NATIONAL UNIVERSITY OF SINGAPORE

CS2104 — PROGRAMMING LANGUAGE CONCEPTS

(AY2024/2025 SEMESTER 1)

FINAL ASSESSMENT

Time Allowed: **2 Hours**

INSTRUCTIONS

1. This **QUESTION PAPER** contains 20 Questions in 8 Sections, and comprises 10 printed pages, including this page.

2. The **ANSWER SHEET** comprises 10 printed pages.

3. Use a pen or pencil to write your Student Number in the designated space on the front page of the ANSWER SHEET, and shade the corresponding circle completely in the grid for each digit or letter. DO NOT WRITE YOUR NAME!

4. You must submit only the ANSWER SHEET and no other documents. Do not tear off any pages from the ANSWER SHEET.

5. All questions must be answered in the space provided in the ANSWER SHEET; no extra sheets will be accepted as answers.

6. Write legibly with a pen or pencil (do not use red color). Untidiness will be penalized.

7. For multiple choice questions (MCQ), shade in the circle of the correct answer completely.

8. The full score of this assessment is 100 marks.

9. This is an Open-Book assessment. You are allowed to use any printed or written material but no electronic aids.

Section A: Syntax [8 marks]

Consider the following macro definition in Scheme:

```
(define-syntax my-let
 (syntax-rules ()
   [(my-let ((var expr)) body)
    ((lambda (var) body) expr)]))
```

(1) [4 marks]

Explain what happens when the following program runs, and what result the program computes. Clearly describe the execution steps involved to produce the result of the program.

```
(my-let ((x (+ 2 3)))
  (+ x x))
```

10

+ (+23) (+23)
+ (+23) 5
+ 5 5

10

(2) [4 marks]

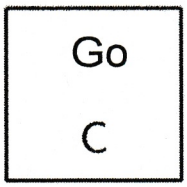
Scheme's macro system prevents variable capture by automatically renaming variables. What would go wrong if variable capture were possible in the above macro definition? Provide a concrete example.

no recursive macro

Section B: T-diagrams [12 marks]

(3) [12 marks]

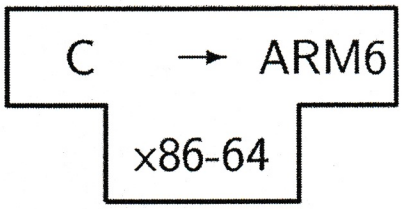
Assume that the following language processors are given: We have an interpreter for Go, written in the language C, and an ARM6 processor.



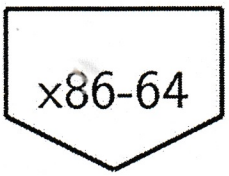
interpreter for Go, written in C

ARM6 processor

We have a compiler that can translate programs written in C to programs written in ARM6 machine code. The compiler is written in x86-64, and we have an x86-64 processor.

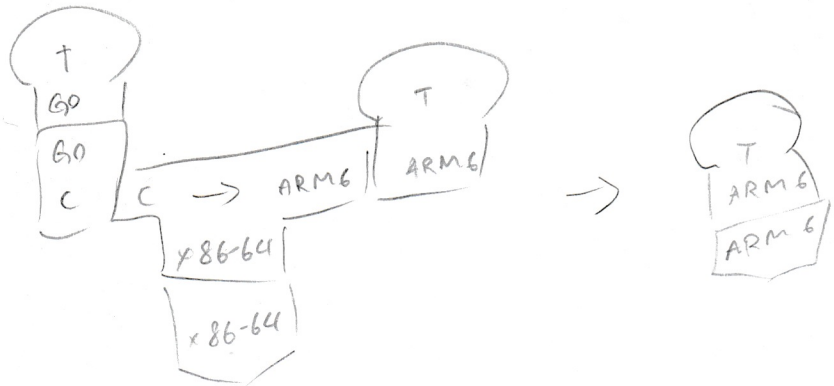


C-to-ARM6 compiler, written in x86-64



x86-64 processor

Give the T-diagrams of the language processing steps that will allow you to run a given Go program "ticketing" on the ARM6 processor.



Section C: CSE Machine [16 marks]

(4) [2 marks]

How many function objects (closures) are created when the following JavaScript program is executed using the CSE machine?

```
const a = x => y => x + y;
const b = x => y => x * y;
const c = x => y => x - y;
[a, b(2), c(3)];
```

A. 3
B. 4
C. 5
D. 6
E. 7
F. 8
G. 9

(5) [2 marks]

How many array data structures are created on the heap ("environment" component) of the CSE machine when the following JavaScript program is executed using the CSE machine?

```
function h(z) {
    return [z, z];
}
function g(y) {
    return h([y, y]);
}
function f(x) {
    return g([x, x]);
}
f([1, 2]);
```

A. 0
B. 1
C. 4
D. 7
E. 10
F. 11
G. 12

(6) [12 marks]

Consider the following JavaScript program. Draw the environment diagram ("E" component of the CSE machine) that results from running the program in the CSE machine, including all function objects and environment frames that get created during the execution.

```
function compose(f, g) {
    return x => f(g(x));
}
const h = compose(y => y + 1,
                  z => z * 3);
h(7);
```

Section D: Memory management [19 marks]

Modern programming languages require implicit heap allocation and deallocation and therefore sophisticated memory management techniques such as garbage collection and reference counting. The language C however is designed for explicit heap allocation and deallocation. Answer each question with up to five full sentences (not counting the examples).

(7) [5 marks]

Describe how C provides heap allocation to the programmer and illustrate your answer with an example.

(8) [6 marks]

Describe two specific technical challenges that programmers face when working with explicit memory **de**allocation in C, and explain how these can lead to program errors. Use concrete examples.

(9) [4 marks]

Modern languages like Python and JavaScript implement functions with heap-allocated closures. What specific restriction in C's function definitions avoids the need for such heap-allocated closures?

(10) [4 marks]

The C programming language provides pointer arithmetic. Explain how this feature interacts with explicit memory management and what risks it introduces. Provide a short example where this feature could lead to memory safety issues.

Section E: Concurrent programming [14 marks]

Consider an event ticketing system written in Java where multiple threads attempt to purchase tickets concurrently. Here is a partial implementation:

```
class TicketSystem {
    private int availableSeats;

    public TicketSystem(int totalSeats) {
        this.availableSeats = totalSeats;
    }
    // Return number of seats successfully purchased
    public int purchaseTickets(int requestedSeats) {
        if (requestedSeats <= availableSeats) {
            // Simulate some processing time
            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {}

            availableSeats -= requestedSeats;
            return requestedSeats;
        }
        return 0;
    }
    public int getAvailableSeats() {
        return availableSeats;
    }
}
```

(11) [4 marks]

This implementation has a race condition. Describe a specific scenario where this race condition could lead to overselling of tickets. Include the initial state, the sequence of thread operations, and the incorrect final state.

(12) [10 marks]

Modify the program to prevent the race condition using synchronized methods. Explain why your solution works.

Section F: Logic programming [10 marks]

Define a relation `interleave (xs, ys, zs)` using the logic programming language presented in the lecture. The relation is true when `zs` is a list formed by alternating elements from `xs` and `ys` in the given order, starting with the first element of `xs`. If one list is longer than the other, the remaining elements should be appended to the end of the result.

Examples:

Query: `interleave(list(1, 2, 3), list("a", "b", "c"), $result)`
Result: `$result = list(1, "a", 2, "b", 3, "c")`

Query: `interleave(list(1, 2), list("a", "b", "c", "d"), $result)`
Result: `$result = list(1, "a", 2, "b", "c", "d")`

Query: `interleave(list(1, 2, 3), list("a"), $result)`
Result: `$result = list(1, "a", 2, 3)`

(13) [5 marks]

Write the rules for the interleave relation.

(14) [5 marks]

Consider your solution for the previous question and the following query:

Query: `interleave($x, $y, list(1, "a", 2, "b"))`

List all possible solutions for `$x` and `$y` in the above query.

1 2 a b

1 a 2 b

1 2 b a

1 a 2 b NIL

NIL 1 a 2 b

Section G: Parallel programming [12 marks]

(15) [4 marks]

Consider the following sequence of assignments that specify a sequential control: Each assignment is executed completely before the next assignment is started.

```
a = x + y
b = y - z
c = a * 2
d = b + 3
e = c + d
```

Organize the assignments into a sequence of groups such that the assignments within each group could be executed in parallel without changing the result of the program.

Briefly explain why the operations you grouped together can run in parallel, while others cannot.

(16) [3 marks]

Consider these execution times (in milliseconds) for multiplying two 4000×4000 matrices:

Single CPU core: 8000ms
8 CPU cores: 1200ms

6.66

Calculate the speedup achieved by the 8-core CPU solution compared to single core. Why is this speedup less than 8x? Explain in 1-2 sentences.

(17) [5 marks]

Consider these execution times (in milliseconds) for multiplying two 4000×4000 matrices:

Single CPU core: 8000ms
GPU (32-bit floating point): 80ms
GPU (16-bit floating point): 40ms

At what matrix size n would you expect the GPU to show NO speedup compared to a single CPU core? Explain your reasoning in 2-3 sentences.

200×200

1×1

Section H: Quantum programming [9 marks]

Consider the following quantum circuit written in Qiskit:

```
qc = QuantumCircuit(2, 2)
qc.h(0)       # Hadamard
qc.cx(0, 1)   # CNOT
qc.measure([0,1], [0,1])
```

(18) [2 marks]

What is the initial state of the two qubits before any gates are applied?

(19) [3 marks]

After the Hadamard gate is applied to qubit 0, what is the state of the system? Explain in one sentence why this state is significant in quantum computing.

(20) [4 marks]

After the CNOT gate is applied, what is the final state of the system? Explain why this state cannot be represented as two separate single-qubit states.

END OF PAPER