

EECS 481 W26 Take-Home Exam

Version 1.1

Overview

This is a cumulative, take-home final examination for EECS 481.

The exam is **due at midnight Eastern on April 22nd**. The exam is not timed; you may spend as much time on it as you like. We encourage you to plan for at least twelve (12) hours spread over multiple days.

The exam is open notes, open lectures, open internet, and open lecture slides. You may use free generative AI tools.

You may not use another person: you may not work in pairs, you may not pay another person to answer questions for you, you may not post a question on a forum (like Stack Overflow or Reddit) and ask others to answer it, etc. The exam must reflect your own work using allowed tools and resources.

We may use plagiarism detection software, like Turnitin, to detect shared work. Students often dislike such detection because of perceived false positives; however, this is a necessary tradeoff in an untimed, unsupervised take-home exam. If you are concerned about potential false positives, we encourage you to spend extra time making sure that your answers reflect your own understanding and voice.

Grading on this examination will be quite strict, especially compared to homework reports. You have as much time as you need to complete the exam; we will be holding your submitted answers to a correspondingly high standard.

Submit your exam as a single PDF via Gradescope. You may re-submit as often as you like; we will grade your final submission before the deadline. (Students sometimes mistakenly submit their HWx report in the HWy report slot or similar; please double-check that you are not doing that with the exam since there may not be an easy fix at the end of the semester.)

Google Docs Logistics

The link given to you for the exam includes a special *copy* directive: each student receives an individual copy of the examination as a separate Google Doc.

You **must** use Google Docs to prepare your responses to the questions, and when you submit the exam you must **share** the Google doc with EECS481Staff@gmail.com such that we see the

Version history. Click on the “File” menu and choose “Share” to share the Google doc with the course staff (and no one else!). You can also click on the “File” menu and choose “Version history” to see the history of edits to your exam document -- but there will not be much to see until you start editing the document over time. At the end of the examination there is a field to include the shareable links of all of the Google docs you used.

We know a few students like to write everything on a separate piece of paper and copy it all over at the last minute. Don’t do that for this examination.

If you mistakenly clear your version history or otherwise cannot present the full history to us, you will receive a zero on the examination.

We will use your version history to protect you when resolving any disputes about cheating or unauthorized aid. (We know that some methods of cheating are still possible, even with a thorough version history. This requirement is more about protecting innocent students from false positive allegations. For example, if Turnitin thinks you copied Student XYZ on Question 123, but we can see that you wrote your answer to Question 123 before Student XYZ did by comparing version histories, you are protected.)

Exam Checkpoint

Attempting to complete the final exam within the last few days of the course is unlikely to result in grades that students are happy with. Students often benefit from answering questions on early-semester topics while those early-semester topics are still fresh in their minds. In addition, prose clarity often benefits from writing draft answers, setting them aside, and then coming back to them later for checking and polishing.

To help keep students on track, an **exam checkpoint is due on February 20th**. For the exam checkpoint, you will submit a PDF of your partially-completed exam. We will grade the checkpoint on length but not on content. That is, we will look to see that you have written some amount of text, but we will not look to see if it is right or wrong.

For the checkpoint, you must have answers for any **two (2)** of the questions from Q1 to Q26. You pick which two questions. It doesn’t matter which you pick: even though some are listed as being worth more points for the final grade, any two questions from Q1 to Q26 are fine for the checkpoint. For the checkpoint, we’re just looking to see that you did *some* work. We will **not** grade the exact content of your checkpoint answers to the two problems (i.e., we will not grade if your answers are fully right or wrong). Instead, for the checkpoint, we will just check to see if you have a reasonable amount of text that is addressing the point of the question (i.e., isn’t just random words to fill space). The purpose of the checkpoint is to provide structure to reward students for not putting all of the exam work off until the last day (which is basically guaranteed to go very poorly). On Gradescope you will indicate which two questions you want us to look at. It is fine if you have other questions partially or fully done as well. For the checkpoint, we will only look at the two you indicate.

Exam Versions

We may release updated versions of the exam text (e.g., to fix typos or clarify wording). The version number is listed at the top. You can always download an updated version and copy your answers into that updated version. We will make every effort not to change the meaning of any question, but you are responsible for submitting answers associated with the final posted exam version.

Forum Posts About The Exam

Make any posts about the exam **privately (i.e., visible to the full course staff only)** using the [exam] Piazza tag. If the course staff think your question or its answer may help everyone, we will remove any identifying information and then make the post public.

Course Concepts

Broad **course concepts** include, but are not limited to:

- Software Development Processes
- Uncertainty, Risk, Planning, Cost Models
- Software Metrics
- Validity, Bias, Perverse Incentives
- Testing
- Test Suite Quality Metrics
- Mutation Analysis
- Oracle-Comparator Model
- Test Generation
- Passaround Code Review, Formal Code Inspection
- Dynamic Analyses, Race Conditions
- Pair Programming, Interviews
- Defect Reports, Triage, Severity vs. Priority
- Static Analyses, Dataflow Analyses
- Debugging, Fault Localization, Profiling
- Delta Debugging
- Requirements, Functional vs. Quality
- Requirements Elicitation, Conflicts
- Designs for Maintainability
- Design Patterns
- Software Engineering Cognition
- Productivity
- Automated Program Repair
- AI and Software Engineering

Note that the above list is not exhaustive. First, each broad concept has a number of associated sub-concepts (e.g., Testing includes Unit Testing, Integration Testing, etc.; Design for Maintainability includes methods for Documentation, Change, etc.) that you are responsible for. Second, concepts may be presented or reinforced in the lectures, in the homeworks, and/or in the required readings; you are responsible for all such information. Third, concepts sometimes have multiple relevant names (e.g., Program Synthesis vs. Program Repair, etc.) or are linked or tied throughout the semester (e.g., Code Review and Code Inspection are both manual Static Analyses methods but are introduced before the lecture on automated Static Analyses, etc.); you are responsible for all such connections. Fourth, some concepts are assessed more extensively by other mechanisms (e.g., Homeworks, Reading Quizzes); the exam does not give equal weight to all concepts.

As you work on the exam throughout the semester, you may benefit from focusing on questions related to concepts we have already covered in class or in the assignments and leaving other questions for later.

General Computing Concepts (Subject Matter Expertise Concepts)

Software Engineering projects are interdisciplinary in that they involve using software engineering methodologies and techniques in service of other goals or in other domains. Developing a networked application requires knowledge of software engineering and also knowledge of networking. Developing a scientific computing application requires knowledge of software engineering and also knowledge of numerical methods. And so on.

Our Requirements Elicitation lecture explains that software engineers doing Requirements Elicitation perform Content Analysis to learn about the relevant system domain. You have the opportunity to practice that skill, and to demonstrate your accomplishments, as part of this exam.

Successfully completing this exam will require you to synthesize your knowledge of Software Engineering with knowledge of these **general computing concepts**:

- (Atomic) Transactions [Databases]
- Client-Server Architecture [Computer Networking]
- Cross-Site Scripting [Computer Security]
- (Distributed) Denial of Service Attacks [Computer Security]
- Model-View-Controller Architecture [Web Applications and GUIs]
- Polygon Meshes [Computer Graphics]
- Server Provisioning [Computer Networking]
- Zero-Day Vulnerabilities [Computer Security]

Students are not expected to be familiar with these concepts already (although a student may be familiar with one or two of them from other classes). **You are responsible** for learning about these concepts on your own. Reading up on these concepts on your own and applying your knowledge of them with your knowledge of software engineering is part of this exam and is something you will be graded on. When a question asks you to pick a General Computing Concept, you must pick one from this list.

You may use resources you are comfortable with (e.g., Wikipedia articles, YouTube videos or tutorials, slides or recordings of other classes, Generative AI summaries, etc.) to become familiar with these concepts.

We recommend that you spend at least 30 minutes per concept. While the exam questions below are not “trick questions”, they may not explicitly call out when one of these concepts is critical to a full-credit answer. If you only have a superficial understanding of one of these concepts, you may not realize that combining it with your software engineering knowledge is secretly the key to a high-quality answer. Losing points on a question because you did not realize that a general computing concept was relevant is a frustrating outcome that can be avoided.

Case Study Context

A significant theme in this course is real-world software engineering. The questions in this exam center on a case study of a high-profile software engineering “blunder” associated with hundreds of millions of dollars: Amazon’s ***New World***. We use the bugs, company actions, user actions, and situations surrounding New World to frame questions about software engineering concepts *in context*.

For convenience, the exam questions reference the events as presented in the popular video [New World - Timeline of a Failure](#), using timestamps into that video, but you are welcome (indeed, encouraged) to use other sources. (If you have an SSD letter indicating that you cannot watch or listen to videos, etc., please email the course staff for alternative accommodations.)

If you are watching *without an ad blocker*, I recommend only opening the video once and then navigating to different time stamps. Optionally, you can download a local copy. If you have to watch an advertisement every time you consider the video, the exam will not be a pleasant experience.

Although the case study has a particular context (a networked multiplayer game), the focus of the assessment is not directly on any contextual detail. Just as HW1 requires students to learn and apply details of computer graphics (such as PNG transparency, colors or error handling) to the software engineering activity of testing and HW4 requires students to learn and apply concepts from computer security (such as CVEs) to the software engineering activity of static analysis defect reporting, this exam only involves networked computer gaming as a substrate.

- Students who are not as fluent with software engineering concepts may struggle to apply them in context, but often **misattribute** those difficulties to a lack of familiarity with that context. If you are working on this exam and have a gut reaction that a question is difficult for you because you are not familiar with this particular context (online gaming), our best advice is to review the course material and software engineering concepts (e.g., re-read all of the one-slide summaries on the lecture slides) even if that does not *feel* like the source of your difficulty. Just as students do not need to be Aerospace Engineers to understand the software engineering implications of Lufthansa Flight 2904 scenario discussed in the Requirements and Specifications lecture, and just as students do not need to understand Search Engine Algorithms to appreciate the software engineering implications of “The Google Technical Interview” reading, this exam is not actually “about” this context. It has been constructed to be approachable to all 481 students. Applying software engineering in context is a real life issue (see the discussion of Content Analysis above and in class) and this class intentionally prioritizes realism. Both this Exam and HW6 intentionally require you to apply software engineering concepts to novel domains you may not be familiar with.

If you are uncertain, we recommend that you first watch the entire video, and then come back and read through and address the questions below (following along with the video a second time).

If you find that the timestamps are not working for you, and that clicking on the video links always takes you to the start of the video (perhaps in a small inner window), be sure to *click on the blue title* (“New World - Timeline”) and not on the big picture. Clicking on the big picture is very tempting, but Google Docs sometimes interprets clicking there as permission to start the video from the beginning.

Questions [200 points total]

Complete the questions below in the boxes provided.

The questions often ask you to consider what Amazon “should” have done or could have done, using course concepts, to avoid (or prevent, or find, or find earlier, or fix, or fix more easily, or address more cheaply, etc.) the failures described. That is, although the historical events surrounding this software product’s launch and deployment are already over, the exam is asking you to consider the case study as a hypothetical situation.

When you are asked to include a quote or figure or graph from course material, you must either:

1. Copy the quote text and indicate where it is from (e.g., via a URL)
- or*
2. Include a cropped screenshot and indicate where it is from (e.g., via a URL)

Some questions, especially at the start of the exam, indicate which Course Concept they relate to. For other questions, you may have to figure out the right approach.

The questions contain an intentional aspect of ambiguity. Students may be tempted to make private forum posts asking about what questions means. While the course staff may be in a position to provide rephrasing or alternate wordings, we are not in a position to provide clarifications about what exactly students should write. Similarly, some questions give sentence limits but others do not. We realize that ambiguity can feel uncomfortable. Students are encouraged to approach the examination questions in the same general way that they approach the homework report writeup questions.

Q1. Content Analysis [24 pts]

For each one of the General Computing Concepts above, separately do the following. Cite or link two sources you used to to gather information on that topic. For each citation, copy 1 relevant sentence (e.g., that you believe is indicative about a possible link to software engineering or that taught you something). Then identify a software engineering mistake (at any point in the software development process or lifecycle) that someone who does not understand this concept might be more likely to make. Use at most 4 sentences to identify and explain the software engineering mistake.

For each of the 8 General Computing Concepts, you will have 2 sources, 1 relevant sentence, and one software engineering mistake. [3 pts per concept]

a. Q1-a. Transactions.

Replace this text with your answer to Q1-a for Transactions
(If you type more, the box will expand. Your answer must be in the box to be graded.)

b. Q1-b. Client-Server Architecture.

Replace this text with your answer to Q1-b for Client-Server Architecture

c. Q1-c. Cross-Site Scripting.

Replace this text with your answer to Q1-c for Cross-Site Scripting

d. Q1-d. Denial of Service Attacks.

Replace this text with your answer to Q1-d for Denial of Service Attacks

e. Q1-e. Model-View-Controller Architecture.

Replace this text with your answer to Q1-e for Model-View-Controller Architecture

f. Q1-f. Polygon Meshes.

Replace this text with your answer to Q1-f for Polygon Meshes

g. Q1-g. Server Provisioning.

Replace this text with your answer to Q1-g for Server Provisioning

h. Q1-h. Zero-Day Vulnerabilities.

Replace this text with your answer to Q1-h for Zero-Day Vulnerabilities

Q2. Uncertainty, Risk and Planning [6 pts]

Consider Amazon's outcomes with Breakaway, Crucible, Lord of the Rings, and The Grand Tour Game as described near <https://youtu.be/F3ZMly9YAPA?t=97>.

- a. Support or refute the position that the level of variability Amazon experienced is consistent with the results reported in [Variability and Reproducibility in Software Engineering: A Study of Four Companies that Developed the Same System](#). Use at most eight sentences. [3 pts]

Replace this text with your answer to Q2-a.

- b. Using <https://eecs481.org/lectures/se-02-risk.pdf> as a guide, explain the relationship between innovation and risk and the outcomes of those four products (i.e., what ultimately happened to those four products). Your answer should include details specific to the context shared by those four products. While the products are all different, they may share some similar aspects with respect to innovation, risk and outcomes. Use at most eight sentences. [3 pts]

Replace this text with your answer to Q2-b.

Q3. Requirements Elicitation [6 pts]

Consider the discussion near <https://youtu.be/F3ZMly9YAPA?t=171> about the potential desire for more "player vs. player" (PvP) product content. In that situation, "many" people were asking for more PvP content but Alpha Testers were not displaying behavior showing that they wanted more PvP content.

- a. Identify three different bolded concepts from <https://eecs481.org/lectures/se-14-valid.pdf> that are relevant to this situation. For each, explain how it is linked to this situation, and then identify a quote from [Requirements Elicitation: A Survey of Techniques, Approaches, and Tools](#) that would help in such a situation. Use at most 12 sentences (4 per concept). [Hint: The exact meaning of "player vs. player" is not relevant for answering this question.] [2 pts per concept]

Replace this text with your answer to Q3-a.

Q4. “The Developers Had To Scramble” [4 pts]

- a. Include a quote or figure/graph from the course that best explains the magnitude of the issue described near <https://youtu.be/F3ZMly9YAPA?t=221> . Explain its relevance in 1-2 sentences. [2 pt]

Replace this text with your answer to Q4-a.

- b. Identify a quote or figure/graph from the course that best explains the “theory” for why Amazon’s vast development resources could not be leveraged to resolve this issue. Explain your answer in 1-2 sentences. [2 pt]

Replace this text with your answer to Q4-b.

Q5. “At Launch” [3 pts]

Consider the Tweet wording discussed near <https://youtu.be/F3ZMly9YAPA?t=312> .

- a. Following <https://eecs481.org/lectures/se-14-valid.pdf> , what type of requirements elicitation conflict is at issue? Justify your answer. Use at most four sentences in your answer. (You may have to consider material covered later in the video or in other sources to answer this question.) [3 pts]

Replace this text with your answer to Q5-a.

Q6. “Permenant [sic] Ban” [8 pts]

Consider the user-facing ban message described near <https://youtu.be/F3ZMly9YAPA?t=369> .

- a. Relate its reception here to the example Bug Bounty program detailed in <https://eecs481.org/lectures/se-24-repair.pdf> . Consider the scope of bug bounty programs (e.g., “only” functional correctness vs. quality properties vs. aspects related to designing for maintainability). Use at most four sentences. [4 pts]

Replace this text with your answer to Q6-a.

- b. Relate this issue to the notions of Severity and Priority. Use at most four sentences. [4 pts]

Replace this text with your answer to Q6-b.

Q7. “Groups of Friends” [6 pts]

Consider the friend-related issue described near <https://youtu.be/F3ZMly9YAPA?t=397> .

- a. If we consider this issue to be a bug, when in the software development lifecycle was it most likely introduced? Use at most 2-4 sentences. [2 pts]

Replace this text with your answer to Q7-a.

- b. Choose one of the Course Concept categories listed above (e.g., Testing, Design Patterns, etc.) and argue that Amazon “should” have done more of that concept (or done it more correctly, etc.) to efficiently and effectively avoid this. If there are multiple correct answers, you should pick the best one: if you describe something that is “not false” but is also not the best answer, you will not receive full credit. Use at most 8 sentences. [4 pts]

Replace this text with your answer to Q7-b.

Q8. “Would Instantly Crash” [8 pts]

Consider the public video referenced at <https://youtu.be/F3ZMly9YAPA?t=427> .

- a. Within the main video directly linked above, another video (entitled “Pro Tip: It’s Faster To Quit The Game By Fishing ...”) is briefly shown and referenced. Identify all of the ways in which that “Pro Tip” video is a good defect report, using multiple concepts and/or components from <https://eecs481.org/lectures/se-10-bugreport.pdf> . You may consider material from elsewhere, such as the “Pro Tip” video referenced. Use at most four sentences. [2 pts]

Replace this text with your answer to Q8-a.

- b. Identify all of the elements of a good defect report that are missing from the video referenced. Use at most four sentences. [2 pts]

Replace this text with your answer to Q8-b.

- c. For the crashing bug referenced above, support or refute the claim that fuzz testing would likely have found it. “Should” Amazon have used fuzz testing? Use at most four sentences. [4 pts]

Replace this text with your answer to Q8-c.

Q9. “Manually Reviewed” [4 pts]

Consider the specification in the Tweet referenced at <https://youtu.be/F3ZMly9YAPA?t=493> .

- a. Using the concepts from <https://eecs481.org/lectures/se-13-req.pdf> , which requirement documentation error type is most relevant and why? Justify your answer. Use at most four sentences. [4 pts]

Replace this text with your answer to Q9-a.

Q10. “New Content At An Astounding Rate” [8 pts]

Consider the complaint at <https://youtu.be/F3ZMly9YAPA?t=608> .

- a. Suppose Amazon were to use the metric “in-game square meters of map created” to measure progress. Assess this metric using at least three sub-concepts from Measurement. Which existing software quality metric do you think this metric is most analogous to, and why? Use at most six sentences. [4 pts]

Replace this text with your answer to Q10-a.

- b. Propose an alternative metric with different properties. (Students who are more familiar with networked online games can suggest domain-specific metrics, but it is also possible to answer this question using only notions from the class, the homework, and the general computing concepts above.) Identify two aspects in which your proposed metric does well and one aspect in which your proposed metric still fails. Use at most five sentences. [4 pts]

Replace this text with your answer to Q10-b.

Q11. “Several Hours Long” [8 pts]

Consider the complaints beginning at <https://youtu.be/F3ZMly9YAPA?t=647> . In addition, consider Amazon's elastic cloud compute (EC2) product: we first used EC2 in HW0, but you may want to read about the general problem it is supposed to solve.

- a. Why do you think Amazon's own game suffered from the problem that Amazon's own EC2 product appears designed to solve? Identify the most relevant Course Concept and explain how it is related. Identify the most relevant General Computing Concept and explain how it is related. Use at most ten sentences. [Hint: This is a question with some nuance; you may benefit from considering it carefully or proposing multiple answers. You may benefit from thinking about the scenario in terms of Process decisions Amazon may or should have made.] [8 pts]

Replace this text with your answer to Q11-a.

Q12. “Hair Model” [9 pts]

Consider the bug described near <https://youtu.be/F3ZMly9YAPA?t=746> .

- a. Support or refute the claim that automated testing would likely have identified the bug before deployment. “Should” Amazon have used automated testing for this? Use at most four sentences. [3 pts]

Replace this text with your answer to Q12-a.

- b. Using the Oracle-Comparator Model described in class, what sort of oracle and comparator would be necessary? Use at most four sentences. [3 pts]

Replace this text with your answer to Q12-b.

- c. Consider two hypothetical test inputs: one that involves a character and apparel combination with no visible clipping, and one that involves a character and apparel combination with visible clipping. For each of Statement Coverage, Branch Coverage and Path Coverage, indicate whether the clipping test has higher, lower or the same coverage as the no-clipping test. [Hint: You do not have access to Amazon’s exact source code. This question asks you to reason from first principles and apply your software engineering knowledge to this context.] [3 pts]

Replace this text with your answer to Q12-c.

Q13. “Copy Paste Versions” [8 pts]

Consider the issue described near <https://youtu.be/F3ZMly9YAPA?t=813> .

- a. Support or refute the claim that a comparator specialized to town mesh geometry would have helped address this problem in testing. Use at most four sentences. [4 pts]

Replace this text with your answer to Q13-a.

- b. Support or refute the claim that at least one of the Design Patterns covered in class would have prevented this duplication. “Should” Amazon have used such a Design Pattern? [4 pts]

Replace this text with your answer to Q13-b.

Q14. “Mass Report Them” [10 pts]

Consider the issue described starting near <https://youtu.be/F3ZMly9YAPA?t=851> .

- a. Identify the most relevant General Computing Concept. Identify three properties of that General Computing Concept. Explain how the issue described in the video aligns with that General Computing Concept by drawing explicit parallels between the three

properties you identified and the activity described in the video. Use at most six sentences. [6 pts]

Replace this text with your answer to Q14-a.

- b. Support or refute the claim that Formal Code Inspection would have helped address this issue. "Should" Amazon have used Formal Code Inspection for this? Use at most four sentences. [4 pts]

Replace this text with your answer to Q14-b.

Q15. "Discovered And Reported" [8 pts]

Consider the issue-reporting policy described near <https://youtu.be/F3ZMly9YAPA?t=906>.

- a. Identify the most relevant General Computing Concept and explain how it is related. [2 pts]

Replace this text with your answer to Q15-a.

- b. Identify three software quality (non-functional) properties covered in class. For each one, explain how the issue-reporting system described in the video fails to provide that property. Use at most three sentences per property. [Hint: This question is asking you to treat the issue reporting process, normally viewed as a sub-component or aspect of the software development process, as a product in its own right. What makes a good issue-reporting process?] [6 pts]

Replace this text with your answer to Q15-b.

Q16. "Client" [10 pts]

Consider the client and server description near <https://youtu.be/F3ZMly9YAPA?t=960>. Consider the model-view-controller (MVC) design and Client-Server Architecture General Software Concepts.

- a. The video mentions three problems (aim, currency, health). Identify one and argue why it would still be possible even in a perfectly-implemented MVC architecture. Use at most three sentences. [Hint: The video's narrator may not be a perfectly accurate source about software engineering details.] [3 pts]

Replace this text with your answer to Q16-a.

- b. Identify another one of those three problems and indicate why it would not be possible in a perfectly-implemented MVC architecture. Use at most three sentences. [3 pts]

Replace this text with your answer to Q16-b.

- c. Support or refute the claim that Amazon's Client-Server design is using Delegation, as in the Design for Maintainability course concept. Use at most four sentences. [Hint: Amazon's subsequent explanation may be relevant: <https://youtu.be/F3ZMly9YAPA?t=1701>] [4 pts]

Replace this text with your answer to Q16-c.

Q17. "Window Drag Glitch" [5 pts]

Consider the window-drag issue described around <https://youtu.be/F3ZMly9YAPA?t=989> and also around <https://youtu.be/F3ZMly9YAPA?t=1388> .

- a. Support or refute the claim that the issue is an instance of a Race Condition. A strong answer would describe relevant Race Condition properties and link them to this case study. Use at most six sentences. [Hint: You may need to consult sources beyond the course slides for more information about Race Conditions.] [5 pts]

Replace this text with your answer to Q17-a.

Q18. "Deflation Crisis" [5 pts]

Consider the gold economy issue described near <https://youtu.be/F3ZMly9YAPA?t=1104> .

- a. Support or refute the position that a Dynamic Analysis tracking player gold over time would have identified this issue earlier during development. "Should" Amazon have used a Dynamic Analysis here? Use at most six sentences. [Hint: This is not a question about macro- or micro-economics or finance details. You can consider "money" at the same level of abstraction as in the banking and ATM examples we cover in class.] [5 pts]

Replace this text with your answer to Q18-a.

Q19. "But Does Not Process The Giving Player As Having Lost Gold" [10 pts]

Consider the exploit described near <https://youtu.be/F3ZMly9YAPA?t=1155> .

- a. Identify the General Computing Concept that best serves as the Requirements/Specification for this activity and explain why it is a good fit. Use at most three sentences. [2 pts]

Replace this text with your answer to Q19-a.

- b. Write two formal Functional requirements for aspects of that General Computing Concept that are relevant to this context. Argue that each is Complete, Consistent and Precise. Use at most eight sentences total. [4 pts]

Replace this text with your answer to Q19-b.

Shortly thereafter, Amazon released a patch <https://youtu.be/F3ZMly9YAPA?t=1315> for that issue that introduced other issues.

- c. In Code Inspection and Review, we discussed how human static code analyses might be ordered with respect to various types of dynamic Testing (e.g., unit testing, integration testing, etc.). In AI and Software Engineering we discuss Pushing Directly To Prod. Synthesize your knowledge of those three Course Concepts and propose a software process that Amazon “should” have followed to avoid this issue. Identify two strengths of your proposal and one weakness. Use at most six sentences. [Hint: A later instance of Amazon doing something similar may provide additional evidence: <https://youtu.be/F3ZMly9YAPA?t=1736>] [4 pts]

Replace this text with your answer to Q19-c.

Q20. “Stuck In A Charging Animation” [6 pts]

Consider the bug described near <https://youtu.be/F3ZMly9YAPA?t=1212> .

- a. Choose Static Analysis or Mutation Analysis. Which one “should” Amazon have used to more efficiently detect this problem early? If you chose Static Analysis, indicate what your Dataflow Facts would be, indicate whether the analysis is Forward or Backward, and indicate which of the Analyses from class it is closest to. If you chose Mutation Analysis, indicate your three most important Mutation Operators, and describe how you will choose the Order of your mutations. Use at most ten sentences. [6 pts]

Replace this text with your answer to Q20-a.

Q21. “Text Box” [6 pts]

Consider the bug described near <https://youtu.be/F3ZMly9YAPA?t=1285> .

- a. Choose Fuzz Testing or Dataflow Analysis. Which one “should” Amazon have used to more efficiently detect this problem early? Give two reasons why your choice is appropriate for this context and one reason why the other choice is not appropriate. For the advantages and disadvantages, be specific to this context, do not simply restate definitions. [For example, arguing that Dataflow Analysis is a poor choice because it requires one to consider Dataflow Facts would not be a strong answer, because that is always true of Dataflow Analysis, and is not specific to this “Text Box” issue.] [6 pts]

Replace this text with your answer to Q21-a.

Q22. “Crouch Heal Bug” [3 pts]

Consider the bug described near <https://youtu.be/F3ZMly9YAPA?t=1437> .

- a. A design in which status effects are influenced by animation state would seem to violate notions of Abstraction and/or Modularity. Choose a Course Concept and a specific technique or activity within that Course Concept. Argue that Amazon “should” have used that technique or activity to avoid this outcome. Be specific, linking the general Course Concept to this context. [3 pts]

Replace this text with your answer to Q22-a.

Q23. “Even More Broken” [12 pts]

Consider the bug described near <https://youtu.be/F3ZMly9YAPA?t=1477> .

- a. Identify the General Computing Concept most specifically relevant to, or descriptive of, this issue. Justify your answer. [2 pts]

Replace this text with your answer to Q23-a.

Suppose Amazon developers capture the text at 25:29 (<https://youtu.be/F3ZMly9YAPA?t=1529>), perhaps as part of a game log when another user reports the issue. An experienced developer may recognize the issue from that text alone. A developer less familiar with this General Computing Concept might benefit from using Delta Debugging.

- b. Describe how a developer could use Delta Debugging to assist in understanding the issue. Indicate what the initial set is and indicate (informally) what the Interesting() function would do. [4 pts]

Replace this text with your answer to Q23-b.

“Should” Amazon hire more experienced developers, or “should” Amazon hire less-experienced developers but train them to use tools like Delta Debugging? Synthesizing your understanding of Productivity, Cost Models, and Software Engineering Cognition:

- c. Give one advantage and one disadvantage of hiring more experienced developers. Include a quote or figure/diagram from the course to support your answer. Your answer should be specific to this context. Use at most six sentences. [Hint: For example, simply indicating as a disadvantage that hiring more experienced developers is more expensive would not be a strong answer; it is not specific to this context.] [3 pts]

Replace this text with your answer to Q23-c.

- d. Give one advantage and one disadvantage of training developers. Include a quote or figure/diagram from the course to support your answer. Use at most six sentences. [3 pts]

Replace this text with your answer to Q23-d.

Q24. “Climb Over The Scenery” [6 pts]

Consider the issue described at <https://youtu.be/F3ZMly9YAPA?t=1667> . Identify three different challenges with using Spectrum-Based Fault Localization to localize this bug. Each challenge should be drawn from a different Course Concept. For each challenge, identify the Course Concept it is associated with, and link it to Fault Localization in general and this context in specific. Use at most 2-3 sentences for each challenge.

- a. First challenge. [2 pts]

Replace this text with your answer to Q24-a.

- b. Second challenge. [2 pts]

Replace this text with your answer to Q24-b.

- c. Third challenge. [2 pts]

Replace this text with your answer to Q24-c.

Q25. “They Forgot To Add It” [9 pts]

Consider the gauntlet bug (<https://youtu.be/F3ZMly9YAPA?t=2034>), hatchet bug (<https://youtu.be/F3ZMly9YAPA?t=1708>) and the great axe bug (<https://youtu.be/F3ZMly9YAPA?t=1914>).

Consider the use of a Creational Design Pattern and the use of a Dataflow Analysis to track paired functions (like the open-close file example in class). For each bug (gauntlet, hatchet, great axe), indicate whether Amazon “should” have used a Creational Design Pattern, Dataflow Analysis, Both, or Neither to more efficiently address it. For each bug, include up to two sentences of logical support.

- a. “Gauntlet” bug answer. [3 pts]

Replace this text with your answer to Q25-a.

- b. "Hatchet" bug answer. [3 pts]

Replace this text with your answer to Q25-b.

- c. "Great axe" bug answer. [3 pts]

Replace this text with your answer to Q25-c.

Q26. "Congratulations!" [8 pts]

Consider the staff interaction described at <https://youtu.be/F3ZMly9YAPA?t=2459> .

- a. Design a Behavioral Interview question that would likely give insight into a candidate Community Moderator's suitability for avoiding such blunders. Your question must be at most two sentences long. Your question should be specific to this context but should not be easy to defeat. [Hint: For example, "Tell me about your greatest weakness" is not specific to this situation, and a candidate could answer it honestly without giving any information relevant to this blunder. By contrast, "Would you ever just say 'Congratulations!' to a user complaining about a bug on a public forum?" is very specific, but is also not useful: a candidate who lacks the relevant skills could still just say "No, of course not". If the graders think your Interview Question would not be good, you may lose points even if you believe it is good: think carefully about your answer.] [2 pts]

Replace this text with your answer to Q26-a.

- b. Argue that your question, while not perfect, would likely be helpful for this purpose. Consider False Positives, False Negatives, True Positives, and True Negatives and company goals. Use at most six sentences. [6 pts]

Replace this text with your answer to Q26-b.

Q27. Honor Pledge

Attest that you completed this work following the [UM Honor Code](#).

After each examination, students must write the Honor Pledge in their test books and sign their names under it. The Honor Pledge is as follows:

"I have neither given nor received unauthorized aid on this examination, nor have I concealed any violations of the Honor Code."

Replace this text with your answer to Q27-a (i.e., type the honor pledge here)

What is your Google Doc link for your copy of this examination? You must share it with EECS481Staff@gmail.com (and only the course staff: no other students). We will use your document edit history to protect you and help resolve disputes about cheating or the use of unauthorized aid. If you used multiple Google docs (e.g., perhaps you worked on Version 1 of the exam, but when Version 2 came out, you started on a fresh copy and pasted your answers over) then include shared links for *all* of those documents. You will receive a zero on the exam if you are unable to share a full version history.

Replace this text with your answer to Q27-b (i.e., your Google doc link for the exam you worked on, shared so EECS481Staff@gmail.com can see the edit history). If you used multiple Google docs, you must include each one.

Q28. Extra Credit [variable points]

- a. Name a course staff member (Derek, Hanchi, Jesse, Priscila, Sathvika, or Wes) and list something that person did that helped you learn. (If no course staff member helped you, indicate something another student in 481 did to help you learn or call out one of the course materials (assignment, reading, etc.) and explain why or how it helped you learn.)

Replace this text with your answer to Q28-a.

- b. What is your favorite part of the class? Alternatively, what is something that we should make sure to retain for future semesters?

Replace this text with your answer to Q28-b.

- c. What is your least favorite part of the class? Alternatively, what is something that we should improve or change for future semesters?

Replace this text with your answer to Q28-c.

- d. Choose any optional reading (it must say **Optional**: before it on the Lectures page) or any post tagged **long-instructor-post** on Piazza that was not about course logistics. Identify it and demonstrate to us that you have read it critically by relating it to your own personal experience, going beyond a Generative AI Summary. (We may not give credit if your answer looks like an impersonal AI summary. The standard here is quite high.)

Replace this text with your answer to Q28-d.

- e. How hard did you find this exam? How much time did you spend on this exam? Which Generative AI tool(s) and version(s) did you use on this exam, if any? For which parts of

the exam did you find them (or speculate that they would be) helpful or unhelpful?
(Remember, free GenAI is allowed, so this is not cheating. This is to help us improve the course, not to get you in trouble.) If you used any other sources that you are uncertain about, you can cite them here.

Replace this text with your answer to Q28-e.