UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE *of* INFORMATICS *and* ELECTRONICS

Department of Computer Science
and Information Systems

End-of-Semester Assessment Paper

Academic Year:	**2005/06**	Semester:	**Semester 1**
Module Title:	**Software Architecture**	Module Code:	**CS4135**
Duration of Exam:	**2½ Hours**	Percent of Total Marks:	**70**
Lecturer(s):	**D. Conroy**	Paper marked out of :	**100**

Instructions to Candidates:

- Answer any four questions.
- All questions carry equal marks.
- Where asked to produce implementation code strict adherence to any particular syntax is not required. However, the code should bear a strong resemblance to Java.
- Where asked to provide *some* code it is important that the code provided fully illustrates your understanding.

Q1 Answer ALL parts. Total marks awarded for this question: 25.

a) Explain the terms ‘component’, ‘container’ and ‘nested containers’ in terms of Swing GUI programming.

2 marks.

b) What is the signature (including parameters) of the method prescribed by the ActionListener interface in Java?

2 marks.

c) Explain using some Java code how the author of a class implementing the ActionListener interface can determine the identity of the object that implicitly invoked it.

4 marks.

d) What is a modifier in the context of a Java GUI event? Describe how to find whether or not a particular modifier has been applied to an event.

4 marks.

e) Consider a Swing GUI for a simple calculator as in Figure 1. Which layout manager would you consider suitable for the top-level pane (e.g. JPanel)? Which layout manager would you consider for the constituent panel (e.g. collection of JButtons)?

Write a Java class CalculatorPanel that extends the Swing JPanel. Show in the code how you would instantiate the layout managers and attach them to the nested containers necessary to provide the calculator interface.

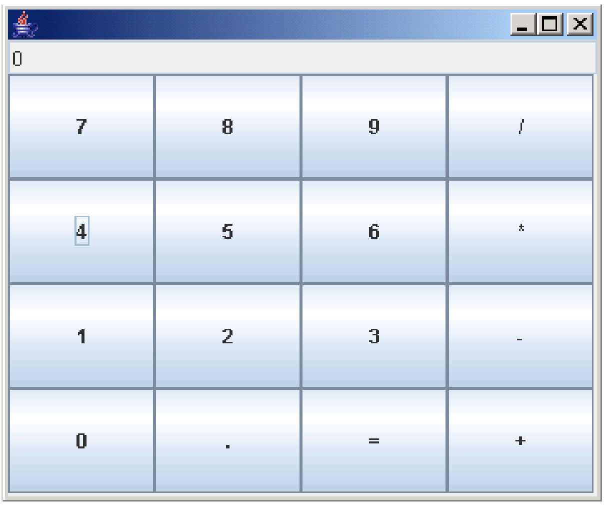

Figure 1.

10 marks.

Explain the meaning of the extra two parameters (5, 5) provided to the GridLayout constructor in the instantiation of panelLayout (but not in the instantiation of boxLayout) below.

3 marks.

```
GridLayout panelLayout = new GridLayout(numRows,numCols,5,5);
GridLayout boxLayout = new GridLayout(numRows,numCols);
```

Q2 Answer ALL parts. Total marks awarded for this question: 25.

a) Briefly describe (one line each) the ten quality attributes of a software architecture identified by Clements, Kazman and Klein.

10 marks.

b) Explain the importance of benchmarks in the context of evaluating the quality of software architectures.

2 marks.

c) Explain the difference between an architectural view and an architectural viewpoint as defined by the IEEE 1471-2000 standard.
or

Provide an example of an architectural viewpoint and suggest two users of the viewpoint.

2 marks.

d) Identify Kruchten's '4+1' views of a system.

5 marks.

e) For each of the '4' views of a system suggest an aspect of the architecture that it might be used to reason about.

4 marks.

f) How does the '+1' view differ fundamentally from the others?

2 marks.

Q3 Answer ALL parts. Total marks awarded for this question: 25.

a) What do you understand by the term delegation? Explain one advantage and one disadvantage of applying delegation.

4 marks.

b)

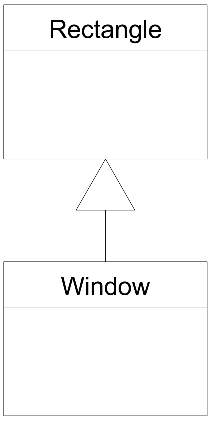

Figure 2.

In a GUI system it is necessary to be able to retrieve the height and width of a window. One approach to providing this would be to implement the window as a subclass of a rectangle class that already provides this functionality, as shown in Figure 2. Explain whether or not you consider this to be an appropriate approach and why. Suggest an alternative approach. Provide some code to illustrate your alternative approach.

4 marks.

c)

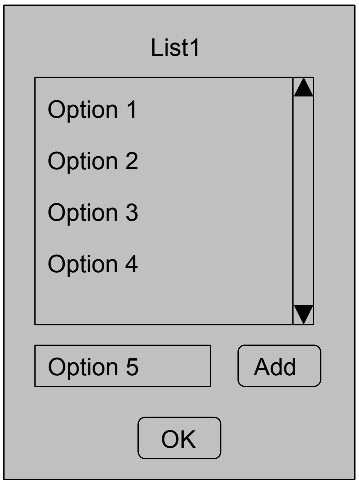

Figure 3.

A dialog in an application GUI (Figure 3) has the following requirement:
When the user clicks on the 'add' button the contents of the text box are appended to the list 'List1'

Suggest a pattern to manage the interaction between the widgets in the dialog box. Draw a class diagram (including method signatures) to illustrate the structure of the pattern in this example.

5 marks.

d) For each of the participant classes of the pattern chosen in part (c) identify and briefly describe the purpose of one of its methods crucial to the implementation of the pattern.

6 marks.

e) The behaviour of the 'add' button must change as follows:
The content of the textbox must be converted to uppercase (use `String.toUpperCase()`) before it is appended to the list.

Explain which participant you would change in order to make that behaviour change. Briefly illustrate (with some code) the change you would make to that participant.

6 marks.

Q4 Answer ALL parts. Total marks awarded for this question: 25.

a) Briefly explain the Implicit Invocation architectural style. Provide an example of how it might be used, a rough diagram will suffice. Give another name for the Implicit Invocation architectural style.

4 marks.

b) Explain why in the java.util package Sun provide an Observer interface to be implemented by observers while the Observable class is extended by subjects (observables).

4 marks.

c) You are required to write your own subject (observable) and observer classes without implementing java.util.Observer or extending java.util.Observable. Call them MyObserver and MyObservable. In that context:

Provide code for your own ‘addObserver()’ and ‘notifyObservers()’ methods ensuring that arguments can be passed to instances of ‘MyObserver’ when they are invoked.

Provide code for your own ‘update()’ method and explain the purpose of its parameters.

10 marks.

d) Explain the difference between ‘push’ and ‘pull’ versions of the Observer pattern. Provide some code that could be added to your MyObservable and MyObserver classes to illustrate the ‘pull’ model.

4 marks.

e) Explain where the Observer pattern could be used in an MVC architecture.

3 marks.

Q5 Answer ALL parts. Total marks awarded for this question: 25.

a) Explain what you understand by the terms ‘abstraction’, ‘encapsulation’. Explain how, in an architecture style of your choice, encapsulation helps in the creation of abstractions.

5 marks.

b) What scope can methods and data have in Java? What do you consider to be a module in Java and why?

3 marks.

c) What is the difference between an ‘open’ and ‘closed’ layered architecture as described by Bennett, McRobb and Farmer?

Provide a brief description, from your experience, of either an open or closed layered architecture.

Explain one advantage offered by a closed layered architecture and one advantage offered by an open layered architecture.

5 marks.

d) What is a native method in Java? Describe one advantage and one disadvantage of using native methods in a Java application.

Does the use of native methods in Java applications results in an open or closed layered architecture for the system? Explain your answer using a simple diagram.

5 marks.

e) Briefly describe the components of the blackboard architectural pattern as described by Garlan and Shaw and describe their interaction.

4 marks.

f) The ‘batch sequential’ style is described by Garlan and Shaw as a degenerate case of which architectural style? How does it differ from the more general form?

3 marks.

Q6 Answer ALL parts. Total marks awarded for this question: 25.

a) Briefly explain the purpose of four components from the RMI architecture. Describe the relationships between those components using diagrams.

6 marks.

b) Unlike current versions of RMI, some distributed systems employ ‘skeletons’. What is a skeleton?

2 marks.

c) Explain one similarity and one difference between Buschmann’s Broker Architecture and the RMI architecture.

3 marks.

d) Explain what is meant by ‘marshalling’. Why is it necessary to marshall parameters in distributed systems?

What interface from the java.io package should a class implement if an instance of it is to be passed between remote objects?

4 marks.

e) Explain the purpose of the following methods from the java.rmi.registry.Register interface named below. Identify and describe the parameters that they take:

bind
list
lookup
unbind

8 marks.

f) Would you consider the Java RMI approach to be suitable for high performance applications? Why?

2 marks.