

**MP3: Debugging and Collaborative bug fixing** (Total: 40 points + 5 bonus points)

Deadline: 18 December, 11.59pm

The goal of MP3 (Invitation link: <https://classroom.github.com/a/YuFdYfZF> ) is for students to learn about debugging and collaborative bug fixing. Assume your selected app is A.

**List your name and student id in the README.md.**

**All answers (except for the code modifications) should be written in README.md.**

1. Select a total of 10 issues from your app A (5 issues are for part a and 5 issues are for part b). *Each selected issue should be (1) open issue that has not been resolved, and (2) bug-related issue (do not select a feature request nor a question).*

a) For **5** of these **open** issues, select 5 **similar closed** issues from your App A.

*You can search for similar closed issues in App A by searching for some keywords that described the open issue (e.g., if the open issue is about `IllegalStateException`, you can search for “`IllegalStateException`” in your open issue or if the open issue is about an incorrect text being displayed in screen B, you can search for “screen B”). **Include the link for the selected issue.** Answer the following questions for each issue:*

- i. Why is this closed issue selected? What are the keywords that you used to search for this closed issue? (5 points, 1 point for each issue)
- ii. Could you find the buggy location (the buggy lines/method/UI component that cause the error/bug) based on the selected closed issue in your selected app in A? If yes, explain why do you think that this location is buggy. If not, explain the reason (you could use the stack trace to find the location for a crash). (5 points, 1 point for each issue)

iii. Could you fix the open issue based on the selected closed issue in your selected app in A? If not, explain the reason. (5 points, 1 point for each issue)

b) For **5** of these **open** issues, select **5 similar closed** issues from another app. You can search for similar closed issues using either (1) search from the issues from apps within the same category by referring to the list of similar apps at <https://github.com/pcqpcq/open-source-android-apps/blob/master/README.md> or (2) search for some keywords describing the bug in <https://github.com/search?q=illegalstateexception&state=closed&type=Issues> (replace “illegalstateexception” with your keywords). Answer the following questions for each issue. **Include the link for the selected issue.**

- i. How is the closed issue selected? Did you use (1) search for issues from similar apps or (2) search for some keywords in GitHub? (5 points, 1 point for each issue)
- ii. Why is this closed issue selected? What are the keywords that you used to search for this closed issue? (5 points, 1 point for each issue)
- iii. Could you find the buggy location (the buggy lines/method/UI component that cause the error/bug) based on the selected closed issue in your selected app in A? If yes, explain why do you think that this location is buggy. If not, explain the reason (you could use the stack trace to find the location for a crash). (5 points, 1 point for each issue)

- iv. Could you fix the open issue based on the selected closed issue in your selected app in A? If not, explain the reason.  
(5 points, 1 point for each issue)

*2. Compare the bug-fixing method in Question 1a (Searching from Different Issues within the same app) and Question 1b (Searching from Issues from different apps) by answering the following question:*

- a) The method in Question 1a is more efficient (time taken) than the method in Question 1b. *(Select one answer, 1 point)*
- i. Strongly Disagree
  - ii. Disagree
  - iii. Undecided
  - iv. Agree
  - v. Strongly Agree
- b) The method in Question 1a is more effective (likelihood of finding a relevant fix) than the method Question 1b. *(Select one answer, 1 point)*
- i. Strongly Disagree
  - ii. Disagree
  - iii. Undecided
  - iv. Agree
  - v. Strongly Agree

c) Compared to referring to similar issues (using methods in Question 1a and Question 1b), I think that manually finding and fixing bugs directly is more efficient (time taken). *(Select one answer, 1 point)*

- i. Strongly Disagree
- ii. Disagree
- iii. Undecided
- iv. Agree
- v. Strongly Agree

d) Compared to referring to similar issues (using methods in Question 1a and Question 1b), I think that manually finding and fixing bugs directly is more effective (likelihood of finding a fix). *(Select one answer, 1 point)*

- i. Strongly Disagree
- ii. Disagree
- iii. Undecided
- iv. Agree
- v. Strongly Agree

e) Explain your answers in a)-d) by including either (1) the number of bugs successfully fixed with a method to support your claim, or (2) a suggestion to improve the methods (1a or 1b) above. *(1 point)*

You can do either of the following to get bonus points (5 points):

- a) If you read an open GitHub issue and could use the information in the related GitHub issues for finding the **fix** for the open issue, file a **pull request** for your selected apps and comment in GitHub by posting at: <https://github.com/orgs/cs409-software-testing2020/teams/allstudents> *(add the link for pull request)*.
- b) If you find a bug, post the bug that you find in GitHub by posting in your team discussion. If all your team members checks that your bug reports fulfilled the bug reports requirement, post it at: <https://github.com/orgs/cs409-software-testing2020/teams/allstudents>