Name____Answer Key___________________________

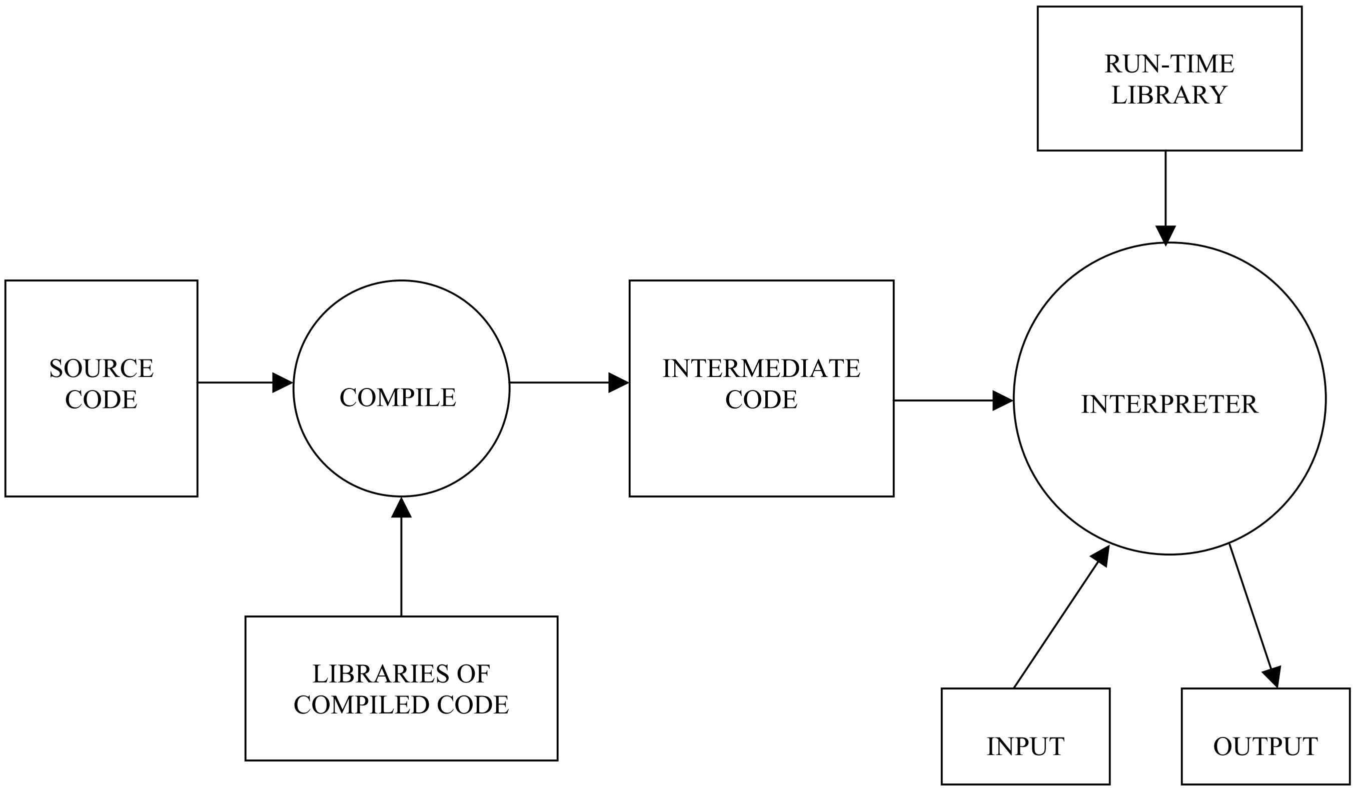
CS471, Programming Language Structure
Spring, 2001

Examination 1

Closed Book. Answer all the questions. The total points for each question or part of a question follows it in parentheses, thus: *(12)*

1

A hybrid language system (such as used in Java) includes elements of both compilation and interpretation. Identify these elements by drawing a diagram showing their relationships. *(16)*



2

What is the problem or problems with the following C++ code? *(6)*

```
struct s {
  char name[100];
  float x;
  float y;
};

void Pig(int a, int b) {
  struct s *p;
  for (int n = 0; n < 100; n++)
     p = new s;
}
```

Give *brief* descriptions, in a few sentences each, of two techniques that are used in language implementations for avoiding this problem. There is no need to alter the source code above. *(5 each)*

There are really two related problems. The first is that the loop creates 100 structures on the heap, using the new operator, but each it replaces the pointer to the previous one with a new pointer to the new structure. Each structure is thus lost since it has no pointer to it any more. This is *memory leakage*. Then, at the end of the procedure, the pointer variable p is destroyed, leaving even the last structure created unreachable. This is also *memory leakage.*

There are two possible solutions:

1. To compile code at the end of every procedure to reclaim space allocated during the procedure when the pointers pointing to that space are just about to be destroyed (the Ada solution). [This may solve the second problem, but not the first].
2. To use a garbage collector that will collect all unused but still allocated space at periods during the program execution. [This will solve both problems].

3

Consider the following grammar:

```
<num> ::= <dig-seq> | <dig-seq> . <dig-seq> | <dig-seq> .
<dig-seq> ::= <digit> | <dig-seq> < digit>
<digit> ::= 0|1|2|3|4|5|6|7|8|9
```

Which of the following can be generated by the grammar? (6)

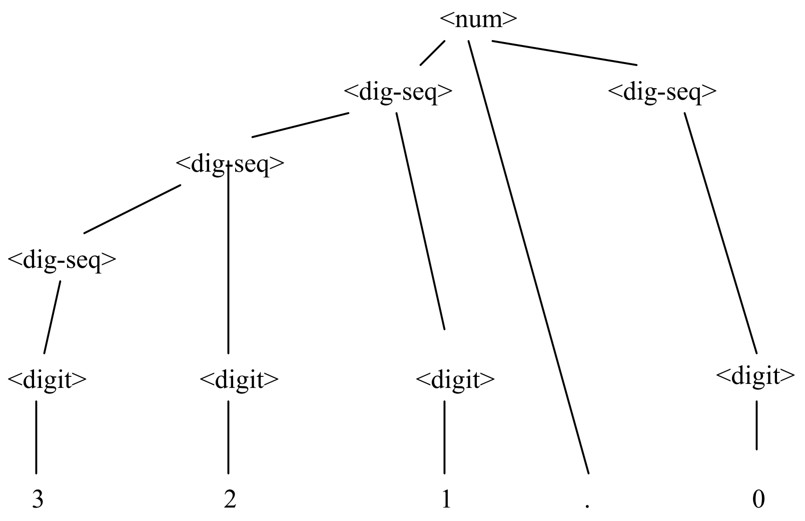
a) 123,456

b) .789

c) 321.0

Draw a parse tree for the one you chose. *(10)*

The only that can be generated is c. It's parse tree is:



4

Consider the following program, written in Pascal syntax:

```
program P;
  procedure A;
    var x : integer;
    procedure B;
      procedure C;
      begin
        print(x)
      end;
    begin
      C
    end;
    procedure D;
      var x : integer;
      begin
        x := 2;
        B
      end;
   begin
     x := 1;
     D
   end;
begin
  A
end.
```

What does the program print:

a) with static scoping? *(8)*

b) with dynamic scoping? *(8)*

If the variables have different names can the program print different results with different scope rules? *(2)*

Static scoping: the program prints 1

Dynamic scoping: the program prints 2

If the variables have different names, the program cannot print different results.

5

What mode of parameter passing is necessary in the swap procedure, which swaps the values in two variables? *(6)*

Write two versions in C/C++ that:

a) Use explicit pointers to implement a pass-by-reference mechanism. *(6)*

b) Use reference parameters to implement pass-by-reference. *(6)*

Write example calls to the procedure in each case.

The mode is in-out.

Explict pointers:

```
void swap(int *a, int *b) {
  int temp = *a;
  *a = *b;
  *b = temp;
}
```

Example call:

```
int x = 1, y = 2;
 swap(&x, &y);
```

Reference parameters:

```
void swap(int &a, int&b) {
  int temp = a;
  a = b;
  b = temp;
}
```

Example call:

```
int x = 1, y = 2;
swap(x, y);
```

6

Rewrite the switch statement in C, below, using only the three structured programming constructs: sequence (compound statement), selection (if-then-else), and iteration (the while loop). *(16)*

```
switch (x) {
  case 0:
  case 1: x = 3; print(x);
  case 2: x = 4; print(x); break;
  default: x = 5; print(x);
}
```

```
if (x == 0 || x == 1) {
  x = 3;
  print(x);
  x = 4;
  print(x);
}
else if (x == 2) {
  x = 4;
  print(x);
}
else {
  x = 5;
  printf(x);
}
```

There are other ways to do this, so look for versions without the || operator, and handling the drop through from case 1 to case 2 differently.