Software Design & Documentation Main Exam

Fall 2026

This exam is made up of three questions. You must pass the exam to pass the course. Grammar, spelling, and citing references count significantly toward your exam grade. Please write in full essay form, rather than just as if you are answering specific questions.

The test is designed to challenge your written communication and critical thinking skills as well as your understanding of the course's material. The exam is open book and open notes. You may use any resource at your disposal to complete this exam (except other students of SD&D). The content **must** be your own work and express your own ideas. **Please include a detailed list of all resources used.** Your answers should reflect your own understanding of the material. Any students caught plagiarizing answers will receive a 0 for this exam, and be referred to the Dean of Students.

The entire Main Exam is due in Submitty
Friday, November 13th 2026 at 11:59pm.

Prior to final submission of your exam, you are required to submit your First Cut. This includes a description of your chosen FOSS, your proposed features and requisite user stories and user scenarios.

The Main Exam First Cut is due in Submitty
Friday, October 16th 2026 at 11:59pm.

We also require that you make at least one appointment with The Writing Lab in the Student Success Center (https://successcenter.rpi.edu/writing-lab) and follow through with the lab's recommendations. Please provide proof of any reviews (indicating that you completed the review), along with the reviewer's name, and date/time of the review.

In addition to the CGCD signed document(s), please provide a cover page of your exam that you have signed yourself indicating that all of the exam work is original.

Additional requirements:

- You must cite references using a standard method.
- You cannot use Wikipedia as a reference.
- You must use at least four published books, industry magazines, videos, and web

articles. Keep in mind you can use any books recommended in class.

As further preparation for the exam, please review the notes and presentation slides from Chapters 4 – 6 from the **Head First OOA&D**: HeadFirstOOAD_Ch05.ppt (or from Google Books). You may use the set of these presentations as one reference since they originated from one book.

Question A (50 points)

Select an open source (FOSS) application or framework **created in an Object-Oriented language** and choose two essential and non-trivial features you would want to add or extend. Make sure that the new or extended features provide a creative solution to satisfy your particular user goals. You will need to design your extension such that it relies on abstraction, low coupling, high cohesion, data encapsulation, inheritance, and polymorphism at the very least. Feature additions that are too simple in nature to express your basic knowledge of OO development will result in a much lower grade. For a framework, you may need to reference a sample application to explain your changes.

Provide a complete overview of the software and why your features would add value for a particular subset of end-users (*Main Exam First Cut):

- * Describe the FOSS. Include a brief overview of its uses and a few features that distinguish this application from similar programs. Don't forget links and specific references to similar applications.
- * Describe your proposed features. Provide good detail that fully explains each feature and how it's useful to users.
- * Provide a complete list of User Stories for each feature. Use the form "As a <type of user>, I want <some goal> so that <some reason>."
- * Provide at least two (2) User Scenarios. (one for each feature) that paints a word picture of a persona achieving their goal using the software. Include a very brief description of the selected personas.
- Create CRC cards that describe the responsibilities and collaborators of each new class needed for each feature. Provide cohesion and coupling scores based on any scale
- Create a class diagram for each feature (with public and private visibility indicators) incorporating the new features. The class diagram should include aspects of the original application design so the new classes are portrayed in context.
- Create at least one sequence diagram for each feature showing the interaction between the several classes that collaborate to achieve the feature.
- Explain your diagrams and choices related to abstraction, low coupling, high cohesion, data encapsulation, inheritance, and polymorphism.
- Describe a third possible feature which would not require any modification to your design. This feature should demonstrate how your design is easily extensible, rather than just another thing to add to the application. Explain how and why your design is flexible enough to allow for this extension. (You do NOT need to create diagrams, user

stories, user scenarios, etc., for this feature.)
- Explain how your design passes the SOLID principles as discussed in class (if it does not, please redesign). Please also include a brief explanation of all five principles.
- Apply the **Ease of Change** test from the *Head First OOA&D* reference above. Please create the appropriate questions and answers for your application.

All diagrams need to be in correct standard UML. Make sure to include the URLs to the original application.

Question B (25 points)

Create a project plan for the features you described in Question A using one of the Agile methodologies discussed in class (XP, Scrum, Crystal, Kanban, etc.).

- Provide a thorough overview of the specific Agile methodology as an introduction to your project plan. Include all pertinent information about the methodology including guiding concepts, values, roles, artifacts, deliverables, phases, ceremonies, communication and other activities, etc.
- Describe some examples of artifacts that are specific to your chosen methodology. Ideally, provide a real example of one of the artifacts.
- Develop a schedule for development, assuming you are planning for a team of two to four (2-4) developers. Include number of developers used, quantity and length of iterations as well as what specifically will be delivered at the end of each one.

Question C (25 points)

Please write a full comprehensive essay that addresses the following topic:

Based on Frederick Brooks' discussion of the lack of calendar time problem in Chapter 2 of *The Mythical Man Month* (located on the course website in the Resources section (https://sites.google.com/site/rpisdd/resources), what are five questions you can ask the team to determine the likelihood that the project will be completed and deployed on time and within spec? Please justify and explain the value of asking each question and discuss what conclusions you would draw from different answers both positive and negative. Finally, outline actions you would have the team take to remediate any identified issues arising from the questions.

Example questions might include:

- What percent of your iteration tasks are you completing by the end of each iteration?
- What are your current risks? How have you prepared for them?
- How much time have you allocated for testing? How do you justify that amount of time and are you on track with that?

Grading rubric

FirstCutRubric.pdf
MainExamRubric.pdf