

THE AUSTRALIAN NATIONAL UNIVERSITY
Mid Semester Examination – September 2009

COMP2310
Concurrent and Distributed Systems

Study Period: 0 minutes

Time Allowed: 2 hours

Permitted Materials: NONE

Questions are NOT equally weighted.

The questions are followed by labelled, framed blank panels into which your answers MUST be written. No additional paper will be provided. Writing outside of the boxes will not be marked.

This exam is marked out of 100. You should answer all questions.

Your mark for this exam will contribute to 20% of your total course mark, according to the assessment scheme given on the course web page.

Your student number must be written at the top of every sheet of the exam paper.

Student Number:

The following are for use by your friendly examiner!

Q1 Mark	Q2 Mark	Q3 Mark	Q4 Mark	Q5 Mark	Total Mark

Question 1 [20 marks] General Concurrency and its Support

- (a) Give **two** fundamentally different motivations (or reasons) for wanting to write concurrent programs AND provide a real life example to illustrate each. (The example can be the same or different, but must clearly illustrate each of the two motivations you give.)

[6 marks]

- (b) A process can have multiple threads. List **two** resources that are shared between all threads in a process AND **two** resources that are unique to each thread.

[4 marks]

UID:

- (c) Standard concepts of program correctness include partial and total correctness. Explain in words what is meant by each of these concepts AND why they may not be applicable to concurrent systems.

[6 marks]

- (d) List **four** fundamentally different events that can cause a running process to relinquish its access to the CPU.

give up

[4 marks]

Question 2 [30 marks] Shared Memory Synchronization Constructs

- (a) Show how you would implement mutual exclusion between two threads using a hardware supported atomic test-and-set operation.

[6 marks]

- (b) List two ways in which the Java notification methods differ from semaphores.

[4 marks]

UID:

(c) Give **five** basic characteristics of a monitor.

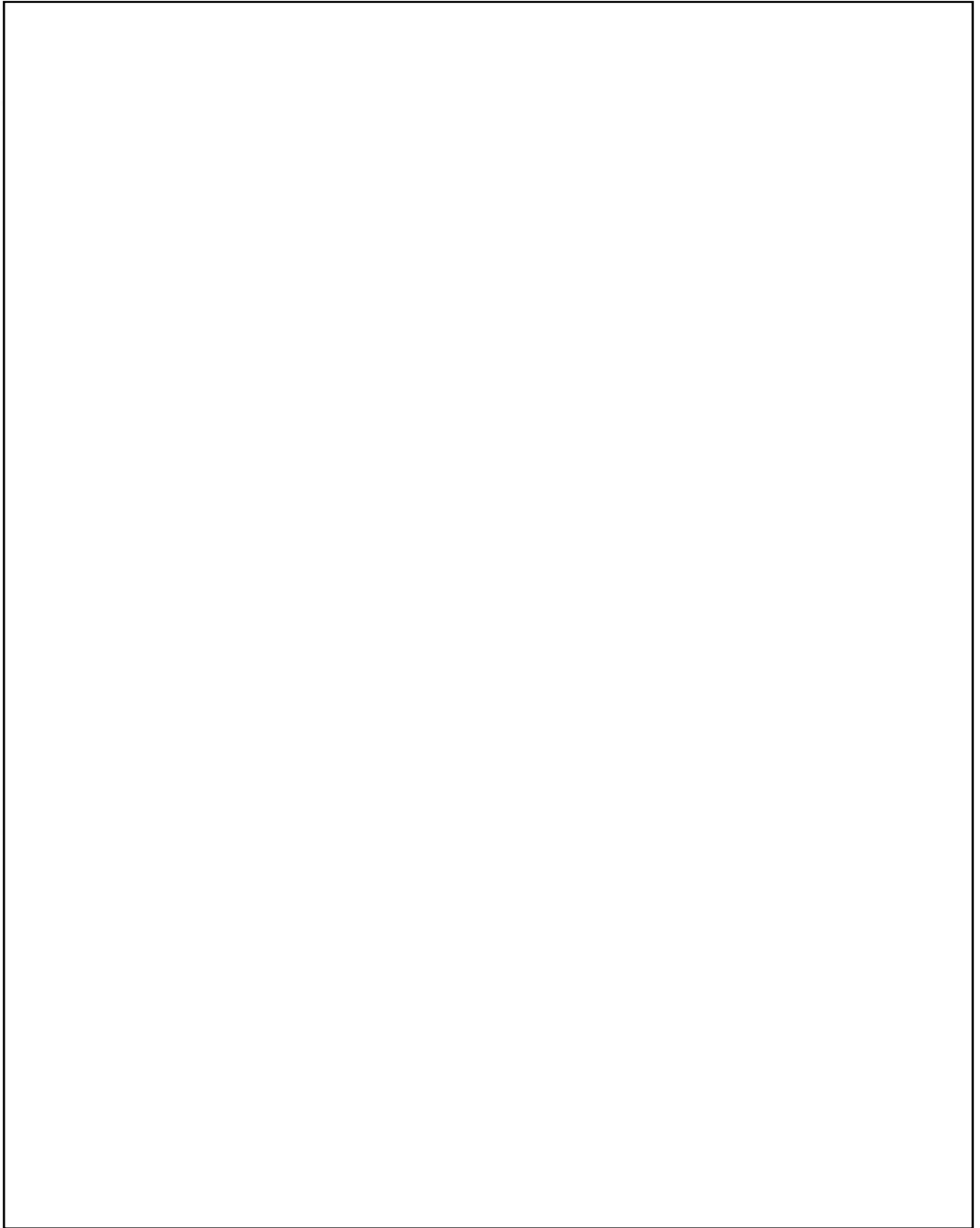
[5 marks]

(d) Show how you would use a binary semaphore to construct a general semaphore.

[6 marks]

UID:

- (e) Explain the difference between a function, procedure and entry in a protected object, AND give one example application scenario that uses a protected object that has all three of these interfaces.



[9 marks]

UID:

Question 3 [15 marks] Message Passing

(a) Give **two** ways in which a remote procedure call differs from simple message passing?

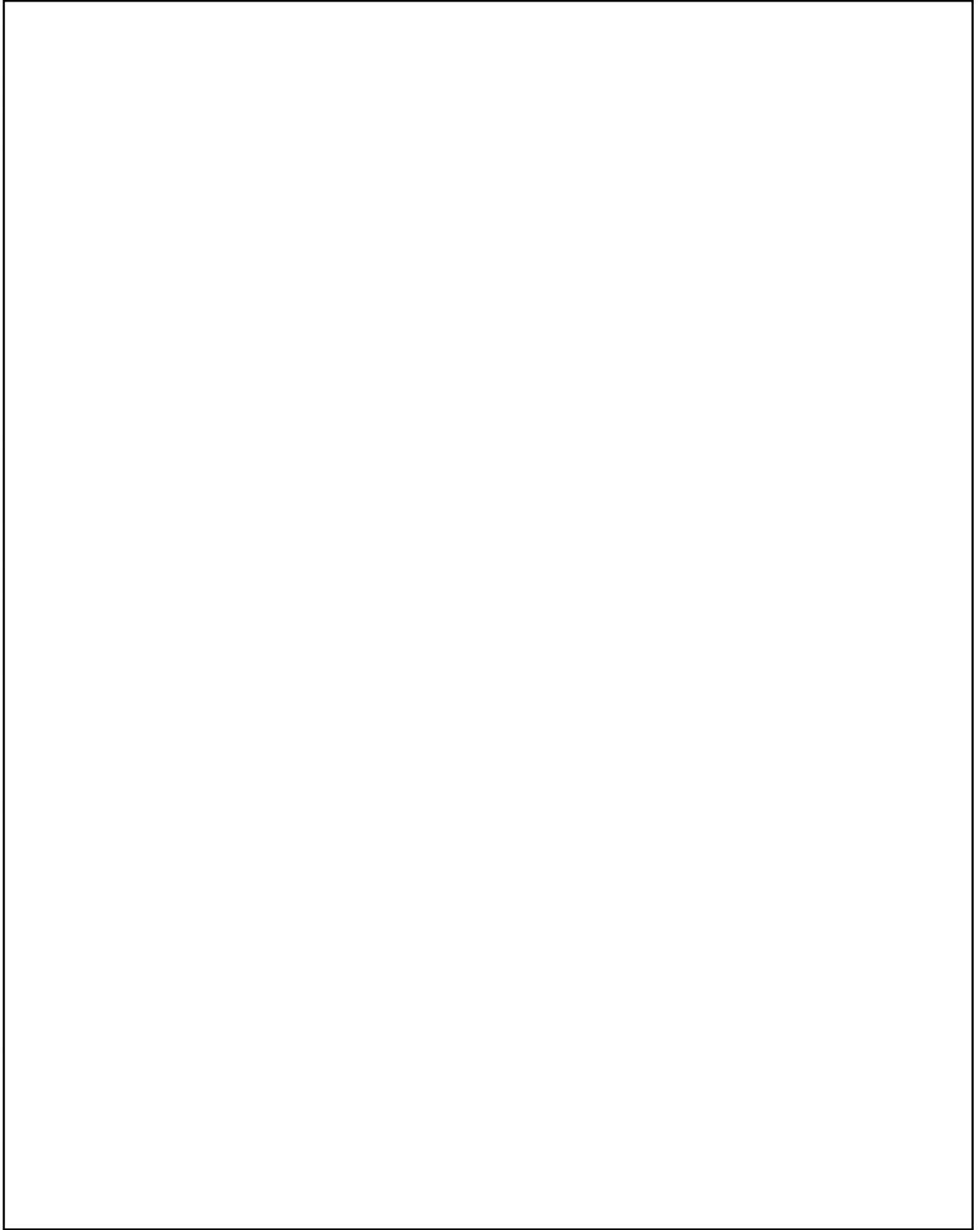
[4 marks]

(b) Draw a diagram to illustrate how you would synchronize N processes using asynchronous message passing.

[5 marks]

UID:

- (c) Message passing systems can be characterized as supporting communications that are i) ordered or non-ordered ii) symmetrical or asymmetrical iii) synchronous or asynchronous iv) direct or indirect. Define what is meant by ANY THREE of these characterizing features.



[6 marks]

Question 4 [10 marks] Non-Determinism

- (a) Explain the statement “Moving from a single-processor to a multi-processor system often exposes hidden non-determinism”.

[2 marks]

- (b) How does Dijkstra’s guarded commands differ from a classical “C” style switch statement?

[4 marks]

- (c) Detail two different ways by which the Ada `select` statement can be used to supports non-deterministic program behaviour.

[4 marks]

Question 5 [25 marks] Concurrent Code Analysis

The following Ada code compiles and runs to completion. It creates a number of child tasks, with each child assigned a unique identifier. Each child task then aims to sum the values of all child task identifiers storing the result in local variable MyCounter.

```
with Ada.Text_IO; use Ada.Text_IO;

procedure Reduction is

    Size : Constant := 8;

    task type Task_Type is
        entry Receive_Task_Id (Task_Id : in Natural);
        entry Combine (Counter : in out Natural);
    end Task_Type;

    Task_Array : array (0..Size-1) of Task_Type;

    task body Task_Type is
        MyId, MySwap, MyCounter, I : Natural;
    begin
        accept Receive_Task_Id (Task_Id : in Natural) do
            MyId := Task_Id;
        end Receive_Task_Id;

        MyCounter := MyId;
        I := 2;
        while I <= Size loop
            if MyId mod 2 = 0 then
                MySwap := (MyId + I - 1) mod Size;
                Task_Array (MySwap).Combine(MyCounter);
            else
                accept Combine(Counter : in out Natural) do
                    Counter := Counter + MyCounter;
                    MyCounter := Counter;
                end Combine;
            end if;
            I := I * 2;
            delay 0.1;
        end loop;

        Put_Line ("Task " & MyId'Img & " MyCounter " & MyCounter'Img );

    end Task_Type;

begin

    for I in 0..Size-1 loop
        Task_Array(I).Receive_Task_Id (I);
    end loop;

end Reduction;
```

UID:

- (a) How many tasks are created AND assuming the program executes according to the specifications, what output should be produced by each task from the Put_Line statement.

[4 marks]

- (b) When run once on a certain machine the code produced the “correct” output (meaning the expected result given the program specification). Draw a diagram to illustrate how the various rendezvous calls are being used to sum correctly the values of MyID across all tasks.

[5 marks]

UID: _____

- (c) When run a second time the same code produced incorrect results, but otherwise terminated normally. Explain what is causing the program to sometimes produce the correct results and sometimes produce wrong results. Are the wrong results always the same wrong results or are there multiple different sets of wrong results? Explain your answer. Outline how you would change the code so it always produces the correct result. (You are not required to provide detailed code, just an outline of what you would do. You should keep code modifications to a minimum and your code should still run concurrently.)

[10 marks]

UID:

- (d) Describe what you expect to happen if the code was recompiled and executed with values for Size of 9 and 16 (i.e. two cases).

[6 marks]

Continuation of answer to Question Part

UID:

Continuation of answer to Question Part

Continuation of answer to Question Part