## <Homework>

1. Some programming languages are typeless. What are the obvious advantages and disadvantages of having no types in a language?

```
x = 1;
y = 3;
z = 5;
def sub1():
   a = 7;
   y = 9;
   z = 11;
. . .
def sub2():
   global x;
   a = 13;
   x = 15;
   w = 17;
   . . .
   def sub3():
      nonlocal a;
      a = 19;
      b = 21;
      z = 23;
      . . .
      . . .
```

indentation

2. Consider the following JavaScript program: List all the variables, along with the program units where they are declared, that are visible in the bodies of sub1, sub2, and sub3, assuming static scoping is used.

```
var x, y, z;
function sub1() {
  var a, y, z;
  function sub2() {
    var a, b, z;
    . . .
  }
  . . .
}
function sub3() {
var a, x, w;
. . .
}
```

3. Consider the following Python program. List all the variables, along with the program units where they are declared, that are visible in the bodies of sub1, sub2, and sub3, assuming static scoping is used.

4. Consider the following skeletal C program. Given the following calling sequences and assuming that dynamic scoping is used, what variables are visible during execution of the last function called? Include with each visible variable the name of the function in which it was defined.
   a. main calls fun1; fun1 calls fun2; fun2 calls fun3.
   b. main calls fun1; fun1 calls fun3.
   c. main calls fun2; fun2 calls fun3; fun3 calls fun1.
   d. main calls fun3; fun3 calls fun1.
   e. main calls fun1; fun1 calls fun3; fun3 calls fun2.
   f. main calls fun3; fun3 calls fun2; fun2 calls fun1.

```
void fun1(void); /* prototype */
void fun2(void); /* prototype */
void fun3(void); /* prototype */
void main() {
int a, b, c;
. . .
}
void fun1(void) {
  int b, c, d;
  . . .
}
void fun2(void) {
  int c, d, e;
  . . .
}
void fun3(void) {
  int d, e, f;
  . . .
}
```

5. Consider the following program, written in JavaScript-like syntax. Given the following calling sequences and assuming that dynamic scoping is used, what variables are visible during execution of the last subprogram activated? Include with each visible variable the name of the unit where it is declared.
   a. main calls sub1; sub1 calls sub2; sub2 calls sub3.
   b. main calls sub1; sub1 calls sub3.
   c. main calls sub2; sub2 calls sub3; sub3 calls sub1.
   d. main calls sub3; sub3 calls sub1.
   e. main calls sub1; sub1 calls sub3; sub3 calls sub2.
   f. main calls sub3; sub3 calls sub2; sub2 calls sub1.

```
// main program
var x, y, z;
function sub1() {
  var a, y, z;
  . . .
}
function sub2() {
  var a, b, z;
  . . .
}
function sub3() {
  var a, x, w;
  . . .
}
```