

Section 1 – Short answer

1. Describe a software engineering skill you've developed through this course.

Answers may vary

2. Describe 1 similarity and 1 difference between software engineering vs other engineering disciplines.

Software engineering projects proceed through a series of stages from planning to finished product, but software is vastly more scalable, so one project can impact billions of people easily.

3. Describe a condition or restriction found in a software license that might prevent some development teams from using the code with that license.

License might require payment, or for the vendor to publish the source code for their own product that uses the licensed software.

4. Describe one of the four types of software maintenance.

See lecture 3 slide 23

5. Describe a possible reason why a block of code might need to be refactored.

Complexity is excessively high, or code has lost cohesion, meaning it accomplishes unrelated functions.

6. Define a corner case in software quality assurance.

The intersection of two edge cases – when multiple inputs (or properties of the input) are at an extreme value for their range. Empty strings or arrays, null references, and very large strings or arrays are typical examples.

7. Describe a bug you've observed in a piece of software you've developed or used.

Answers may vary

8. Define the relationship between Mean Time To Recover and software downtime.

Downtime is the percentage or fraction of time that a software is not operating correctly. Mean Time to Recover measures the average duration of a period when it is not operating correctly, and they are related by the equation "downtime = MTTR/MTBF", which gives that fraction.

9. Describe the role of a Quality Assurance engineer within a software development team.

The role of a QA engineer is to ensure quality is maintained at a high level throughout the lifetime of a software product.

10. Describe one department or role outside of the software engineering team that might need to approve a release candidate for distribution, and why they might need to do so.

Marketing department may need to certify that they are ready to capitalize on the announcement of new features with marketing materials coinciding with the software release.

11. Describe a business task outside of product development that can be automated with software.

Answers may vary.

12. Describe one of the 12 Agile Alliance principles.

See lecture 3 slide 16

13. Describe the expected state of a software project at the “code complete” milestone.

Implementations of all functionality that will be included in the release should be working, with only minor changes and bug fixes to follow for the remainder of the release process.

14. Describe the software product or system currently operating that you believe has the largest impact on society today.

Answers may vary

15. Version 4.3 of a product has been released. What is one factor that might be considered in the decision as to whether the next release is numbered as version 4.4 or version 5.0?

Significance of new functionality or degree of compatibility with functionality in version 4.3

16. Describe a way in which software accessibility considerations may impact product design.

Accessibility guidelines make software more robust for all users by adding design elements and constraints that make the software easy to use. For example, by conveying hyperlinks with both blue text AND underlining, all users can clearly see when text is a hyperlink, whereas if the product design only used blue text, any user might miss the fact that text is clickable.

17. What is the difference between a functional requirement and a non-functional requirement?

A functional requirement is a function that the software must *do*, while a non-functional requirement is a constraint on *how*.

18. Define the formula for risk exposure and explain the inputs to the formula.

Exposure = Impact*Probability – impact is the cost associated with a risk, and the probability measures how likely the risk is to materialize.

19. Describe the difference between the *severity* and *priority* of a bug.

Severity is degree of impact to the end user, directly measuring how much it impairs affected users' ability to use the software. Priority is the impact to the project development team and their business, which allows for minor issues that affect many users to be ranked ahead of major impacts that affect only a few users.

20. Describe one of Lehman's Laws of Software Evolution.

See lecture 3 slides 26-28

Section 2 – Five of a Kind

21. Describe five common components of a software requirements specification

See lecture 7 slide 15

22. Describe five roles in a software engineering project team.

See lecture 4 slide 3

23. Describe five cross-cutting concerns that might apply to a software product.

See lecture 13 slides 6-9

24. List five git commands and describe the purpose of each

See lecture 5 slides 23-24

25. Give five examples of elements that might be found in a coding standards document or style guide.

See lecture 15 slide 11 with descriptions as given in lecture 15, Keyfactor guidelines on Brightspace, or guidelines you found for your team projects.

26. Describe five potential outcomes of triaging a bug report.

See lecture 18 slides 16, 19, and 23-27

27. Describe five activities that occur for a software release after the code complete milestone.

See lecture 21b slides 5, 8, 9, 12

28. Describe five artifacts that might be included in a software release other than the main program.

See lecture 21b slide 13

Section 3 – Requirements Engineering and Product Design

Refer to the project description attached to the exam for this section. You should also refer to the source code for part 4 for guidance on what components might be included in your models, but remember that the code is only a *partial* implementation; you will need to consider classes that are not addressed in the code.

29. (10 pts) Develop a class model for the IDE project.

Diagram to be added separately

30. (10 pts) Develop a behavioral model (state diagram) for the IDE project

Diagram to be added separately

Section 4 – Implementation and Testing

Refer to the project description and teammate's pull request contents attachments for this section.

31. (5 pts) Identify a block of code that demonstrates adherence to appropriate coding standards & how

Answers may vary. Editor class has no substantive issues and variable names and indentation are consistent.

32. (10 pts) What is the cyclomatic complexity of each function defined in the pull request? What code quality issues are present here that you would comment on in a code review? (You may mark up the code instead of writing here)

onKeyPress() CC is 17. IDE::Build is 3. workspace::save CC is 2. All others are 1

The top code issue is the amount of functionality in onKeyPress that's only superficially related by the source of the input. It is not a cohesive function and it has a high cyclomatic complexity. It should do minimal processing itself and pass the concern of updating state of objects like editor and workspace to functions in their respective classes. It is likely that the complexity of this function results in bugs due to incorrect branching (e.g. missing "else" statement after the "Enter" key handler will result in subsequent lines also being run). Some minor issues like inconsistent capitalization are present too, many sections aren't adequately commented, and there are no precautions taken with the input to the "system" call, which could result in unsafe operations.

33. (5 pts) Does testRunHelloWorld pass (return true), assuming the build, execution, and output capture functions work as intended? Does it uncover any bugs?

The test does not pass as the provided project fails to build due to a bug in workspace::save, where the editor's lines are concatenated into the output file without newline separators. This makes a great test case because it uncovers that bug, and the test passes one that is fixed.

34 (10 pts) Write one unit test case for the OnKeyPress function that validates a requirement. Describe in general terms how you would write additional tests to achieve adequate code coverage.

```
bool testEditorInsert() {
    editor e = editor{ "", vector<string>(), 0, 0 };
    IDE ide = IDE{ 0, e, workspace() };
    onKeyPress(Keycodes[" "], false, false, ide);
    return ide.editor.lines.size() == 1 && ide.editor.lines[0].starts_with(" ");
}
```

This unit test covers the situation where the IDE is in editor mode and the user types a space into an empty file, then verifies that the editor has the single space. Test cases would also need to be written for other operations in the editor as well as switching modes and running commands, and more would need to be added as the onKeyPress() function implements additional functionality.

35. (10 pts) Refactor the code for the OnKeyPress functionality to improve the code quality and separation of concerns. Does your unit test still pass?

```
void onKeyPress(Keycode key, bool shift, bool ctrl, IDE ide) {
    if (ctrl) { // Used to switch modes in the IDE
        for (int i = 0; i < 5; i++) {
            if (key == Keycodes[to_string(i)]) {
                ide.mode = i;
                return;
            }
        }
        if (key == Keycodes["S"]) { // Ctrl+S=save
            ide.openWorkspace.save(ide.editor);
        }
        // TODO - Ctrl+N new file, etc
    }
    if (ide.mode == 0) { // Typing into the editor
        editor e = ide.editor;
        if (key == Keycodes["Enter"]) {
            e.lines.push_back("");
        }
        if (key == Keycodes["Backspace"]) {
            if (e.cursorColumn == 0) { // Special case for start of line
                e.lines.erase(e.lines.begin() + e.cursorLine);
                return;
            }
            string line = e.lines[e.cursorLine];
            line.erase(line.begin() + e.cursorColumn);
        }
        if (KeycodeIsCharacter(key)) {
            e.lines[e.cursorLine].insert(e.cursorColumn, KeycodeToCharacter(key,
shift));
        }
    }
    if (KeycodeIsFunctionKey(key)) { // One of the commands specified in
requirements
        if (key == Keycodes["F4"]) { // Push
            ide.PushCodeToRemoteRepo();
        }
        if (key == Keycodes["F5"]) { // Debug
            ide.LaunchDebugMode();
        }
        if (key == Keycodes["F6"]) { // Build
            ide.Build();
        }
        if (key == Keycodes["F7"]) { // Run tests
            ide.RunTests();
        }
        if (key == Keycodes["F10"] && ide.mode == 4) { // Step through debug
            ide.ExecuteNextDebugLine();
        }
    }
}
```

Commented [JK1]: Move this into a separate "changeModes" function

Commented [JK2]: Move this into an editor::type() method

Commented [JK3]: Move this to a separate "handleCommand()" function

Section 5 – Project Management and SCM

36. (5 pts) The project management team would like to automate some parts of employee performance evaluation for developers on the IDE project using Lines of Code per day as a metric. Use pseudocode to list the steps a script could take to assist with this task by examining the repository commit history.

Initialize dictionary mapping users to number of lines

Get list of commits in the period to be examined

For each commit in list {

 Get author of the commit

 Get number of lines changed in the commit

 Add number of lines changed to the total for the author

}

Get total number of days in range being considered

Output each user with their total number of lines accumulated across all their commits divided by the number of days under consideration

37. (5 pts) Describe a business risk related to the requirement to store credentials for a repository in the workspace file and the potential impact of that risk.

Credential disclosure from insecure storage could compromise the code stored in the repository that the credentials access, which could impact customers using the IDE project with financial loss that the IDE vendor could be responsible for.

38. (5 pts) A last-minute customer request comes through to add support for breakpoints in debug mode (run-to-next-breakpoint on command, as an alternative to execute-next-step on command). Create a task list for this new requirement.

Assign a function key to the new command and handle the input

Add a list of integers to editor class representing lines with breakpoints, and tools to save/load this info.

Add mechanism to toggle breakpoint on a line (e.g. ctrl+b in editor mode could add one at the cursor)

Add functionality to debug mode to repeatedly execute additional steps without prompting until it detects another breakpoint.

39. (5 pts) Describe a set of manual acceptance tests that could be performed to help validate that the finished product meets the requirements.

Testers should verify that users are able to compose, build, run, and test real projects interactively, and synchronize the code with a repository. Several such complete end-to-end tests should be done by multiple people to ensure that functionality works as expected. The interactive nature of the IDE means that some aspects of holistic usability cannot be adequately assessed with automated testing, so it is important to do a good deal of manual testing of this type before shipping the IDE to customers. Ideally, the development team should switch to using a build of the IDE itself for further development of the project.

Extra Credit

You're given a function "getContents" that takes a directory path as a string and returns a list of strings representing the contents of that directory (empty list if the path doesn't exist or is a file rather than a directory). Write a function in the language of your choice to search the entire filesystem for all files matching a given filename, with the lowest possible cyclomatic complexity.

See lecture 26 slide 17