

CARDIFF UNIVERSITY EXAMINATION PAPER

Academic Year:	2010/2011
Examination Period:	Autumn
Examination Paper Number:	CM0247
Examination Paper Title:	Application Building with Java
Duration:	2 hours

Do not turn this page over until instructed to do so by the Senior Invigilator.

Structure of Examination Paper:

There are 3 pages.

There are 3 questions in total.

There are no appendices.

The maximum mark for the examination paper is 80 and the mark obtainable for a question or part of a question is shown in brackets alongside the question.

Students to be provided with:

The following items of stationery are to be provided:

ONE answer book.

Instructions to Students:

Answer all THREE questions.

The use of a translation dictionary between English or Welsh and another language, provided that it bears an appropriate school stamp, is permitted in this examination.

1. (a) Explain what the access level of a member of a class determines and how to declare the access levels. Discuss the differences between the four access levels. [10]
- (b) Explain the concept of event-driven programming. What is the basic idea? Which main types of object are involved and how do they interact? How in principle can an event processing system be implemented? [15]
2. (a) Explain how to create a thread in Java which can always be cancelled on request. (There is no need to write a complete program, but in addition to your description you should provide pieces of Java code demonstrating the principle). [10]
- (b) Why must threaded programs use locks? Discuss the two main issues involved and explain why locking is essential for each of them. [15]
- (c) Explain what problem the program below exhibits. Briefly discuss a general strategy to avoid this problem in your own programs.

```
import java.util.concurrent.*;

class ThreadTest {
    Double n1 = new Double (1.0);
    Double n2 = new Double (2.0);

    Thread thread1 = new Thread () {
        public void run () {
            while (true) {
                synchronized (n1) {
                    synchronized (n2) {
                        n1 = (n1 + n2) / 2.0;
                        System.out.println ("n1_=_ " + n1);
                    } } } } };
    Thread thread2 = new Thread () {
        public void run () {
            while (true) {
                synchronized (n2) {
                    synchronized (n1) {
                        n2 = (n2 + n1) / 2.0;
                        System.out.println ("n2_=_ " + n2);
                    } } } } };

    public static void main (String[] args) {
        ThreadTest test = new ThreadTest ();
        test.thread1.start ();
        test.thread2.start ();
    }
}
```

[10]

3. Write the Java code for the classes needed to implement a network server which receives incoming requests on port 8080, and dispatches them to the `task ()` method of a `Helper` class using an executor service. If the incoming request is a shutdown-server request, `task ()` will return **true**; otherwise it will return **false**. The server should terminate accordingly if `task ()` returns **true**. Briefly discuss what happens if the server receives a lot of requests within a short time.
(You may assume that the `Helper` class exists and provides a method **boolean** `task (Socket s)` to deal with the request from a client via the socket `s`). [20]