

**PRIFYSGOL CYMRU; UNIVERSITY OF WALES**

**DEGREE EXAMINATIONS MAY/JUNE 2003**

**SWANSEA**

**Computer Science**

**CS 338 Internet Computing**

**Attempt 2 questions out of 3**

**Time allowed: 2 hours**

**Students are permitted to use the dictionaries provided by the University**

**Students are NOT permitted to use calculators**

**CS\_338**  
**INTERNET COMPUTING**

*(Attempt 2 questions out of 3)*

**Question 1**

(a) *Briefly* describe the sequence of events that occurs when an RMI client object looks up an RMI server object in the RMI registry, and makes a remote method call on that server object. You may find a diagram helpful. **[5 Marks]**

(b) Extend your answer to part (a) to explain what happens when the server object is behind a firewall that:

- i. does not allow the Java RMI (JRMP) protocol;
- ii. only permits HTTP connections on Port 80; and
- iii. requires all communication with machines behind the firewall to go via proxy servers.

(For simplicity, you may assume the RMI registry is *outside* the firewall, and you need not concern yourself with how the initial server object binding process is managed.)

**[5 Marks]**

(c) The following class is from a socket-based application that controls a demand dialling manager for a dialup internet connection. (Note that this is substantially simplified from the version in the course notes.)

```
import java.io.*;
import java.net.*;

public class ControlSMPPPD {

    private String remoteHost;

    public ControlSMPPPD(String host) {
        remoteHost = host;
    }

    public void linkUp() throws IOException {
        controlLink("start");
    }

    public void linkDown() throws IOException {
        controlLink("stop");
    }

    private void controlLink(String command) throws
        IOException {

        /* Code which opens a socket and communicates
         * using PrintStream and BufferedReader omitted,
         * though see Question 2.
         */
    }
}
```

Suppose you wished to use this class to make RMI server objects.

- What changes would you need to make to the code in order to enable it to function as an RMI server object?
- Write an appropriate interface for `ControlSMPPPD`.
- Write the *server side* initialisation, or *launch*, code that would be needed.
- Write the *client side* initialisation, or *launch*, code that would be needed.
- `ControlSMPPPD` can throw an `IOException`. What happens to exceptions that are thrown on the *server side* as a consequence of remote method calls from the *client side*?

Minor syntactic errors in your Java code will not be penalised.

**[15 Marks]**

## Question 2

(a) In the context of TCP/IP and Java, what is a *socket* and how does it differ from a *port*? What are *well-known ports*? Give some examples of well-known ports. **[5 Marks]**

(b) Consider the following code, and explain briefly (a) what it does and (b) how it works.

```
for (int i = 0; i < MAX_PORT; i ++) {
    try {
        Socket socket = new Socket(host, i);
        out.println("Open Port number: " + i);
        socket.close();
    }
    catch (IOException e) {
    }
}
```

You may assume that `MAX_PORT` and `host` have been appropriately defined.

**[5 Marks]**

For the remaining parts of Question 2, consider the implementation of the `controlLink` method that was omitted from Question 1 (you do not need to have answered Question 1 to attempt this question).

The class to which the (private) method `controlLink` belongs only calls it with values for the `String` parameter `command` of: “start” or “stop”. (However, it would be good practice to consider the possibility of future changes which could result in `controlLink` being called with other values for `command`.)

```
private void controlLink(String command) throws
    IOException {

    Socket socket = new Socket(remoteHost, SMPPPD_PORT);

    PrintStream out =
        new PrintStream(socket.getOutputStream());

    out.println(command);

    finally {
        out.close();
        socket.close();
    }
}
```

(c) Briefly explain the meaning of the lines from “`Socket socket = ...`” to “`out.println(command)`”. Why are the two `close` method calls in a `finally` clause? **[5 Marks]**

(d) The method `controlLink` opens a *client* socket connection, which requires a listening *server socket* to function. Assume the existence of a class called `Dialler` with methods `start()` and `stop()`, and write some Java code that does the following.

- (i) Accepts socket connections from clients over a socket `SMPPPD_PORT` (the precise port number is not important).
- (ii) Reads lines of text from the socket. If a line consists of the text “start”, it should invoke the `start()` method; if a line consists of the text “stop”, it should invoke the `stop()` method; and otherwise it should do nothing.
- (iii) When a socket connection is closed, it starts listening for new connections on port number `SMPPPD_PORT`.

Minor syntactic errors in your Java code will not be penalised.

**[10 Marks]**

### Question 3

You are involved in developing a *shared diary* system for a corporate intranet system. The system will enable employees to manage their appointment diaries on-line, and will enable other employees to see when anyone is busy. Initially, two *prototypes* are being built to (a) evaluate technologies; and (b) get user feedback.

The prototypes will be a bit basic, and will have the following characteristics.

- Users will be able to enter *events* into their diaries at any time, and will be able to specify the duration of an event.
- Users will be asked to enter a free-text description of an event, and also a *category*. The category will broadly categorize events (for example, categories could include things like: *management meeting*; *client meeting*; *off-site meeting*).
- Access to a users' own diary will be password protected, and only the owner of a diary will be able to see the free-text descriptions. However, *any* user will be able to view the categories of *any* other user.
- An *administrators* interface. This will permit, for example, definition of categories, and the creation/deletion of users. Access to this interface will need to be protected.

The strategic decision has been made to represent diary data using XML – for easy manipulation, storage and distribution.

The two prototypes will use the following technologies.

- Java applets managing the user-interface, communicating via RMI with Java server objects that manage the database of diaries.
- Client-side scripts within HTML pages managing the user-interface, communicating via HTTP with server-side scripts that manage the database of diaries.

You are working on the Java version.

(a) *Briefly* describe the two different sets of technologies involved.

**[5 Marks]**

(b) Outline a possible *design* for the part of the Java-based system including the user interface applets and the server side code they communicate with. Your answer should consider the following aspects.

- The *functionality* of the *user interface*, and the effect it will have on the required functionality of the server-side code.
- The *functionality* of the *administration interface*, and the effect it will have on the required functionality of the server-side code.
- *Security* issues – remember, that some password mechanism is required, and users should only have limited access to other employees' diaries.
- What *information* needs to be stored.

Note that you are not being asked to write any code. Nor are you being asked to argue that the Java approach is better than the other approach. The principle issues you should focus on are (i) communication and (ii) the server-side API. You are not being asked to consider database issues, and nor should you concern yourself with details of user interface appearance. **[10 Marks]**

(c) The team implementing the client- and server-side scripting version is considering three ways to render the XML-based diary data to the display.

- Transform it on the server using an XSL style sheet.
- Transform it in the client's browser using an XSL style sheet.
- Render it using an XML parser and client-side scripting within the client's browser. (For example, Microsoft's DOM-based XML parser is built-in to recent versions of Internet Explorer for Windows, and is accessible from Javascript/JScript and VBScript.)

Discuss the advantages and disadvantages of each of these approaches. **[10 Marks]**