

17-214: Principles of Software Construction
Spring 2021 Sample Midterm Exam 2

Name: _____

Andrew ID: _____

Recitation section or TA name: _____

Instructions:

- Make sure that your exam has 18 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.
- Write your answers in the space provided below the problem. Clearly indicate your answers.
- This sample exam has 6 questions with a maximum score of 166 points. The point value of each problem is indicated. We planned this exam to take approximately one minute per point; you may pace yourself accordingly. This sample exam is substantially longer than an actual exam, to provide questions with a greater breadth of content and type than would otherwise be possible. The actual midterm will have under 60 points.
- If our questions are unclear, please make and state your assumptions.
- Good luck!

Question	Points	Score
Selected shorts	14	
Winter is coming	58	
Evaluating design alternatives	12	
Frameworks	6	
Short questions	14	
Plane and simple	62	
Total:	166	

Question 1: Selected shorts (14 points)

- (a) (3 points) What is the principal difference between a specification-based test suite and a structural-based test suite?
- (b) (3 points) Describe why the observer pattern is an important pattern in framework design. Explicitly refer to the design goals and design principles it achieves in that context.
- (c) (8 points) For each GUI feature below, which design pattern does the feature best demonstrate? (it is not necessary to know the specifics of the GUI framework)
- i. The `JScrollPane` adds a horizontal and/or vertical scrollbar to an arbitrary other GUI component.
 - ii. When a `JLabel` is clicked, it may process the event for the mouse click or propagate the event to its parent component, which may process the event for the mouse click or propagate the event to its parent component, and so on.
 - iii. A `JPanel` can contain three `JPanels` (and/or other GUI components), which can contain thirty-nine `JPanels` (and/or other GUI components), and so on.
 - iv. Swing layout managers enable containers to delegate the size and position of their internal components to one of many layout manager implementations.

Question 2: Winter is coming (58 points)

In this problem, you will design a snowplow tracking service that allows residents of Pittsburgh to receive status updates for particular city blocks. Specifically:

- A city block, exemplified at right, consists of a street name (e.g., Fifth Avenue), a neighborhood (e.g., Oakland), and a contiguous range of street numbers (e.g., 4400–4500, exclusive).
- A resident may register for an account, specifying a user name, email address and/or phone number, and a preferred contact method.
- A resident may subscribe to and unsubscribe from updates for a city block. A resident receives a status update as an SMS message or email, based on their preferred contact method. Residents can be subscribed to status updates for multiple city blocks (e.g., their home and workplace).
- A snowplow driver can publish a status update for a city block. An update is associated with a single city block and contains an arbitrary message (e.g., *The block has been plowed*).



We will not grade any work on this page. You may use this page and the last pages of the exam for scratch work.

- (a) Based on the description above, create a domain model for the snowplow tracking system. Use standard UML where possible; you may explain any non-standard notation you use. Avoid modeling concepts that are not part of the problem description, even if the concepts are part of your personal knowledge of the domain.

- (b) Create a system sequence diagram for the snowplow tracking system. Use standard UML notation where possible; you may explain any non-standard notation you use.

- (c) Suppose a snowplow driver publishes the message *Plowing has been scheduled for 11:05AM* for 4400-4500 5th Ave. in Oakland. Create an interaction diagram that describes the object-level interactions for this scenario, using a UML sequence diagram and standard UML notation where possible; you may explain any non-standard notation you use. Your interaction diagram should describe any interactions for the publication itself as well as any interactions for notifications triggered by the publication.

- (d) Create an object model for the snowplow tracking system using a UML class diagram and standard UML notation where possible; you may explain any non-standard notation you use. Your solution will be evaluated in part by its achievement of reasonable design goals, even if those design goals are not explicitly stated in the problem description.

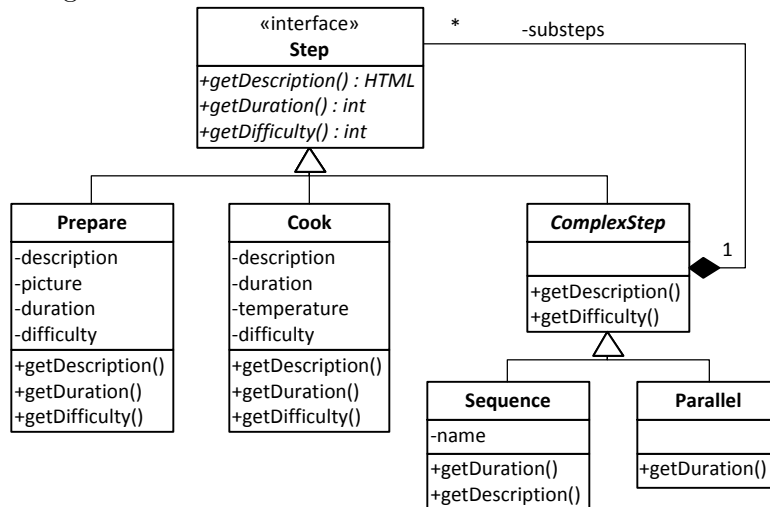
- (e) Name all design patterns in your solution. For each design pattern you use, explain why its use is appropriate in your design.
- (f) In your design, is your representation of a city block mutable or immutable? Justify this design choice, explicitly referring to the design goals and design principles that guided your decision.
- (g) In your design, what object is responsible for tracking subscriptions for a city block? Why did you assign this responsibility to that object, as opposed to alternatives? Justify this design choice, explicitly referring to the design goals and design principles that guided your decision.

Question 3: Evaluating design alternatives (12 points)

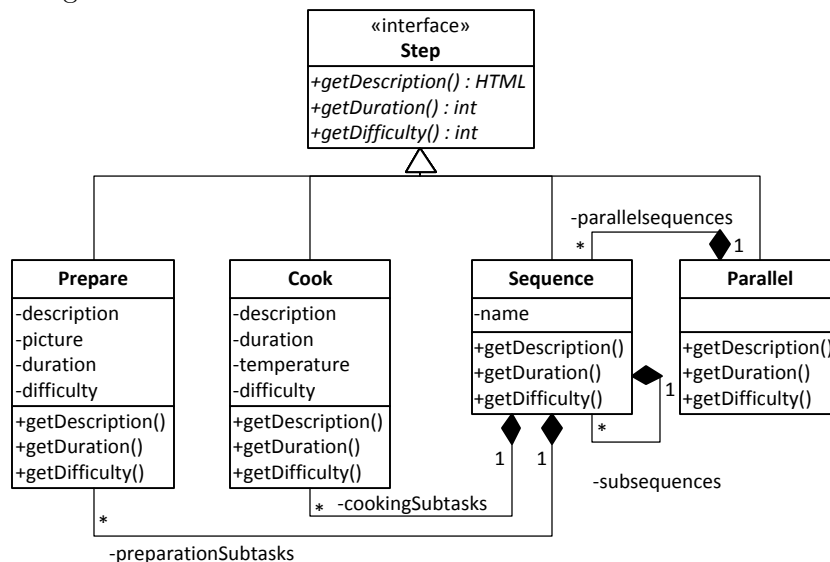
Recipes typically consist of multiple steps, with each step having a duration, a difficulty from 1 (easy) to 10 (hard), and a description that can be rendered as an HTML page (possibly with pictures). A typical recipe describes several steps in a sequence, where the duration is the sum of the durations 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

Question 4: Frameworks (6 points)

- (a) (6 points) From the following scenario give an example of functionality for which a framework or a library is a better mechanism of reuse. Briefly *justify* why a framework or library is the appropriate choice for that functionality.

A company specializes in 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 reuse much code from their previous games. Common features, for example, include high score, multi-player support, 2D graphics rendering, and animations.

- ii. Functionality for which a framework is better (and justification):

Question 5: Short questions (14 points)

Answers to multiple choice questions without a justification will not receive any credit.

- (a) (6 points) The strategy pattern is as general as the template method pattern. In other words, although the template method pattern might better solve a design problem, the strategy pattern could always be used instead of the template method pattern any time the template method pattern solves the problem.

Agree

Disagree

Justify your answer:

- (b) (4 points) Describe two reasons for decoupling a GUI from the core implementation of a system.

Reason #1:

Reason #2:

- (c) (4 points) As part of a commercial application, a black-box framework is usually preferable to a white-box framework.

Agree

Disagree

Justify your answer:

Question 6: Plane and simple (62 points)

In this problem you will design a crowdsourced flight-tracking system that allows users to publish and subscribe to updates about current airline flights. Specifically:

- A flight consists of a departure airport, arrival airport, date, and flight code. A flight code represents both an airline and a flight number, e.g. United Airlines flight 123 has the flight code UA0123. Some flight codes are reused (with different airports) during a day.
- The flight-tracking system supports different kinds of flight status updates: updates for a flight's actual departure time, updates for a flight's estimated arrival time, and updates for a flight's actual arrival time. All flight status updates include a flight and a time, regardless of the type of update.
- A user may register for an account, specifying a user name, email address and/or phone number, and a preferred contact method.
- A user may subscribe to and unsubscribe from updates about a flight. A user receives a flight status update as an SMS message or email, based on their preferred contact method.
- A user may publish a status update about a flight.

- (a) Based on the description above, create a domain model for the flight-tracking system. Use standard UML where possible; you may explain any non-standard notation you use. Avoid modeling concepts that are not part of the problem description, even if the concepts are part of your personal knowledge of the domain.

- (b) Create a system sequence diagram for the flight-tracking system. Use standard UML notation where possible; you may explain any non-standard notation you use.

- (c) Suppose a user publishes a status update for flight UA0123 (DEN to PIT) for November 2nd, 2020, with an estimated arrival time of 10:30 p.m. Create an interaction diagram that describes the object-level interactions for this scenario, using a UML sequence diagram and standard UML notation where possible; you may explain any non-standard notation you use.

- (d) Create an object model for the flight-tracking system using a UML class diagram and standard UML notation where possible; you may explain any non-standard notation you use. Your solution will be evaluated in part by its achievement of reasonable design goals, even if those design goals are not explicitly stated in the problem description.

- (e) Name all design patterns in your solution. For each design pattern you use, explain why its use is appropriate in your design.
- (f) In your design, is your representation of a flight mutable or immutable? Justify this design choice, explicitly referring to the design goals and design principles that guided your decision.
- (g) In your design, what object is responsible for tracking subscriptions for a flight? Why did you assign this responsibility to that object, as opposed to alternatives? Justify this design choice, explicitly referring to the design goals and design principles that guided your decision.