

Name _____

CPSC:480 Software Engineering Midterm

Do not open exam until instructed to do so!

Each question in section 1 is worth 2 points.

Each question in section 2 is 10 pts (2/item).

Each question in section 3 is 10 pts

One sentence per description is adequate.

100 points total.

Section 1 – Short answer

1. Explain the difference between perfective and adaptive maintenance.
2. Give an example of a non-functional requirement for a class registration app.
3. Describe one of Lehman's Laws of Software Evolution.
4. Describe the concept and benefit of a DevOps process in software engineering.
5. Explain the concept of encapsulation for software modules.

6. Describe an early software program that ran on the UNIVAC or predecessors.

7. Describe one function of project management software.

8. Describe the role of a product owner in a software development organization.

9. Describe a reason a software engineer might use command-line development.

10. Describe one benefit of formal specification and verification.

Section 2 – Software development process

11. Draw the prototyping model of software development with five phases.

12. Describe five classic software engineering domains

13. Describe five common components of a software requirements specification.

14. Describe five principles of effective software design.

Section 3 – Modeling and Design

In this section, use the following description and requirements of a software system:

Many bars and other venues hire piano players to entertain guests by playing customers' song requests on a keyboard. A common problem is for the keyboard player to receive requests for songs they do not know and for which they cannot easily find the sheet music specifying how to play it. A bar owner believes he can use software to overcome this limitation and increase the number of guests – and therefore his revenue – by providing a better music experience.

He knows that the keyboard at his bar has three pedals that aren't used in most pieces, and that it supports a MIDI (i.e. USB) interface to a computer that can register when a pedal is pressed. He aims to use these features along with a computer screen above the keyboard to allow the keyboard player to select a song from the current list of requests, then display and turn pages in a digital copy of the sheet music for that song.

Furthermore, the owner employs several piano players of different skill levels – “beginner”, “intermediate”, and “advanced” – and needs to automatically select the version of the sheet music for a given song that matches the skill of the current player. He has a database of songs with the title, original artist, and each version of the sheet music. He also wants to track the total number of songs each player has played in order to pay players per song. Finally, for the case when no song is requested, he wants to maintain a list of the overall most played songs to select from as a default fallback.

The software does not need to maintain a current set of requests, only the sheet music for each song the player might have requested. However, the software must allow the piano player to interact with the software entirely through use of the three pedals “P₁”, “P₂”, & “P₃”. You may assume each piece of sheet music has four pages. The functional requirements for this software are as follows:

1. As a player, I want to view a list of available artists, select one, view a list of their songs, and select one to play.
2. As a player, I want to be able to turn pages forward and back in the sheet music displayed and be able to close it and return to the menu at any time.
3. As a bar owner, I want to be able to select the name of the current piano player and their skill level for tracking and automatic difficulty selection.
4. As a bar owner, I want to track how many songs each player plays and what the overall most played songs of all time are.

15. Develop a class model for this system.

16. Develop a possible behavioral model, depicted as a state diagram.

17a. Develop a scenario model for req #1, depicted as a swimlane diagram.

17b. Develop a list of tasks for implementation of requirement #1.

18. Describe how you could decompose this system into at least two distinct modules. Briefly explain what object types you would have in each module and how they would interact.

Extra Credit

Your teammate forked a branch “test” after commit 07a45e0 and merged it back to “main”. However, in this branch “config.txt” was accidentally changed to contain invalid content. Write a sequence of git commands to restore the version of config.txt from before the branch, but keep all other changes from test branch.

Grading

Question	Points	Grade
1	02	
2	02	
3	02	
4	02	
5	02	
6	02	
7	02	
8	02	
9	02	
10	02	
11	10	
12	10	
13	10	
14	10	
15	10	
16	10	
17a	05	
17b	05	
18	10	
Extra Credit	05	
Total	100	