Washington University **CSE132. Computer Science II**

Midterm Exam

Kenneth J. Goldman March 8. 2006

Name: _____

Student ID Number: _____

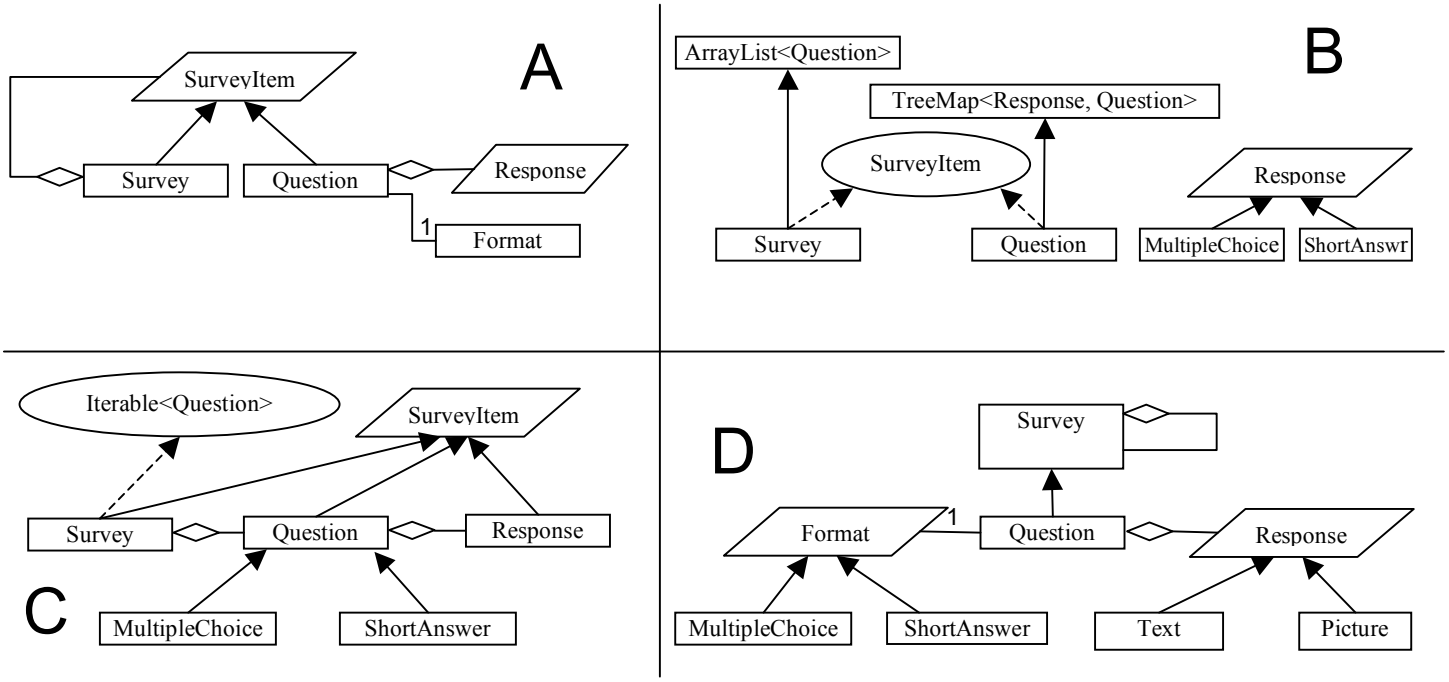
Signature: _____

Directions:

- This exam is open book and open notes. No electronic devices are permitted.
- Please check that you have *pages 1 through 4*.
- Do your own work. No discussion or collaboration with other students is permitted.
- If a question seems ambiguous, write an explanation of how you interpreted the question.
- The exam is 83 minutes long (11:37am – 1:00pm). Be sure to pace yourself.
- When you finish: *If fewer than 10 minutes remain, please do not turn in your exam early, since getting up may disturb other students who are trying to finish.*

Section	Score	Possible
Class Hierarchy Design		24
Persistence		16
User Interfaces		10
Total		50

Class Hierarchy Design (24 points)



Directions: Indicate whether each of the above designs has the property listed in the table. The first line has been done as an example. Each correct entry is ½ point.

Property	**A**	**B**	**C**	**D**
0. The question and survey classes have some shared code.	*YES*	*NO*	*NO*	*YES*
1. Uses the composite pattern, at least to allow surveys to contain questions, which in turn may contain responses.				
2. Surveys may contain other surveys.				
3. Questions may contain other questions.				
4. The implementer of the Survey class would *not* need to provide methods to reorder questions.				
5. If responses do not implement Comparable, a comparator would be required to order the responses.				
6. A survey question can be changed from being a multiple choice question to being a short answer question without replacing the question object itself.				
7. A given question can have a mixture of short answer and multiple choice responses.				
8. The design would directly support writing the following code **for (Question q : survey) System.out.println(q);**				
9. The implementation would need to override a method in order to prevent linking a response to a question that is not in the survey.				
10. PropertyChangeSupport functionality to add and remove listeners for publish/subscribe notification (only about changes in surveys and questions) could be implemented in one place.				
11. Questions are permitted to contain surveys.				
12. The Survey class must have a public method to get the question associated with a particular response, although it could throw an exception.				

Persistence (16 points)

Consider the following ways (labeled A, B, and C) that a user of a survey package might save and load files. Assume that **f** refers to a File object and **s** refers to an instance of a Survey class that implements *Serializable*. Assume that all variables are of the proper type. Also, assume exceptions are handled in a surrounding block.

	Save	**Load**

A	x = new FileOutputStream(f);
	y = new ObjectOutputStream(x);
	z = new BufferedOutputStream(y);
	y.writeObject(s);
	x.close();
		x = new FileInputStream(f);
		y = new ObjectInputStream(x);
		z = new BufferedInputStream(y);
		s = (Survey) y.readObject();
		x.close();

B	s.save(f);	s = Survey.load(f);

C	out = new SurveyOutputStream(new FileOutputStream(f));
	out.writeSurvey(s);
	out.close();
		in = new SurveyInputStream(new FileInputStream(f));
		s = in.readSurvey();
		in.close();

Directions: (9 points) Indicate whether each of the above designs has the property listed in the table. For each entry, indicate **YES**, **NO**, or **CBD** (cannot be determined). Each entry is ½ point.

Property	**A**	**B**	**C**
11. To support this design, the survey package would need to provide at least one static method.			
12. The code takes advantage of buffering for efficiency.			
13. Executing this code would correctly save the survey.			
14. The design is a natural extension of the java.io framework.			
15. The user code (above) would *not* need to be modified if the survey package designer decided to use XML encoding instead of serialization.			
16. The user of the package could modify the code, without changing the survey package, to save two surveys s1 and s2 in the same file.			

17. (3 points) Briefly describe a simple modification for A, B, or C that would result in a more efficient implementation.

18. (4 points) Provide a complete class definition for the SurveyInputStream use in C.

User Interfaces (10 points)

Assume that the class Scribble displays a GeneralPath, which consists of a sequence of line segments. Assume that Scribble also displays a JButton, which is returned by its getButton() method.

```
public class Controller extends MouseAdapter implements MouseMotionListener {
        GeneralPath model;
        Scribble view;

        public Controller(Scribble view) {
                this.model = view.getSketch();
                this.view = view;
                view.addMouseListener(this);
                view.addMouseMotionListener(this);
                view.getButton().setAction(new AbstractAction("clear") {
                        public void actionPerformed(ActionEvent ae) {
                                model.reset();
                                view.repaint();
                        }
                });
        }
        public void mousePressed(MouseEvent me) {
                model.moveTo(me.getX(), me.getY());
        }
        public void mouseDragged(MouseEvent me) {
                model.lineTo(me.getX(), me.getY());
                view.repaint();
        }
        public void mouseMoved(MouseEvent arg0) {
        }
}
```

Directions: Consider the code above and answer "true" or "false" for each of the following.

19. _____ It would be better to use buttons to manipulate the sketch because direct manipulation is error prone and unnatural for the user.
20. _____ The mouseMoved method could be omitted since it doesn't do anything.
21. _____ To enable showing more than one view of the same sketch, the Scribble class should accept a GeneralPath as a parameter to either a contructor or a method.
22. _____ If multiple views were shown of the same sketch, then if the user drew in one of the views, the others would update continuously and automatically, as is typical for the model/view/controller paradigm.
23. _____ If the line "view.addMouseListener(this)" were removed, then all of the user's lines would be connected together.
24. _____ If the line "view.addMouseMotionListener(this)" were removed, then the cursor would not move if the user dragged the mouse.
25. _____ If the controller and the view are in different packages, the getSketch() method of the view should be declared **protected** so that the controller can call it.
26. _____ If the Controller class were a member class of the view class, then the variables "model" and "view" would need to be declared **final.**
27. _____ The action for the clear button is an instance of an anonymous class.
28. _____ Setting the clear action in the constructor won't work because the controller is not added as a listener to the button, as is required in the publish/subscribe pattern.