

OLLSCOIL Na hÉIREANN
THE NATIONAL UNIVERSITY OF IRELAND

NATIONAL UNIVERSITY OF IRELAND, GALWAY.

SUMMER EXAMINATIONS 2000

Fourth University Examination in Information Technology

CT414 Distributed Systems

Prof. D Bell
Dr G. Lyons
Mr. D. Chambers

Answer any 5 questions
All questions carry equal weight
Time allowed: **three hours**

1:a: The Java approach to distributed object programming is to use the built-in Remote Method Invocation (RMI) mechanism. Describe briefly the main differences between Java RMI and CORBA. **4 MARKS**

b: With respect to Java Remote Method Invocation explain the following:

- (i) The *rmic* compiler.
- (ii) The Remote interface.
- (iii) The Serializable interface.
- (iv) The RMI Registry.

8 MARKS

c: Using the JAVA RMI Model and the *rmic* compiler - outline the design of a simple Virtual Banking System, which will be used to manage the accounts available on a remote bank server.

- The service can be used by authorised account holders to browse their accounts, and conduct transactions as required e.g. make deposits, make withdrawals, transfer funds to other accounts, etc.
- Design should incorporate a *capability* based security mechanism for access to privileged remote methods (e.g. account access methods), initial user validation can be done using a password.

Full implementation classes **are not** required but answer should include the necessary JAVA interfaces and specify what classes would be automatically generated by *rmic* in this case. **8 MARKS**

2.a: Explain briefly what is meant by a *Distributed System*. Why do these often occur naturally in a typical computing environment ? 4 MARKS

b: In the context of distributed load balancing, describe how *flooding* might occur in a distributed system. What steps can be taken to avoid flooding processing elements with jobs ? 8 MARKS

c: Describe the semantics of a typical synchronous *Remote Procedure Call* operation. Based on this description, show how an RPC library might implement these semantics on top of a connection-less transport layer (like UDP). In particular, show how this supports *non-idempotent* operations. 8 MARKS

3.a: What is the *Distributed Computing Environment* from the Open Software Foundation ? Outline briefly some of the services it provides. 4 MARKS

b: Describe the main properties of an *atomic transaction*. Explain how, using the two-phase commit protocol, an atomic commit might be achieved in a distributed system. 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

4.a: Outline the basic purpose of name resolution services. Using the CORBA standard name service as an example, show briefly how a client program could utilise this service. 4 MARKS

b: Describe the function and operation of the CORBA Event Service. Include in your description the main interfaces provided by this service. What kind of applications could benefit from using this service ? 8 MARKS

c: Describe a suitable set of CORBA interfaces that could be used to manage distributed multimedia data streams. In particular, outline the steps required (and the interface interaction) during stream establishment. 8 MARKS

5.a: What output does the CORBA IDL compiler typically produce ? For example, if a file called bank.idl is compiled using a CORBA IDL to Java mapping, what output files are produced and what are their function ? 8 MARKS

b: Given the following IDL interface definition, for a Hotel Reservation System, write a suitable implementation class, in Java, for the Hotel interface:

```
// File Hotel.idl
typedef string Date;
```

```
interface Hotel
```

```
{
```

```
    readonly attribute atring name;
```

```
    readonly attribute string tel_No;
```

```
    readonly attribute string fax_No;
```

```
    readonly attribute string no_Rooms;
```

```
    Booking makeBooking(in string c_name, in short r_type, in Date when);
```

```
    float getPrice(in Date when);
```

```
    boolean checkRoomFree(in short r_type, in Date when);
```

```
};
```

You may assume that the interface Booking is defined separately.

Assuming that an instance of Hotel is set-up on node *geminga.nuigalway.ie* with the registered server name *Ritz*, write the client code in Java to query the Hotel on availability and price. 12 MARKS

6:a: In the context of distributed object technology, discuss briefly what is meant by a "*Smart Proxy*". What complications can arise when using smart proxies and how might these be resolved ?
4 MARKS

b: Explain how distributed objects may be uniquely identified using object references. In particular, describe what information is stored in an object reference. How are object references typically made available to (or obtained by) client applications ?
8 MARKS

c: Describe the typical usage and operation of "*filter*" objects. Using a suitable example, show how a Thread Filter could be used to provide support for a multithreaded server implementation.
8 MARKS

7:a Explain briefly the purpose of Java Servlets. What are the main advantages of using Servlets instead of CGI Scripts ?
4 MARKS

b: Describe the top-level architecture of the *javax.servlet* package. When a servlet accepts a call from a client it is passed two parameters i.e. a *HttpServletRequest* object and a *HttpServletResponse* object. Show using a suitable example, how a servlet might use these parameters.
8 MARKS

c: Using the example of a Virtual Book Store, write a Java Servlet that provides session tracking based on using the *HTTPSession* interface in conjunction with a user defined *ShoppingCart* object.
8 MARKS