University of Toronto
Faculty of Arts and Science

April 2015 Examinations

CSC488H1S / CSC2107HS

Duration – 2 hours (120 minutes)

OPEN BOOK ALL written aids, books and notes are allowed.
ALL non-programmable calculators allowed.
NO other electronic aids allowed.

120 marks total, 8 Questions on 4 Pages. ANSWER ALL QUESTIONS
Write all answers in the Exam book.

You must receive a mark of 35% or greater on this final exam to pass the course.

WRITE LEGIBLY Unreadable answers cannot be marked.
Line/rule reference numbers on the left side of of programs and grammars are provided for ease of reference only and are not part of the program or grammar .
The notation **. . .** stands for correct code that has been omitted for brevity
State clearly any assumptions that you have to make to answer a question.

1. [10 marks] The grammar shown below is LL(2).

1	A	$\rightarrow$	a B a a
2		$\rightarrow$	b C b a
3	B	$\rightarrow$	b
4		$\rightarrow$	λ
5	C	$\rightarrow$	b
6		$\rightarrow$	λ

Where λ is the empty string

a) Show the *First* and *Follow* sets for the non terminals A, B and C.

b) Show the *Predictor Sets* for this grammar.

2 . [20 marks] Consider the program fragment shown below.

```
20      integer J, K, L, N ;
21      real A [ 1 .. 100 , 1 .. 101 ] ;
        /* reals use 4 bytes of storage */

        /* Assume A is given a value here */

30      N := 100
31      for J := 1 to N - 1 do
32          for K := J + 1 to N do
33              for L := J + 1 to N + 1 do
34                  A[ K - 1 , L ] := A[ K , L ] * A[ J , J] / A[ K , J ] - A[ J , L - 1] ;
35              end
36              A[ K , J ] := 0.000000 ;
37          end
38      end
```

Describe the classical machine independent optimizations that an optimizing compiler would perform on the code fragment on lines 30 to 38

3. [15 marks] Describe the semantic analysis checks that a compiler would perform on the code fragment in Question 2 .

4. [10 marks] During the course project two changes were made to the definition of anonymous functions in the project language.

1. **exit** and **exit when** can not be used in the statement part of an anon function unless they are inside a while or loop construct.
2. Both forms of the **return** statement cannot be used in the statement part of an anon function unless they are inside a procedure or function.

Explain why each of these changes simplified the implementation of the project language.
Give examples of problems that would arise if these changes had not been made.

5. [20 marks]Some programming languages include a *tagged* version of the case / union data structure. For example in Modula-2:

```
1     type shapeKind = (square, rectangle, circle);
2     type shape  =
3         record
4             centerx : integer ;
5             centery : integer ;
6             case kind : shapeKind of
7                 square : ( side : integer ) ;
8                 rectangle : ( length, height : integer ) ;
9                 circle : ( radius : integer ) ;
10        end ;
```

The record type *shape* can be used to create variables like any other type. The case tag *kind* indicates which alternative of the case is currently active.
Languages with a tagged case usually enforce strict access control to the alternatives of a case. For example:

Any attempt to access a member of the data structure whose existence depends on a particular value of the case tag, while the case tag is not the expected one, raises an run time error.

For example trying to use the value of *radius* when *kind* has the value *square*.

Describe the run time mechanisms necessary to enforce the restrictions on access to fields in the tagged data structure.

6. [15 marks] Given the declarations:

```
boolean P, Q, R[ 20 ]
integer J
```

Using the quadruples from the lecture notes show the quadruples that would be generated for the statement:

```
R[ J ] <=  ! P | ( Q & ! R[ J -1 ] ) & ! ( Q | ! P )
```

7. [10 marks] In the ANSI C programming language the definition for floating point constants is:

- a floating-point constant can take one of two forms:
  - a group of digits from 0 to 9 which includes a single period, like 3.141593, 0. or .1
  - a group of digits from 0 to 9, possibly containing a period, followed by 'e' or 'E' and an exponent, like 1.0e3, 2.593e+7 or 3141593e-6
- a floating-point constant may be followed by: an 'f' or 'F' to indicate a float constant, like 35.79f Note that 123f is an error, because 123 is not a floating-point constant

Write one regular expression that could be used to recognize floating point constants in ANSI C.

8.[20 marks] A proposal has been made to extend the project language to allow arrays in procedures and functions to have a dynamic size, i.e. the dimensions of the array are determined by runtime *expressions*. Example:

```
 1    procedure flexi ( integer N, integer M )
 2        begin
 3            integer A[ N , 15 ] , B[ N + 3 , M - 1 ]
 4            integer J , K
              % some code here
13            loop
14                get J , K
15                boolean P [ J , K ]
                  % more code here
23            end
              % more code here
47        end       % flexi
```

1. Describe the runtime mechanisms necessary to implement this proposed feature.
2. You still have a chance to influence the language designer over this proposal. What changes would you request to make the proposal easier to implement?

Total Marks = 120