UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

MASTER OF SCIENCE IN COMPUTER SCIENCE

MOBILE COMPUTATION

Friday 20 April 2001

14:00 to 15:30

Examiners:

I. Watson (External)

R.N. Ibbett (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. (a) Mobile code which is loaded over the network is assumed to be untrusted. The Java Virtual Machine submits such code to *byte-code verification*. What properties are checked in this process? [*6 marks*]

   (b) After byte-code verification Java applets are prevented from inflicting damage to the local file system by *sandboxing*. One part of sandboxing is preventing applets from reading and writing to the local file system. What other restrictions are imposed by sandboxing? [*5 marks*]

   (c) Which component of the Java run-time system is responsible for enforcing the sandboxing policy? How does this component ensure that it can control access to security-sensitive operations? [*3 marks*]

   (d) The Jalapeño Virtual Machine is a virtual machine for Java servers which is itself written in the Java language.

      i. How does the Jalapeño Virtual Machine implement the run-time services of the virtual machine? [*3 marks*]

      ii. With regard to the other processes of the virtual machine, what special care must be taken by the Jalapeño Virtual Machine while executing these run-time service methods? [*3 marks*]

      iii. What special requirements are placed on a Java virtual machine for a Java server over and above those for a Java virtual machine which is to run on a typical workstation? [*5 marks*]

2. (a) The Java programming language provides native support for concurrent *threads* of execution within a program. However, many workstations have only a single processor. What is the usefulness of the thread construct in this context? [*4 marks*]

(b) What are the two methods of creating new threads of execution in Java? [*4 marks*]

(c) How are *daemon* threads treated differently from other threads in the Java Virtual Machine? [*2 marks*]

(d) Each Java object contains a *lock* (also called a *monitor*). What role does this play in multi-threaded programming? [*4 marks*]

(e) The thread methods `stop()` and `suspend()` have been deprecated since Java 1.2. The use of these methods was found to encourage poor concurrent programming style.

i. What were the common programming problems which occurred with the use of the `stop()` method? What caused these problems? [*3 marks*]

ii. What were the common programming problems which occurred with the use of the `suspend()` method? What caused these problems? [*3 marks*]

(f) How do Java programmers implement graceful thread termination *without* using the `stop()` method? Illustrate your answer with a Java program fragment. [*5 marks*]

3. (a) Describe briefly Java's *remote method invocation* (RMI). [*5 marks*]

   (b) What is the role of *remote exceptions* in RMI? [*5 marks*]

   (c) How do *activatable objects* differ from other remote objects? [*5 marks*]

   (d) What are the limitations of RMI when compared to other distributed object systems such as CORBA? [*10 marks*]