<https://docs.qq.com/sheet/DWE94aEN2cWJWahVn>

for each of the selected project. For each project, a maximum of three teams could choose the same projects. The selection is based on first-come-first-served basis, which means that the first three group to post the name and link of the selected project wins the bid for that project and other groups need to select other project. What are **names** and **links** to the open-source project that your group select?

(2 points)

4. Why do your group select this project? Explain the reason in terms of the criteria below: (Total: 17 points)

- Ease of use: Have you successfully compiled the project? (2 points)
- Existing Tests: Do the project contain some test cases? (2 points)
- Popularity: How many stars does the project have in GitHub? (2 points)
- Actively Maintained: When is the latest commits to the project? *Please make sure that the latest commits were within a year ago* (2 points)
- Number of open GitHub issues: How many open issues the project have? (2 points)
- Does the project have any contributing guidelines (Contributing guidelines are usually at <https://github.com/aaa/bbb/blob/master/CONTRIBUTING.md> where aaa/bbb is the name of the project)? If yes, include the link of the contributing guidelines. Make sure that when fixing the issue, your group member will follow the contributing guidelines (2 points)
- Diversity of team members: The team members should be diverse with each member playing different roles in a software development team. Include a division of roles for each team member for the first iteration of the project. Note that for each iteration, the member will get to take up different roles. (5 points)

Name	Role
	Leader
	Developer
	Developer
	Designer (Contribution guidelines & code review)

	Tester
	Developer & Documentation (JavaDoc)

1. Select 5-20 issues to fix. Each issue needs to be selected by following the “**guidelines for selecting issues**”. The number of selected issues depends on (1) the number of group members, (2) the difficulty of the selected issues, and (3) the estimated time to fix the issue (there may be some related issues that could be fixed together so you will need to adjust the estimated accordingly). Note that you need to consider the time taken in understanding the project and the issue (1 week for learning is allowed for each issue). The total estimated time for the group members should be at least 6 weeks per person. This means that if you have x team members in your group and the total estimated time for fixing all issues should be 6 multiplied by x (Total=6x). For example, if an issue takes 3 weeks to fix, then each person could choose to fix 2 issues for the entire project. **Please plan your issues accordingly, fixing those issues that are more likely to be fixed by others first.** For all the selected issues, create a table such as below: (8 points)

****All answers for the above questions should be posted in README.md. 1 point will be deducted if the file name is different.***

Frequently asked questions and answers:

1. Could I have less/more people in my group?

Link to issues	Type of issues (Bug/Feature)	Estimated Time to fix each issue	Number of people for fixing this issue (≤ 2 members for each issue)	Estimated Difficulty (in a scale of 1-5, 1 means very easy to fix and 5 means very difficult to fix)
https://github.com/INRIA/spoon/issues/3118	Feature	3 weeks	1	5
...				
Total: 50 weeks				

Answer: The requirement state clearly that the allowed number of team members is 4-6 people per team. Having less people in group is not recommended will reduce the number of “roles” you have in your team. Having more than 6 people will be a too large group that makes coordinating and communicating among team members to be challenging.

2. What if my selected issues have been fixed?

Answer: You should select issues that have not been fixed before. If your selected issues have been fixed, then you will need to select other issues.

3. Can I select other Java projects are not listed at the link?

Answer: You can select other Java projects as long as it satisfies all of the criteria listed above but the projects need to be a Java project because you need to write JUnit tests for testing your implementation.

4. What if there are not enough open issues that fulfilled the above criteria for one project?

Answer: You can select up to 2 Java projects for getting the GitHub issues to fix.

5. Can we have a repository for the class project where we can use for the entire semester?

Answer: Yes, we provide a main repository for you to commit your code for the project for the entire semester in this link: <https://classroom.github.com/g/yV-OP75>. There is no deadline for this repository so you can use it for the entire semester even after finishing the class.

Other projects:

<https://github.com/Sable/soot>

<https://github.com/randoop/randoop>

<https://github.com/javapathfinder/jpf-core>

<https://github.com/typetools/checker-framework>

<https://github.com/pmd/pmd/>

Below is the list of projects where SUSTech students have contributed to last year (these projects have histories of accepted pull requests from students):

<https://github.com/joel-costigliola/assertj-core>

<https://github.com/DiUS/java-faker>

<https://github.com/jhy/jsoup>

<https://github.com/JabRef/jabref>

<https://github.com/kennycason/kumo>

<https://github.com/stefan-niedermann/nextcloud-notes>

<https://github.com/remkop/picocli>

<https://github.com/perwendel/spark>
<https://github.com/alibaba/fastjson>
<https://github.com/square/javapoet>
<https://github.com/selenide/selenide>
<https://github.com/rampatra/jbot>
<https://github.com/iluwatar/java-design-patterns>
<https://github.com/halo-dev/halo>
<https://github.com/halo-dev/halo-admin>
<https://github.com/square/retrofit>
<https://github.com/google/guice>
<https://github.com/pmd/pmd>
<https://github.com/ankidroid/Anki-Android>
<https://github.com/ronmamo/reflections>
<https://github.com/jOOQ/jOOL>
<https://github.com/GoogleContainerTools/jib>
<https://github.com/OpenRefine/OpenRefine>
<https://github.com/spotbugs/spotbugs>