

1. Consider the following JavaScript skeletal program:

```
// The main program
var x;
function sub1() {
    var x;
    function sub2() {
        ...
    }
}
function sub3() {
    ...
}
```

Assume that the execution of this program is in the following unit order:

main calls sub1

sub1 calls sub2

sub2 calls sub3

- a. Assuming static scoping, in the following, which declaration of x is the correct one for a reference to x?
 - i. sub1
 - ii. sub2
 - iii. sub3
- b. Repeat part a, but assume dynamic scoping.

2. Assume the following JavaScript program was interpreted using static-scoping rules. What value of x is displayed in function sub1? Under dynamic-scoping rules, what value of x is displayed in function sub1?

```
var x;
function sub1() {
    document.write ("x = " + x + "");
}
function sub2() {
    var x;
    x = 10;
    sub1();
}
x = 5;
sub2();
```

3. Consider the following JavaScript program:

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

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 program, written in JavaScript-like syntax:

```
// main program
var x, y, z;

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

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 visible variable the name of the unit where it is declared.

- main calls sub1; sub1 calls sub2; sub2 calls sub3.
- main calls sub1; sub1 calls sub3.
- main calls sub2; sub2 calls sub3; sub3 calls sub1.
- main calls sub3; sub3 calls sub1.
- main calls sub1; sub1 calls sub3; sub3 calls sub2.
- main calls sub3; sub3 calls sub2; sub2 calls sub1.

5. Consider the following skeletal C program:

```
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;
    ...
}
```

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.