

Design Patterns covered in this exam:

Decorator, Template Method, Composite, Command, Chain of Responsibility, Factory Method, and Observer. Model-View-Controller

There won't be any specific questions on Adapter, Singleton, State, and Strategy, but you might be asked to compare a design pattern to others that can include the first four, or you might be asked to identify patterns which work well with the current list. You would be able to choose from the first four. In other words, you do not have permission to forget about the first four for this exam, but study these other 7 with more vigor.

1.) Design Pattern Program

Look at the code labelled Practice Exam 2 Question 1 in Moodle. Use this program to answer the different parts for question 1. This is a variation of the Vacation program that we discussed in class.

- a.) Which Design Pattern is used in this program? Just provide the design pattern.
- b.) Justify your choice – explain why this program uses the design pattern that you provide.
This should be an essay that explains the design pattern in general AND shows how it is implemented and used in the program. (~350 to ~450 words).
- c.) Explain how this program uses the design pattern to perform its computation.

2.) Design Pattern Program 2

Look at the code labelled Practice Exam 2 Question 2 in Moodle. Use this program to answer the different parts for question 2.

- a.) Which Design Pattern is used in this program? Just provide the design pattern.
- b.) Justify your choice – explain why this program uses the design pattern that you provide.
This should be an essay that explains the design pattern in general AND shows how it is implemented and used in the program. (~350 to ~450 words).
- c.) Explain how this program uses the design pattern to perform its computation.

3.) Design Pattern Concepts

Describe three benefits of using Design Patterns in software development.

4.) Variations on Patterns

Explain the difference between each pair of items given. Give an example of each.

- a.) PUSH vs. PULL style Observer patterns
- b.) Creating chains vs. existing chains for the Chain of Responsibility pattern
- c.) Command Design Pattern vs. Executing a method call
- d.) Template Method implemented with: Inheritance vs. Composition

5.) Design Patterns in Context

For the program scenario described below, select a design pattern from the ones in the list above that, in your opinion, can best be applied. (NO CODE is to be written for this question)

- 1.) Identify the design pattern that you would choose.
- 2.) Explain how you would apply it to this scenario – be specific.

You wish to write program to control an interactive exhibit at a museum. The program has 81 monitors arranged in a 9x9 grid mounted on the wall. On a display table in front of the wall are 81 keypads with the numbers 1-9. These keypads are also arranged in a 9x9 grid on the table. When someone presses the '4' on the upper, leftmost keypad, the number 4 appears on upper, leftmost monitor on the wall. You guessed it, this is a giant, interactive Sudoku puzzle. There is a button labelled START which when pressed loads a puzzle into the display by placing the initial, permanent numbers into the grid.

Your program determines the background color of the monitors. Sudoku puzzles start with some numbers already in the puzzle. These are permanent and do not change. These numbers should be displayed with a BLUE background. When the monitor is blank, the background should be WHITE. If there is a number on the monitor and that number is NOT in conflict with any other numbers in the same row, column, or sub-grid, then the background should be GREEN. If the displayed number is in conflict with any other number in the same row, column, or sub-grid, then the background color should be RED.

For example, the puzzle begins with 15 permanent numbers displayed, all with a BLUE background these will not change, regardless of any numbers typed on the keyboard. All the rest of the monitors are empty and have a WHITE background. As museum visitors enter numbers in the keypads, the numbers appear on the monitors with a GREEN background (none of these conflict with any others). Someone presses a keypad to put a '7' in the lower right corner, however, someone else has already put a '7' in the lower, left hand corner, so these are in conflict. Both monitors should switch their backgrounds to RED. To fix this, someone changes the lower, left hand corner to a '2'. Both the '2' and the other '7' now have a GREEN background as they are no longer in conflict with each other or any other numbers. When the puzzle is solved – every location has a number from 1-9 and there are no conflicts – then all the backgrounds turn GOLD and stay that way until START button is pressed again.

6.) Decorator and the Composite Patterns

Both the Decorator and the Composite patterns store information in an organized hierarchy.

Create two NEW applications of these patterns (One for Decorator and one for Composite). Explain why the patterns apply well to your example. Further explain why the other pattern would NOT be good for each application. (ie. Why the application that you describe for the Decorator pattern would not be as well served by the Composite pattern, and vice versa).

7.)

8.) Studying Code

Study the Puzzle Solver program posted to Moodle PracticeExam2Question8. What Design Pattern(s) are in this program? What Design Pattern(s) might help this program?

9.) Studying Code

Study the Puzzle Solver program posted to Moodle PracticeExam2Question9. What Design Pattern(s) are in this program? What Design Pattern(s) might help this program?

IMPORTANT!!!

Study this program (NY Times Spelling Bee) well, the code will be on the actual exam!!

10.) Studying Code

In Moodle under PracticeExam2Question10 is a program that implements the NY Times Spelling Bee game. Study this program and bring general questions about this program to class on Wednesday.