DUBLIN CITY UNIVERSITY

SEMESTER ONE EXAMINATIONS 2011

MODULE TITLE:	Concurrent Programming
MODULE CODE:	CA463/CA463D
COURSE:	BSc. in Computer Applications (Software Engineering Stream), Study Abroad
YEAR:	4/X
EXAMINERS:	Dr. J. Power Dr. F. Bannister Dr. Martin Crane Ext: 8974
TIME ALLOWED:	2 Hours
INSTRUCTIONS:	Please answer **any 3** questions:

Requirements for this paper
Please tick (X) as appropriate

	Log Table
	Graph Paper
	Attached Answer Sheet
	Statistical Tables
	Floppy Disk
	Actuarial Tables

THE USE OF PROGRAMMABLE OR TEXT STORING CALCULATORS IS EXPRESSLY FORBIDDEN

Please note that where a candidate answers more than the required number of questions, the examiner will mark all questions attempted and then select the highest scoring ones.

PLEASE DO NOT TURN OVER THIS PAGE UNTIL YOU ARE INSTRUCTED TO DO SO

Question 1 **[Total marks: 33]**

1(a) [6 marks]

Define Amdahl's law in words and as a formula.

1(b)

The efficiency of a parallel code is the serial cost: parallel cost ratio and is given by E_P = T_1/(*p*XT_P) (where T_1 is time on one processor and T_P is time on p processors).

[3 marks]

(i) Using Amdahl's law derive an expression for the efficiency E_P in terms of the number of processors *p* and the serial fraction *s*.

[3 marks]

(ii) Using the result in 1 (b) (i) above, what would be the effect on the parallel efficiency (i.e. increase or decrease) of an *increase* in the number of processors *p*? Give a concise reason for your answer.

[4 marks]

(iii) Using the result in 1 (b) (i) above, what would be the effect on the parallel efficiency (i.e. increase or decrease) of an *increase* in the size of the problem? Give a concise reason for your answer.

[4 marks]

(iv) Using the result in 1 (b) (i) above, what would be the effect on the parallel efficiency (i.e. increase or decrease) of an *increase* in the speed of the processors? Give a concise reason for your answer.

[4 marks]

(v) Derive a functional relationship which shows how *s* would have to vary to maintain constant efficiency.

1(c) [9 marks]

Suppose you are given some piece of sequential code that is to be parallelised. You are told that 20% of the code cannot be sped up at all and, of the remainder, 50% can be sped up by a factor of 3 and 50% can be sped up by a factor of 2. Using the fact that the reduced execution time, T_P, is given by the original time T_1 divided by the speedup, S_P, derive an expression for *overall* speedup that can be achieved on the given code. Comment on your result.

--[End of Question 1]--

Question 2 **[Total marks: 33]**

2(a) [8 marks]

Explain the differences between semaphores and monitors in SR.

2(b) [8 marks]

Give a high level description of the algorithm for the solution of the Producer-Consumer problem for infinite buffers using semaphores. Write SR code that implements the algorithm.

2(c) [17 marks]

Using monitors, write code in Java to implement a solution to the Producer Consumer Problem. Your implementation should provide the following classes:

1. `class Buffer` which has methods `get` ( ) and `put` ( ) to retrieve and append integers onto a buffer. (The `Buffer` need only store one integer at a time).
2. `Producer` and `Consumer` classes which utilise the `put( )` and `get( )` methods (above) respectively.
3. A `ProducerConsumerTest` class to test the classes above by instantiating producer and consumer objects.

--[End of Question 2]—

Question 3 **[Total marks: 33]**

3(a) [9 marks]

Briefly describe synchronous and asynchonous parallel heartbeat algorithms. How do the termination conditions differ for centralised and distributed heartbeat algorithms?

3(b) [9 marks]

Implement in SR and briefly describe the Parallel Synchronous Odd-Even Exchange Sort Heartbeat Algorithm for sorting an array into ascending order over 2 processors.

3(c) [15 marks]

Write code that implements the general form of the *centralised* Odd-Even Exchange Sort Algorithm for an array to be sorted over `k=n` processors in Java. You may presume that the array to be sorted, `a[],` is stored by all processors and that the algorithm will terminate after `n/2` iterations.

--[End of Question 3]—

[Total marks: 33]

Question 4

4(a) [10 marks]

Explain the difference between Low Level Concurrency Objects and High Level Concurrency Objects in Java. Why would you use the latter rather than the former?

4(b) [23 marks]

Write code that implements the Bounded Last-In, First-Out (LIFO) Stack Class in Java using the `Lock` and `Condition` objects and the `LinkedList` class, giving code for the `void push(Object o)` and `Object pop()` methods.

--[End of Question 4]—

Question 5 **[Total marks: 33]**

5(a) [6 marks]

What are the three major components of a Java application that uses Remote Method Invocation (RMI)?

5(b) [9 marks]

Describe the steps involved in programming a Java RMI application.

5(c) [18 marks]

Write code that implements the Remote Database Server in Java RMI (i.e. the interface, the client and the server). Your code should implement `read()` and `write()` methods on the database server, where `read()` returns the value of the (integer) database and `write(int value)` sets the database to be a particular (integer) value.

--[End of Question 5]