

**CSI 3125**  
**Fall 1999 Midterm**  
Professor: Ken Barker

Monday, November 1, 14:30

<i>Family Name</i>	
<i>Given Name</i>	
<i>Student Number</i>	

**Notes:**

1. This is a closed book midterm. Textbooks, notes, cheat sheets and microfiche are **not** allowed.
2. Calculators, computers, Palm V's and all other computing devices are **not** allowed.
3. There are 10 pages. Write your name and student number on **every** page.
4. There are 6 questions. Answer **all** 6 of them.
5. Write **all** answers and work in the space provided. Use **no** other paper.
6. You have 80 minutes to complete the midterm.

**Marks:**

Question						Total (25 marks)
1 (7 marks)	2 (4 marks)	3 (3 marks)	4 (5 marks)	5 (6 marks)	6 (1 mark)	

**Question 1: Extended BNF Grammars**

a) Write an extended BNF (EBNF) grammar for the language of freight trains. Here is some information about freight trains:

- the terminal symbols are:  
engine   cattlear   tanker   fridgecar   lumbercar   caboose
- the goal symbol is:  
<train>
- every train starts with one or two engines and ends with a caboose
- a tanker must not come right after an engine
- a lumbercar must not come right after an engine
- sequences of cattlear must be preceded and followed by a lumbercar

Here are some legal sentences in the language:

- engine fridgecar lumbercar lumbercar caboose
- engine engine caboose
- engine fridgecar lumbercar tanker tanker tanker caboose
- engine fridgecar lumbercar cattlear cattlear lumbercar caboose

b) Here is an EBNF grammar for some language:

```
<S> ::= <A> <B>
<A> ::= [ <C> ] <D>
<B> ::= <H> [ <A> ]
<C> ::= d
<D> ::= { <E> } <F>
<E> ::= j
<F> ::= n [ <G> <A> ]
<G> ::= p
<H> ::= v
```

Show a parse tree for the sentence dj jnvj npdn

## Question 2: Names, Binding, Scope

Here is a Pascal program that does nothing of interest

```
program main;

var X, Y: integer;

  procedure foo(A: integer);
  var X: integer;

  begin
    X := 10;
    bar(X)
  end;

  procedure bar(foo: integer);
  var Z: integer;

    procedure foobar(X: integer);
    begin
      Z := X * foo
    end;

    begin
      foobar(foo);
      X := Z
    end;

begin

  X := 5;
  Y := 13;

  foo(Y)

end.
```

a) Give the *scope* of the following names:

main.X \_\_\_\_\_

main.foo \_\_\_\_\_

main.bar.foo \_\_\_\_\_

main.bar.foobar.X \_\_\_\_\_

b) Give the *referencing environment* for the following statements.

$S(\textit{procedurename})$  refers to the statements in procedure *procedurename*.

S(main) \_\_\_\_\_

S(main.foo) \_\_\_\_\_

S(main.bar) \_\_\_\_\_

S(main.bar.foobar) \_\_\_\_\_

### Question 3: Data Types

The following Pascal program is perfectly correct given the proper variable declarations. Show the variable declarations necessary to make the program correct.

```
program p;  
  
  var t: _____ ;  
      u: _____ ;  
      v: _____ ;  
      w: _____ ;  
      x: _____ ;  
      y: _____ ;  
      z: _____ ;  
  
begin  
  ...  
  w := x * y;  
  u := chr(z + 3);  
  v := u in w;  
  t := ord(v)  
  
end.
```

### Question 4: Parameter Passing

Show the output (if any) for the following Pascal-style program for each of the five parameter passing modes. Use the table below to show your answers.

```

program main;

var X, i: integer;
    A: array[1..5] of integer;

procedure p(mode U, V: integer);
var i: integer;

begin
    U := U + V;
    V := U;

    write(U); write(', '); writeln(V)
end;

begin
    X := 1;
    for i := 1 to 5 do
        A[i] := 5 - i;

    p(X, A[X]);

    write(X); write(', '); writeln(A[X])
end.

```

***parameter passing mode***    ***program output***

<i>pass-by-value</i>	
<i>pass-by-result</i>	
<i>pass-by-value-result</i>	
<i>pass-by-reference</i>	
<i>pass-by-name</i>	

## Question 5: Subprogram Implementation

Here is yet another useless Pascal program

```
L01      program main;

L02      var X: integer;
L03          B: boolean;

L04          function f(W: integer): integer;
L05      begin
L06          if B then begin
L07              B := false;
L08              f := f(W + 1)
L09          end
L10          else begin
L11              write(W);
L12              f := W
L13          end
L14      end;

L15      procedure p(U: integer);
L16      var X, i: integer;

L17      begin
L18          X := U;

L19          for i := 1 to 4 do
L20              X := X * U;

L21              write(f(X))
L22          end;

L23      begin
L24          B := true;
L25          X := 2;
L26          p(X);
L27          writeln(X)
L28      end.
```

Using the empty stack on the next page, draw the complete activation stack when IP = L12. Twenty-five rows in the stack should be *more* than enough.



Name: \_\_\_\_\_

Student Number: \_\_\_\_\_

S01	<i>(dynamic link)</i>	(main)
S02	<i>(static link)</i>	
S03	<i>(return address)</i>	
S04	<i>(return value)</i>	
S05		
S06		
S07		
S08		
S09		
S10		
S11		
S12		
S13		
S14		
S15		
S16		
S17		
S18		
S19		
S20		
S21		
S22		
S23		
S24		
S25		

**Question 6: Concepts of Programming Languages**

1. The term *orthogonality* refers to
  - a) a kind of abstraction in which the same symbol is used for different operations
  - b) the extent to which constructs in a language can be combined freely
  - c) the incompatibility between row-major and column-major arrays
  - d) the use of prostheses for amputated limbs
2. A *non-terminal* symbol
  - a) is a name that cannot be resolved by the compiler due to a circular reference
  - b) is a symbol in a grammar that is *not* composed of sequences of other symbols
  - c) appears on the left side of production rules
  - d) is used for public washrooms in museums and theatres, but not bus stations
3. Which of the following is a *primitive type* in most languages?
  - a) float
  - b) enum
  - c) pointer
  - d) CSI 3125 prof
4. *Row-major* and *column-major* refer to
  - a) conventions for mapping multi-dimensional arrays to one-dimensional memory
  - b) conventions for determining whether two arrays are compatible
  - c) conventions for the Baton Twirlers Association
  - d) two different designations of the Bachelor of Arts degree
5. A *dangling reference*
  - a) is prohibited in strongly-typed languages
  - b) may go undetected by a Pascal compiler
  - c) is the sort of nonsense up with which I will not put
  - d) can get you arrested in this town
6. *Control statements*
  - a) prevent the evaluation of the rest of an expression once the value of the expression has been unambiguously determined
  - b) determine the order of execution of statements in a program
  - c) were often overheard due to Cone of Silence malfunctions
  - d) or the government will control them for you!
7. Yashin should
  - a) swallow it and hook up with the Sens on their next road trip
  - b) listen to his agent, 'cause agents are really smart
  - c) seriously consider a U of O degree: The right choice. The right University.
  - d) come out Sunday nights... it's only 10 bucks!
8. This exam
  - a) bites
  - b) stinks
  - c) rocks
  - d) grooves, daddy-o