The Australian National University
Mid Semester Examination – August 2016

Comp2310 & Comp6310
Concurrent and Distributed Systems

Study period:	15 minutes
Time allowed:	1.5 hours (after study period)
Total marks:	50
Permitted materials:	None

Questions are **not** equally weighted – sizes of answer boxes do **not** necessarily relate to the number of marks given for this question.

All your answers must be written in the boxes provided in this booklet. You will be provided with scrap paper for working, but only those answers written in this booklet will be marked. Do not remove this booklet from the examination room. There is additional space at the end of the booklet in case the boxes provided are insufficient. Label any answer you write at the end of the booklet with the number of the question it refers to.

Greater marks will be awarded for answers that are simple, short and concrete than for answers of a sketchy and rambling nature. Marks will be lost for giving information that is irrelevant to a question.

Student number:

The following are for use by the examiners

Q1 mark	*Q2 mark*	*Q3 mark*

Total mark

1. [8 marks] General concurrency

(a) [2 marks] What can you say about the temporal relation between two tasks if they are executing concurrently?

(b) [4 marks] Explain two significant and different ways how a task might influence the execution behavior of another task.

(c) [2 marks] Are mutual-exclusion programming methods required even if you know that all concurrent tasks are ultimately executed as a single, strictly sequential stream of machine instructions? Give precise reasons.

2. [33 marks] Communication & Synchronization

(a) [6 marks] 3 tasks all concurrently call a procedure named `Uncritical`, then print out "`|Start-->`", then call a procedure named `Critical` and then print out "`End|`" with the expectation that "`|Start-->End||Start-->End||Start-->End|`" appears on the terminal (as opposed to for instance "`||Start-->End|St|Start-->aEnd|rt-->End|`").
Provide a program (in any programming language which you prefer, including pseudo-code) such that the expected output is guaranteed under all circumstances.
Hint: there are many correct ways to solve this problem, so make a conscious choice before you start writing.

(b) [17 marks] Read the following Ada program carefully. The program is syntactically correct and will compile without warnings. See questions below and on the following page.

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

procedure Message_Chain is

   type Nodes is range 1 .. 5;

   task type Node is
      entry Handover_Id (Assigned_Id : Nodes);
      entry Token;
   end Node;

   Chain : array (Nodes) of Node;

   Sum_A, Sum_B : Natural := 0;

   task body Node is

      Id : Nodes := Nodes'Invalid_Value;

   begin
      accept Handover_Id (Assigned_Id : Nodes) do
         Id := Assigned_Id;
      end Handover_Id;

      Sum_A := Sum_A + 1;

      accept Token;

      Sum_B := Sum_B + 1;

      if Id /= Nodes'Last then
         Chain (Id + 1).Token;
      end if;

      Put_Line ("Task" & Nodes'Image (Id) & " sees at last:"
                              & Natural'Image (Sum_A) & Natural'Image (Sum_B));
   end Node;

begin
   for n in Nodes loop
      Chain (n).Handover_Id (n);
   end loop;

   Chain (Chain'First).Token;
end Message_Chain;
```

(i) [2 marks] How many tasks are implemented by this program? Name them.

(ii) [2 marks] A task could potentially be blocked at multiple operations inside this code. Enumerate those potentially blocking operations.

(iii) [3 marks] Is this program deterministic? Give precise reasons for your answer.

(iv) [4 marks] Will this program terminate always, sometimes, or never? Give precise reasons for your answer. If you think that some tasks will terminate while others won't, then also enumerate those non-terminating tasks.

(v) [6 marks] What output (or multiple possible outputs) would you expect from running this program? If you found the output to be non-deterministic, then do not write out all possible outputs, but provide rules which describe the possible outputs (for instance: "the printed value for `Sum_A` will always be the negative value of the local task id" or "output line *a* will always appear before line *b*"). Give precise reasons for your answers.

(c) **[10 marks] Seven male gangsters are preparing for their next robbery. In order to keep things on a need-to-know basis they do not reveal their names, but refer to each other by colours. Of course everybody wants to be Mr. Black and nobody wants to be Mr. Pink. Here is where the boss comes in and hands out names on a first-come-first-served basis, so the first gangster who asks to be Mr. Black will indeed become Mr. Black.**
The Ada code below (which is syntactically correct and will compile without warnings) shows the activities for each `Gangster` task.
Write the package `Boss_Office` such that no requested name is ever confirmed twice and therefore the output on the screen shows seven different names. While Ada is an obvious choice, you can write this package in any programming language which you see fit incl. pseudocode – as long as the essential structure of your program stays recognizable. Do not focus on syntax details, but on the logical structure of your package. (In slight adaptation of a famous movie.)

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

procedure Shootout is

   type Colours is (Black, White, Blue, Blonde, Orange, Brown, Pink);

   package Office is new Boss_Office (Choices => Colours);

   task type Gangster;
   task body Gangster is

   begin
      for Colour in Colours loop
         if Office.Can_I_be_Mr (Colour) then
            Put_Line ("I am Mr. " & Colours'Image (Colour));
            exit;
         end if;
      end loop;
   end Gangster;

   Gangsters : array (Colours) of Gangster; pragma Unreferenced (Gangsters);

begin
   null;
end Shootout;
```

(answer the question on the following page)

3. [9 marks] Data-parallelism

(a) [9 marks] Consider the function:

$$\text{stdev}(X) = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{X})^2}{n-1}} = \sqrt{\frac{(x_1 - \overline{X})^2 + (x_2 - \overline{X})^2 + \ldots + (x_n - \overline{X})^2}{n-1}}$$

$$\text{where } \overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{x_1 + x_2 + \ldots + x_n}{n} \quad \text{and } X = (x_1, \ldots, x_n)$$

and then answer the following questions:

(i) [6 marks] Could a function like this be implemented with a concurrent program and gain performance compared to a sequential implementation? (Ignore potential overheads of data distribution, creating or destructing tasks.)
If you think that a concurrent implementation could improve performance then how many concurrent tasks would you need for maximal performance?
Give precise reasons for all answers.

(ii) [3 marks] What is the computational time complexity of a sequential implementation? Will the computational complexity of the algorithm change for your concurrent implementation? If so: in what way? Give precise reasons for your answer either way. (Again: ignore potential overheads of data distribution, creating or destructing tasks.)

continuation of answer to question [ ] *part* [ ]

continuation of answer to question [ ] *part* [ ]

continuation of answer to question [ ] *part* [ ]

continuation of answer to question [ ] *part* [ ]