Ollscoil na hÉireann, Gaillimh
National University of Ireland, Galway

GX 1521

Semester II Examinations, 2003/2004

Exam Code(s)	4IF1
Exam(s)	Fourth Year Information Technology
Module Code(s)	CT432
Module(s)	Distributed Systems
Paper No.	1
External Examiner(s)	Prof. P. Nixon
Internal Examiner(s)	Prof. G. Lyons
	Dr. D. Chambers

Instructions: Answer any 5 questions.
All questions will be marked equally.

Duration	3hrs
No. of Answer Books	1
No. of Pages	5
Department(s)	Information Technology

1.a: Describe briefly the typical architecture for a Distributed Operating System that uses the *processor pool model*. What types of servers are normally used to support this model? 4 MARKS

b: Outline the main differences between a *two-tier* and a *three-tier* Client-Server architecture. Include in your description the main limitations of using a *two-tier* approach and how using a *three-tier* approach can overcome some of the potential problems. 8 MARKS

c: Explain the semantics of a typical synchronous *Remote Procedure Call* operation. Describe briefly the purpose of the following components, as used in the Microsoft RPC environment:

- IDL Compiler
- Runtime Library and Header Files
- RPC Name Service
- RPC Endpoint Mapper 8 MARKS

2.a: Explain briefly the main differences between Java Servlets and Java Server Pages (JSP). In what circumstances do you think it's better to use JSP scripts instead of Servlets? 4 MARKS

b: Explain, using a suitable code example (e.g. a Shopping Cart), the operation of the Session Tracking mechanism available in the Java Servlet API. How would you support session tracking for users that access a servlet with a browser that does not support cookies, or that is set up to reject cookies? 8 MARKS

c: Consider the following JSP script:

```
<%@page language="java" contentType="text/html" %>
<html> <body>
<jsp:useBean id="test" class="MessageBean">
<jsp:setProperty name="test" property="*" />
The following message was set on the bean:
<ul><li>Message: <jsp:getProperty name="test" property="message" /> </ul>
</body>
</html>
```

Explain the operation of the JSP code shown. In particular, describe the operation of the various JSP action and scripting elements used. Also, write the Java Code for the MessageBean class, this can be a simple implementation that allows a message to be stored in and retrieved from the bean object.

8 MARKS

3.a: Describe how incoming requests are processed in a CORBA Server using the the *Portable Object Adapter*. Show, using a suitable example, how a server application could create a *Child POA* to use a different set of Policies from the default policies used by the Root POA. 8 MARKS

b: Using the example of a simple Employee Directory Service, outline the steps required to implement this as a CORBA based application. The following design guidelines will apply:

- It should allow any user to view all other entries.
- The service should allow users to modify their own entries.
- Managers will have extra privileges that allow them modify / delete access to the entries for employees in their department.
- Possible list of interfaces include directory, user, manager, department.
- To keep the task simpler, nesting of departments (orgchart type functionality) is not required!

Include in your answer the *IDL* file definitions and the steps required to complete the application development. Also show how a simple client program might use or interact with the server. Full source code for the implementation classes **is not** required. 12 MARKS

4.a: Describe briefly the operation of CORBA filters / interceptors. What are they typically used for? 4 MARKS

b: Discuss the various policies that affect the design of distributed load balancing systems. Consider the example of adding load balancing capabilities to a *Unix Shell*, which algorithm do you think would work best in this case? How would varying the load exchange period affect the results? 8 MARKS

c: Outline the design of the Secure Sockets Layer (SSL) Handshake Protocol. In particular, explain how the following issues are addressed within this protocol:

- Client and Server Authentication.
- Secure sharing of session keys. 8 MARKS

5.a: Web services represent an evolution and convergence of a number of important areas of technology and business. Describe briefly these technology areas and explain how Web Services builds on previous capabilities. Include in your explanation an overview of the main enabling technologies used to provide Web Services. 8 MARKS

b: Using the example of a simple Currency Converter Service, outline the steps required to implement this service as an EJB based application. The following minimum rules should be adhered to:

- The service will be used to allow clients to interact with a Currency Converter available on an EJB based server. It can be used by client software to convert between the Euro and other common currencies.
- The design of the system should be based on a single session bean called *EuroConverter*. The methods of the *EuroConverter* bean should facilitate the functionality outlined.

Include in your answer the source code for the EJB interfaces required and the source code for a simple standalone client that interacts with a *EuroConverter* bean. Full source code for the implementation class is **not** required, but the answer should show the Java definition for this class. Outline the steps that would be required to complete the application development.

12 MARKS

6. You have been asked to develop a commercial online bookstore using J2EE based technologies. The bookstore architecture and design should be able to support different types of client browsers and should use a three-tier application model i.e. a client tier to support different clients, a middle tier that implements the application business logic and an information tier to persist the application state. Based on these requirements:

   a. Describe the top-level application architecture. Identify the technologies that will be used and explain the role each of these technologies plays in the overall system architecture. 8 MARKS

   b. Identify the various Session and Entity EJBs that will be needed. Explain the functionality that will be provided by each EJB (source code for the EJBs is NOT required). 7 MARKS

   c. Explain how you could generate an appropriate web interface for different types of clients e.g. standard HTTP web browsers and WAP enabled phones. 5 MARKS

7. Using Java Remote Method Invocation, outline the design for an Internet based questionnaire / survey system. The server allows users to get a list of questionnaires available from the system and to download a questionnaire that can then be completed locally, using a suitable client application. The completed questionnaire can then be submitted back to the server for verification and analysis. The following interfaces / classes should be provided:

   - *SurveyServer* - this (remote) interface should provide a number of methods as follows: a method to download an array (of *QuestionnaireSummary* objects) containing a summary of the questionnaires available on the server; a method to download a full questionnaire; and a method to submit a completed questionnaire.
   - *Questionnaire* – this (serializable) interface is used to access a particular questionnaire. A questionnaire is made up of multiple questions and each question can either present a multiple choice option to the user or allow the user to enter a text based answer. The interface should provide methods for the retrieval of information about the questionnaire, and the retrieval / answering of questions. It should also have a method to output the selected answer to each question - the answer provided to a question can then be changed, if desired, prior to submission of the questionnaire.
   - *QuestionnaireSummary* - this (serializable) class is used to encapsulate summary information about a questionnaire. The summary information should include the name of the questionnaire, its purpose and the number of completed questionnaire that have already been submitted to the server.

   The design of the system should make it possible for new Questionnaire implementation classes to be easily added to the system in the future, making the system very flexible. The design should use Java RMI and Object Serialisation to download and then submit objects that implement the Questionnaire interface i.e. these objects are passed by value from the server to the client and then back again to the server. Full implementation classes (for the SurveyServer and Questionnaire interfaces) are not required but the answer should include the full source code for the Java interfaces and class described above. Also include the mainline server code to initialise the server and show how a simple client program might use or interact with the server. 20 MARKS