Name: Sample Solutions Student #: soln

Engineering 9874
Software Design & Specification
Mid-Term Test

Dr. D. K. Peters

February 15, 2008

Instructions: Answer all questions. Write your answers to all questions in the space provided on this paper.

This is a closed book test, no textbooks, notes, calculators, electronic translators, or other aides are permitted.

Write your name and student number in the space provided on this sheet. Write your student number only in the space provided on other sheets of this paper. Please ensure that your answer book is clearly identified as indicated on the front cover and that you follow the rules listed there.

If you find a question to be ambiguous or feel that you must make assumptions in order to complete your answer, please clearly state those assumptions on your paper.

Q1	/ 10
Q2	/ 9
Q3	/ 14
Q4	/ 17
Total	/ 50

1. [10 points] In the following table enter "T" (true) or "F" (false), as appropriate, for each statement.

Statement	**T/F**
A "traditional" software development process is better suited to projects with large teams and stable, well defined, requirements.	*T*
In incremental development, integration is done late in the life-cycle.	*F*
Stepwise refinement focuses on the key nouns in the problem description.	*F*
The pre-condition for a method in a derived class can be weaker than for the same method in the base class.	*T*
It is the duty of the calling code to ensure that the post-conditions for the methods being called are true.	*F*
Dynamic aspects of a system design can be represented in class diagrams.	*F*
Abstraction should be avoided in software design.	*F*
Aggregation and composition can both be said "is a part of".	*T*
The principle of information hiding suggests that a class interface should be less complicated than its implementation.	*T*
A pattern will often be a design that can be encoded in a class and used as it is for a wide variety of applications.	*F*

2. [9 points] Give brief definitions of each of the following terms as they were used in this course:

   a) Class
   A specification for objects. Will identifiy attributes and methods that the objects will have.

   b) Design Pattern
   A named problem-solution pair that can be applied in new contexts.

   c) (Component) Interface
   The set of assumptions that other components can make about a component. Includes syntax and semantics.

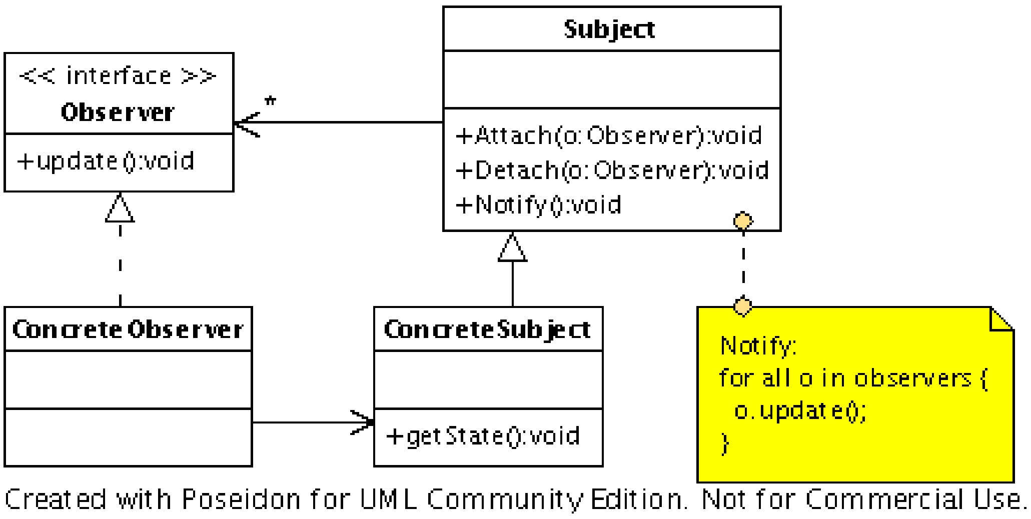
3. [14 points]

   a) [2 points] What are the essential components of a design pattern description?

      - *Name*
      - *Problem*
      - *Solution*
      - *Consequences*

   b) [2 points] The java classes java.awt.event.ActionEvent and java.awt.event.ActionListener are a good example of what pattern? Explain.
      Command.
      ActionListener is the command, ActionEvent is the Concrete Command. Menu items or toolbars are the invokers.

c) [5 points] The observer pattern was used frequently in this course. Sketch a UML class diagram that indicates the key participants and their relationships and briefly describe the roles of the participants.



Subject *Knows observers.*

- *Any number of observers may observe a subject.*
- *Provides and interface for attaching and detaching observers.*

Observer *(interface) defines updating interface for objects that should be notified of changes.*

ConcreteSubject *Stores state (model). Notifies observers of changes.*

ConcreteObserver

- *Maintains reference to ConcreteSubject.*
- *Implements Obsever update interface to keep its state consistent with subject.*

d) [5 points] One of the key challenges in the observer pattern is managing the information flow in response to a change in state. Briefly discuss the issues and alternatives with respect to this.

Update triggering *Whose responsibility is it to call update?*

- *state-setting methods in subject — may lead to too many updates.*
- *clients — error prone.*

Update protocols *How is the information about he change communicated to the observer?*

- *Push model — subject sends observers detailed information about the change. Subject needs to know more about observers.*
- *Pull model — subject sends minimal information, observer goes and gets it. May be less efficient.*

Specifying changes of interest explicitly *Observers register as being interested in specific kinds of changes.*

Change manager *Encapsulates particularly complex update semantics (example of* Mediator *pattern).*

4. [17 points] Consider a software system to allow two players using different computers on the internet to play the game of "Othello", as described below.

   Game Rules for "Othello"

   The playing space consists of a 8 by 8 board of possible positions, which is presented to the players in a graphical display. At the start of the game the board is laid out as illustrated in Figure 1(a). Two players, randomly designated "White" and "Black", play the game by alternately taking turns selecting (by mouse click) an unmarked position in the game space on which to place a marker (a white disc for "White" or a black disk for "Black"). A marker can only be placed such that it becomes an endpoint of at least one straight line (horizontal, vertical or diagonal) terminated by another marker of the same colour and containing at least one marker of the opposite colour (i.e., a sandwich) as illustrated in Figure 1(b). For every such line formed by placing the marker, all of the opposite coloured markers in the sandwich are changed to the same colour as the newly laid marker (see Figure 1(c)). If a player cannot place a marker then she must skip a turn (pass). If a player can place a marker then she must do so. Figure 2 shows other sample moves. The game ends when neither player can place a marker. The winner is the player with the most markers of their colour on the board.

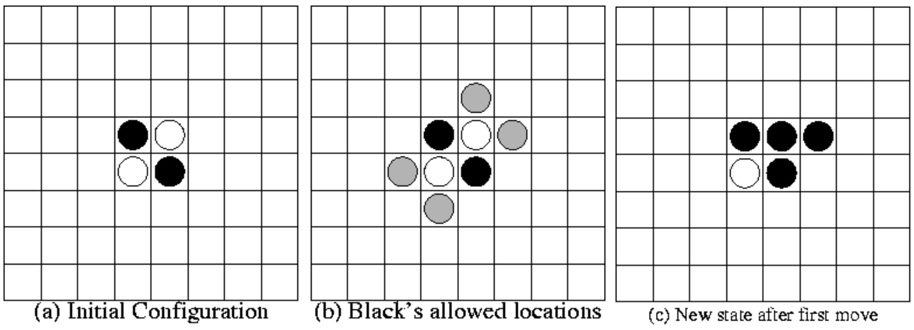

(a) Initial Configuration (b) Black's allowed locations (c) New state after first move

Figure 1: Game Layout

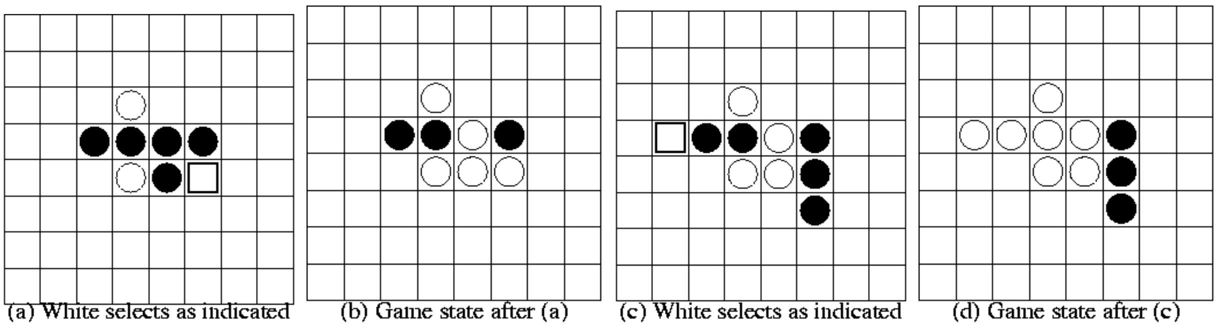

(a) White selects as indicated (b) Game state after (a) (c) White selects as indicated (d) Game state after (c)

Figure 2: Sample Moves

(Questions are on the following pages.)

a) [12 points] The Model-View-Controller pattern can be obviously applied to this system. List the responsibilities of each of the key participants in that pattern as applied to this design.

b) [5 points] Draw a sequence diagram to illustrate the interactions between the system components when a user clicks to place a piece on the board.