DUBLIN CITY UNIVERSITY

AUGUST/RESIT EXAMINATIONS 2016/2017

MODULE: CA4006 – Concurrent and Distributed Programming

PROGRAMME(S):

CASE	BSc in Computer Applications (Sft.Eng.)
CPSSD	BSc in ComputationalProblem Solv&SW Dev.
ECSAO	Study Abroad (Engineering & Computing)
ECSA	Study Abroad (Engineering & Computing)

YEAR OF STUDY: 4,O,X

EXAMINER(S):

Dr. Martin Crane	(Ext:8974)
Prof. David Bustard	
Dr. Ian Pitt	

TIME ALLOWED: 3 Hours

INSTRUCTIONS: Answer 4 questions. All questions carry equal marks.

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

There are no additional requirements for this paper.

QUESTION 1 ***[TOTAL MARKS: 25]***

Q 1(a) **[8 Marks]**

Distinguish clearly between processes and threads. Show by means of a diagram and definition what is meant by Multithreading. Explain the difference between Implicit and Explicit Multithreading.

Q 1(b) **[17 Marks]**

.

[3 marks]

(i) Describe Amdahl's Law for the theoretical speedup of a program executed across multiple processors. The answer should give the formula involved and explain it.

[4 marks]

(ii) For a piece of code you have written, you know that memory operations currently take 30% of execution time. A new widget speeds up 80% of memory operations by a factor of 4 and a second new widget speeds up 1/2 the remaining 20% by a factor or 2. Using Amdahl’s law calculate the total speed up from these two widgets.

[5 marks]

(iii) Latency may be loosely defined as the time interval between the stimulation and response, or, from a more general point of view, as the time delay between the cause and the effect of some physical change in the system being observed. Interpret latency in terms of Amdahl’s law, expressing it in terms of the speedup and the parallel fraction. Use this to show that, in terms of the speedup, things can only get so fast, but they can get arbitrarily slow.

[5 marks]

(iv) Gustafson-Barsis’ law contrasts with Amdahl's law, as it describes a limit on the speed-up that parallelization can provide, given a fixed data set size. On the other hand, Gustafson-Barsis’ law says that there is a fixed amount of parallel work per processor i.e. computations involving arbitrarily large data sets can be efficiently parallelized. Given the formula for speedup according to Gustafson-Barsis’ law:

$$Speed-up, S = k(P - s(P-1))$$

where *S* is the speedup on a particular processor, *P* is the number of Processors, *s* the non-parallelizable fraction of work on any parallel process and *k* is constant, graph *S* against *P*. What are the implications of Gustafson-Barsis' law for large datasets that Supercomputers are currently applied to?

[End of Question 1]

QUESTION 2 ***[TOTAL MARKS: 25]***

Q 2(a) **[8 Marks]**

Define Dekker's Algorithm in words and implement it in SR for two processors. Explain clearly all parts of the code.

Q 2(b) **[17 Marks]**

Peterson's algorithm is a variation on Dekker's algorithm whereby each processor uses two variables, `flag` and `turn`. A `flag` value of 1 indicates that the process wants to enter the critical section. The variable `turn` holds the process ID whose turn it is. Entrance to the critical section is granted for process P0 if P1 does not want to enter its critical section or if P1 has given priority to P0 by setting `turn` to 0.

[9 marks]

(i) Implement Peterson's algorithm in Java and briefly compare it with Dekker's algorithm.

[8 marks]

(ii) Show that, in Peterson's algorithm in (i), Mutual Exclusion is satisfied and No Starvation occurs.

[End of Question 2]

QUESTION 3 ***[TOTAL MARKS: 25]***

Q 3(a) **[8 Marks]**

What are the 4 types of processes in Message Passing Programs?

Q 3(b) **[6 Marks]**

Describe and give pseudo-code for the Synchronous Sieve of Eratosthenes algorithm with message passing. What is the principal disadvantage of the synchronous version of the algorithm?

Q 3(c) **[11 Marks]**

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

[End of Question 3]

QUESTION 4 ***[TOTAL MARKS: 25]***

Q 4(a) **[3 Marks]**

Outline the steps involved in programming a Remote Method Invocation (RMI) application.

Q 4(b) **[7 Marks]**

Explain in detail the role of the RMI Registry in the context of Remote Method Invocation (RMI) applications. What is the role of the `bind` and `lookup` methods provided by the `Naming` Class? What is the role of the Security Manager in RMI applications?

Q 4(c) **[15 Marks]**

A Remote Method Invocation (RMI) Interface to allow a client to download any type of file (plain text or binary) from a remote machine is shown in Figure Q4 You are required to implement the remote interface, develop the server and develop a client that invokes the remote method `downloadFile`. Your code should be fully commented and should document the function of each major component.

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface FileInterface extends Remote {
   public byte[] downloadFile(String fileName) throws
   RemoteException;
}
```

Figure Q4. `FileInterface.java`

[End of Question 4]

QUESTION 5 ***[TOTAL MARKS: 25]***

Q 5(a) **[7 Marks]**

Give four differences between SOAP & Rest Web services. Are there any circumstances in which it is preferable to use SOAP over Rest Web Services?

Q 5(b) **[6 Marks]**

What is meant by saying that a command is idempotent? Describe the four Rest operations, saying whether or not they are idempotent.

Q 5(c) **[12 Marks]**

Part of an Java Interface `Addition.java` to allow a client to invoke a command to add two (long) integers is shown in Figure Q5. Using Java SOAP Web Services, give the implementation of the following components of this interface: the Service Endpoint Interface (SEI), the Service Implementation Bean (SIB) and the Endpoint Publisher. You should fully comment your code.

```
public long add(long x, long y){
      // implementation omitted
}
```

Figure Q5. Interface `Addition.java`

[End of Question 5]

[END OF EXAM]