

15-214: Principles of Software Construction

Fall 2014 Sample Midterm II
Charlie Garrod and Jonathan Aldrich

Name: _____

Andrew ID: _____

Recitation section or TA name: _____

Instructions:

- Make sure that your exam has 10 pages (not including this cover sheet) and is not missing any sheets. Then write your full name, **Andrew ID**, and recitation section on this page (and all the others if you want to be safe).
- Write your answers in the space provided below the problem. Clearly indicate your answers.
- The exam has 6 questions with a maximum score of 76 points. The point value of each problem is indicated. Ideally you should take approximately one minute per point; pace yourself accordingly.
- This exam is **CLOSED BOOK** and **CLOSED NOTES**. You may not use a calculator, laptop or any other electronic or wireless device.
- If our questions are unclear, please make and state your assumptions.
- Good luck!

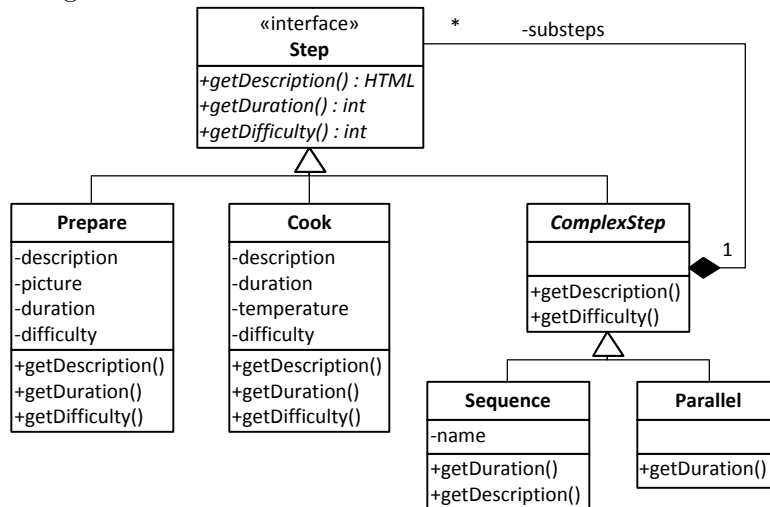
Question	Points	Score
Evaluating design alternatives	14	
Frameworks	13	
Designing a data library	6	
Multiple choice questions	9	
Short questions	10	
A flight reservation system	24	
Total:	76	

Question 1: Evaluating design alternatives (14 points)

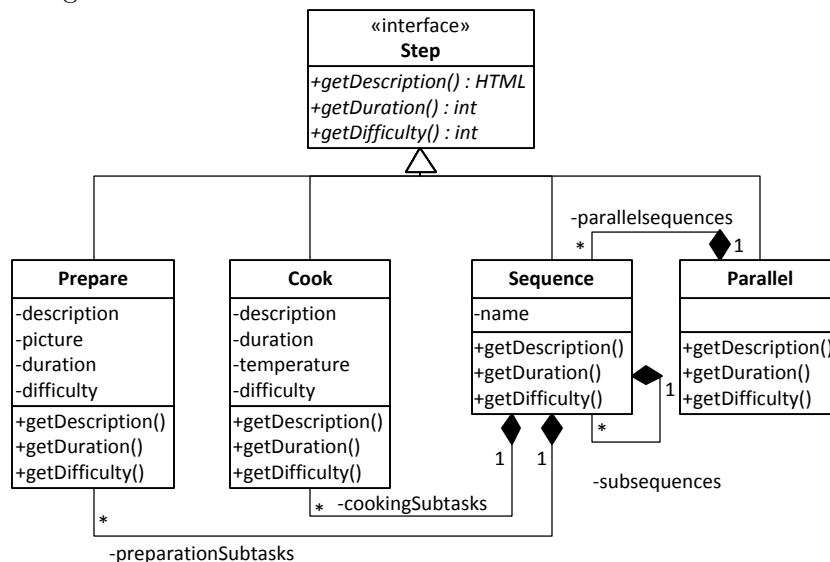
Recipes typically consist of multiple steps, with each step having a duration, a difficulty from 1 (easy) to 10 (hard), and a description that can be rendered as an HTML page (possibly with pictures). A typical recipe describes several steps in a sequence, where the duration is the sum of the durations of all steps and the difficulty is the difficulty of the most difficult step. Sometimes recipes describe steps that should be executed in parallel, such as “Boil the water” and “Cut the vegetables”.

Consider two alternative designs in terms of the design goals and strategies discussed in class:

Design 1:



Design 2:



(a) (2 points) Name the design patterns, if any, in Design 1.

(b) (2 points) Name the design patterns, if any, in Design 2.

(c) (2 points) Which design is better in terms of coupling? Circle your answer and justify your answer in 1-2 sentences.

Design 1

Design 2

Neither

(d) (2 points) Which design is better in terms of cohesion? Circle your answer and justify your answer in 1-2 sentences.

Design 1

Design 2

Neither

(e) (2 points) Which design is better for change? Circle your answer and justify your answer in 1-2 sentences.

Design 1

Design 2

Neither

(f) (2 points) Which design is better for reuse? Circle your answer and justify your answer in 1-2 sentences.

Design 1

Design 2

Neither

(g) (2 points) Which design is better for robustness? Circle your answer and justify your answer in 1-2 sentences.

Design 1

Design 2

Neither

Question 2: Frameworks (13 points)

- (a) (6 points) From the following scenario given an example each what functionality the company could reuse through a framework or through a library. Briefly *justify* why a framework or library is the appropriate choice for that functionality.

A company specializes in developing puzzle games for the Android mobile platform. After their first five games they realize that the games share many commonalities and that the develops could re-use much code from their previous games. Common features, for example, include high score, multi-player support, 2D graphics rendering, and animations.

- i. Example where a library would be a good solution (and justification):

- ii. Example where a framework would be a good solution (and justification):

- (b) (4 points) Suppose the company wants to create a library to connect the game from the previous scenario to a micro-payment system to create a virtual currency for in-game purchases. The company is unsure, however, what functionality and dependencies to include in their library.
- i. What problems might occur if the library they create is too small? Give an example within this scenario.
 - ii. What problems might occur if the library they create is too large? Give an example within this scenario.
- (c) (3 points) After the developers have built a framework to achieve re-use for many commonalities in their games, they later realize they forgot an important method common to many of their games. Explain why it might be difficult to add this feature to their framework.

Question 3: Designing a data library (6 points)

Two designers, Pat and Drew, are discussing plans to implement a data library. The library will store data in a stable, standard format and will allow clients to manipulate the data using one of several algorithms, to be chosen at run-time by the client. Some clients may want to define their own algorithms.

Pat says that the library should define algorithms within the data storage class, and it should provide a unique string for each algorithm the library defines. The client passes in the string for the algorithm it wants to use, and the library case-analyzes the string to pick the algorithm. Pat advocates this approach because it groups the algorithms with the data they use, hiding the data representation better. Plus, the data storage class is an information expert.

Drew says the library should define a data storage class that provides public accessors for all data needed by the algorithms. Then the library should define an interface for invoking the algorithms, and one class for each algorithm implementation. Drew argues that although this does not hide the data representation as well, it hides the precise set of algorithms from the data storage class, allowing clients to define new algorithms more easily.

- (a) (4 points) Given the description of the problem, which design is better? Justify your answer.

- (b) (2 points) One of the designers is proposing the use of a design pattern. What is the pattern's name, which designer is proposing it, and how is the design pattern being used in the proposed design?

Question 4: Multiple choice questions (9 points)

Be careful: For each of the parts below, you will receive 1 point for a correct answer, 0 points for leaving the question blank, and -1 points for an incorrect answer. The expected value of guessing the answer is zero.

Circle the best answer for each of the statements below. You do not need to justify your answers.

- (a) (1 point) A domain model describes real-world concepts of the problem.

True

False

- (b) (1 point) A domain model describes key implementation classes of the domain (e.g. card games).

True

False

- (c) (1 point) A domain model facilitates communication between developers and customers.

True

False

- (d) (1 point) The domain model is part of the _____.

Solution space

Problem space

- (e) (1 point) To save time, a domain model can be reverse engineered from the implementation.

True

False

- (f) (1 point) A domain model changes as developers refactor their code.

True

False

- (g) (1 point) Design patterns describe reusable implementation snippets for common problems.

True

False

- (h) (1 point) A key goal of design patterns is to raise the level of abstraction in communicating design decisions.

True

False

- (i) (1 point) Design patterns are _____.

Invented

Discovered

Question 5: Short questions (10 points)

- (a) (4 points) In Homework 4 (Scrabble) we required you to separate the core logic of your game from its GUI. Describe two design rationales (design goals or design strategies, not design patterns!) to support this separation and (for each rationale) how an implementation benefits from this separation.

i.

ii.

- (b) (6 points) Name three design patterns that are used in the Swing framework. For each pattern describe the design goals/strategies and benefits of this choice over alternatives.

i.

ii.

iii.

Question 6: A flight reservation system (24 points)

For this problem consider the following description of a flight reservation system.

The system stores a set of flights, each with a source and destination city, flight number, time of departure, time of arrival, and capacity. The system also stores a set of flight reservations, each of which identifies a seat on a flight for a passenger on a particular date. When a reservation is created, the passenger must be charged for the cost of the flight. When a reservation is changed, the passenger must be charged an additional change fee.

Group reservations can be made for multiple passengers. Any operation that can be done on a single reservation can also be done on a group reservation. For example, changing a group reservation's flight will move all passengers in the group from one flight to another.

A ticket agent sometimes assists passengers with making and changing reservations. The ticket agent can change a reservation without charging a fee (for example, if there is a family emergency). The system does not need to track which agent made a reservation, nor does it need to track the rationale for changing reservations without a fee.

The flight reservation system calculates the cost of a flight based on multiple criteria: the base flight cost, baggage and special seat fees, taxes as a percentage, and various possible discounts (which may be fixed, a percentage, or some more complicated scheme). An agent making a reservation can specify which discounts and fees apply to that reservation. New kinds of discounts and fees may be added in the future. Multiple fees and discounts may apply to a single flight.

(a) (4 points) Suppose a flight has a set of reservations, and a group reservation is a set of individual reservations. Consider the following two possible uses of the composite design pattern:

1. Making a flight a composite of the reservations it contains.
2. Making a group reservation a composite of the individual reservations within it.

Is the composite design pattern an appropriate solution for both situations, neither situation, or just one situation? Justify your answer.

(b) (4 points) Which design pattern should be used to implement pricing for the flight reservation system? Why is that design pattern most appropriate here?

(c) (6 points) Draw an object diagram (as UML class diagram) for the pattern you chose in part (b) applied to the pricing system.

- (d) (4 points) Sometimes a reservation must be changed due to bad weather or other circumstances. When this occurs, passengers must be notified of the change, the employee scheduling system must be notified to update its schedules—and in the future other notifications might need to occur, too.

Which design pattern can best accommodate this? Justify your answer.

- (e) (2 points) Name one concept from the problem description that might appear in the domain model, but is unlikely to be in the source code.

- (f) (2 points) Name a class or interface you would create in the implementation that would not likely appear in a domain model.

- (g) (2 points) The system should be able to report the duration of a flight. Which of the following classes (which are part of our flight reservation system design) should calculate this information: class `ReservationSystem`, class `Flight`, or class `Reservation`? Briefly justify your answer by appealing to a design goal or strategy.