Leiden University
Leiden Institute for Advanced Computer Science

Concepts of Programming Languages

F. Arbab
Final Exam

8 January 2016

Student name:

Student number:

This exam consists of 4 pages and 10 questions for a total of 100 points.

Instructions:

- Write your name and student number on this hand-out.
- Verify that your copy of this hand-out is complete and legible.
- This is a closed-book, closed-notes, individual exam.
- You are not allowed to use your laptop, phone, or any other computing or telecommunication device.
- You can work on this exam only within the allocated time-slot.
- Do not unstaple or tear off pages of this hand-out.
- Do not write your answers on this hand-out.
- You must return all pages of this hand-out to the proctor at the end of the exam, regardless of whether or not you have written anything on it.

1. **10 points**
What are the relative advantages disadvantages of implementing a programming language with an interpreter as compared with implementing the language with a compiler? Give at least two advantages for each approach.

2. **10 points**
Consider the following BNF definition:

<S> → <A><A> | c<B><A>
<A> → a<A> | <B> | ε
<B> → c<B> | c

(a) Give a rightmost derivation for the string: ccaa
(b) Show that the BNF definition is ambiguous.

3. **10 points**
Compute the weakest pre-condition for each of the following program fragments and its given post-condition:
(a) a = 3 * (b + a); b = a – 2 { b < 7 }
(b) a = -4 * b – 8 { a > 8 }
(c) if (a > b) b = 4 * a + 1 else b = 4 * a – 5; { b > 0 }
(d) a = 1;
```
y = b * b – 4 * a * c;
if (y >= 0 ) {
    z = sqrt(y);
    x1 = (b + z) / (-2 * a);
    x2 = (b - z) / (-2 * a)
}
```
{ x1 = -7, x2 = 3 }

4. **10 points**
Explain the meaning and purpose of
(a) Operational semantics,
(b) Denotational semantics,
(c) Axiomatic semantics.

5. **10 points**
(a) In the context of sub-program level concurrency, what is a task (or a process)?
(b) What is the difference between heavyweight and lightweight tasks?
(c) Define competition and cooperation synchronization.
(d) Explain how semaphores work.
(e) Explain the concept of message passing rendezvous.

6. **10 points**

Write a Scheme function merge that takes two sorted lists of numbers as arguments and returns the result of merging them into one sorted list. For example, the call

```
(merge '(1 4 7 9) '(3 4 5))
```

should return: (1 3 4 4 5 7 9).

7. **10 points**

Suppose you have access to a compiler for a new strongly typed language X but you do not know whether X uses:

(a) Name type equivalence or structure types equivalence by default. Show a simple test program (fragment), by running which you can determine which notion of type equivalence is used in X, and explain your reasoning.

(b) Dynamic scoping or static scoping. Show a simple test program (fragment), by running which you can determine which notion of scoping is used in X, and explain your reasoning.

(c) Call-by-reference or call-by-value result for parameter passing. Show a simple test program (fragment), by running which you can determine which notion of parameter passing is used in X, and explain your reasoning.

(d) A left-to-right or a right-to-left order for evaluating the operands in an expression. Show a simple test program (fragment), by running which you can determine which order of evaluation is used in X, and explain your reasoning.

8. **10 points**

Consider the following program:

```
#include <iostream.h>

void foo(int a, int b) {
        a = 3;
        a = a + b;
        b = a + 2;
}

int main() {
        int x = 5;
        foo(x, x);
        std::cout << x << std::endl;
}
```

What output does this program produce under each of the following parameter passing mechanisms? (Explain your reasoning).

(a) call-by-value

(b) call-by-reference.

9. 10 points

Consider the following program:

```
#include <iostream>
int x = 3, y = 5;

void bar(void) {

        y = y + 2;
        std::cout << x << " " << y << std::endl;
}

void baz(void) {
        int y = 7;
        x = x + 2;
        bar();
        std::cout << y << std::endl;
}

int main() {
        int x = 5;
        x = x + 2;
        baz();
        std::cout << x << std::endl;
}
```

What output does this program produce under each of the following scoping rules?
(a) Static scope
(b) Dynamic scope

10. 10 points

Write a Prolog relation subset that determines whether its first argument (a list representing a set) is a subset of its second argument (also a list representing a set). For example, the queries

```
subset([ ], [3, 4]).
subset([4, 3, 4], [4, 3, 2, 1]).
```

should both succeed, and the query

```
subset([4, 3, 1], [4, 3, 2]).
```

should fail.