

DUBLIN INSTITUTE OF TECHNOLOGY
KEVIN STREET, DUBLIN 8

BSC IN APPLIED SCIENCES

YEAR 4

SUMMER EXAMINATIONS 2003

COMPUTER NETWORKS AND DISTRIBUTED SYSTEMS

MR. CIARÁN O'LEARY

DR. BRENDAN O'SHEA

MR. J. W. McCAGUE

Answer **FOUR** questions, **TWO** from each section.

Section A

1. (a) “One of the main advantages of Java as a language for distributed systems is that the same code runs on multiple platforms.”
- (i) Discuss briefly how Java facilitates the same code to run on multiple platforms, and identify why this is important for mobile code.
 - (ii) “One of the few exceptions to the *write once – run anywhere* Java philosophy is in its implementation of concurrency.”
Discuss.

(9 marks)

- (b) (i) Using a Producer-Consumer example with the aid of a diagram, show *why* it is necessary to control access to shared data from multiple threads in a Java application.
- (ii) Using the Producer-Consumer example from part (i) above, describe in detail how Java’s use of *locks* and *monitors* manages access to shared data.

You must use sample code in your answer, and clearly describe the function of each Java keyword you use.

(16 marks)

2. (a) “*Building a distributed system is difficult*”.
Identify and discuss five fundamental reasons why this is generally accepted, contrasting an application distributed throughout the Internet with an application built to run on a single machine.

(10 marks)

- (b) (i) Explain what is meant by *transparency*, in the context of distributed systems.
- (ii) Identify five different types of *transparency* that are used in designing distributed systems. Use examples to explain each one in detail.
- (iii) Using your knowledge of the Java language, APIs, tools and platform, discuss the ways in which some forms of transparency are utilised in developing distributed Java applications.

(15 marks)

3. (a) Discuss the role played by the *verifier*, the *class loader* and the *security manager* in ensuring Java code's security. **(6 marks)**
- (b) (i) Explain the role played by digital signatures and certificates in exchanging documents/code securely
- (ii) Using as an example the exchange of application code between hosts in a distributed system, explain how Java's tools can be used for signing code and generating signatures. **(8 marks)**
- (c) (i) Compare and contrast the encryption algorithms used by RSA and DES. Upon what do the strengths of both algorithms rely?
- (ii) Using an example, show how keys are generated using RSA.
- (iii) Show how the keys generated in part (ii) above can be used to encrypt and decrypt data. **(11 marks)**
4. (a) (i) Discuss in detail the function of the Internet protocol, IPv4 and its successor, IPv6.
- (ii) Briefly describe mobile IP. **(7 marks)**
- (b) "In developing a new protocol for a distributed system, a decision must be made about which existing protocol to use to transmit the messages of the new protocol."
- Traditionally, TCP and UDP have been used, although recently HTTP has been used as a protocol to transmit messages for new protocols such as SOAP.
- Compare and contrast the services that are provided by
- (i) TCP
- (ii) UDP
- (iii) HTTP
- to a higher layer protocol, and identify the situation(s) where one may be preferred over the others. **(10 marks)**

- (c) Describe the Java APIs that are used to manage
- (i) TCP sockets
 - (ii) UDP datagrams
- Contrast Java's treatment of the two protocols in light of the differences you identified in part (b) above.

(8 marks)

Section B

5. (a) Define the term *middleware*. Using examples describe why it can be advantageous to the application developer.

(4 marks)

- (b) "Java RMI permits the use of dynamic class loading." Describe what is meant by dynamic class loading and in what situations it would be useful.

(4 marks)

- (c) You are required to design a distributed object based airline booking system. The system must have the following characteristics:

- ◆ The client can connect to the server and request information on a given flight, identified by a flight number represented as a Java string. An object is returned containing the flight times as an array, the stopovers as an array and the price as a string.
- ◆ The client can book a flight by passing an object which contains the customer's name, address and credit card details, as well as a separate string to represent the flight number.

- (i) Using diagrams, identify the various classes that are necessary for this application. Clearly distinguish the remote and non-remote classes.
- (ii) List and explain the sequence of events that takes place when a client contacts the server to get flight information, and then book a flight, assuming that the client must perform lookup and dynamically load the required classes.
- (iii) How are remote objects handled differently to non-remote objects when passed as parameters to remote methods?

(17 marks)

6. (a) What is meant by the term *client callback*, when used in a distributed object system, and how is it implemented using Java RMI?
(5 marks)
- (b) Using a sample scenario, describe *distributed event based systems*. Identify the roles fulfilled by each of the participants.
(8 marks)
- (c) Show how the scenario presented in part (b) above could have been implemented using Java RMI. Clearly describe all the classes used, and the roles they fulfil.
(12 marks)
7. (a) Show using an example, the potential uses of Enterprise Java Beans (EJB).
(6 marks)
- (b) Describe fully the *ACID* properties of transactions.
(8 marks)
- (c) Describe in detail the difference between *flat transactions* and *nested transactions*.
(5 marks)
- (d) Using an example, explain the *two phase commit* protocol.
(6 marks)
8. (a) Define the term *web service*.
(3 marks)
- (b) Using diagrams and sample applications, describe the role fulfilled by each of the following protocols/standards in a web service architecture.
(13 marks)
- (i) HTTP
 - (ii) XML
 - (iii) SOAP
 - (iv) WSDL
 - (v) UDDI
- (c) Compare and contrast web services, Java RMI and CORBA.
In your answer you should identify how their various components/services relate to each other.
(9 marks)