

1. 다음 식별자 형태들 중에서 어느 것이 가장 좋은 판독성을 갖는가? 그 이유를 설명하라.

1) FirstName 2) First_Name 3) firstname

2. 어떤 프로그래밍 언어는 정적 변수를 가진다. 정적 변수를 가지는 언어의 명백한 장단점은 무엇인가?

3. 여러분이 알고 있는 언어로 한 개의 산술 연산자를 포함하는 단순 배정문을 작성하라. 문장이 실행될 때, 문장의 각 구성 요소에 대해서 그 의미를 결정하는 데 요구되는 다양한 바인딩을 나열하라. 각 바인딩에 대해서 그 언어에 대해 사용된 바인딩 시간을 나타내라.

4. 동적 타입 바인딩은 묵시적 힙-동적 변수와 밀접한 연관이 있다. 이 관계를 설명하라.

5. 동적 영역이 프로그램을 읽기 힘들게 만드는 상황을 묘사하라.

6. 다음 JavaScript 골격 프로그램을 생각해보자.

// main 프로그램

var x;

function sub1() {

var x;

function sub2() {

 ...

 }

}

function sub3() {

 ...

}

위의 프로그램 실행이 다음의 순서로 진행된다고 가정하라.

main이 sub1을 호출한다. sub1이 sub2를 호출한다. sub2가 sub3을 호출한다

a. 정적 영역을 가정하고서, 다음 함수에서 x 에 대한 참조를 하면, x 의 어느 선언을 의미하나?

i. sub1

ii. sub2

iii. sub3

b. 동적 영역을 가정하고서, a의 문제를 다시 해결하라.

7. 다음 JavaScript 프로그램이 정적-영역 규칙을 사용하여 해석된다고 가정한다. 함수 sub1에서 x 의 어떤 값이 출력되는가? 동적-영역 규칙을 사용하면, 함수 sub1에서 x 의 어떤 값이 출력되는가?

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

8. 다음 JavaScript 프로그램을 생각해보자.

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

```

function sub3() {
    var a, x, w;
    ...
}

```

정적 영역이 사용된다고 가정하고, sub1, sub2, sub3의 각 몸체에서 가시적인 모든 변수들을 그 변수가 선언된 프로그램 단위와 함께 나열하라.

9. 다음 Python 프로그램을 생각해보자.

```

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;
        ...
    ...

```

정적 영역이 사용된다고 가정하고, sub1, sub2, sub3의 각 몸체에서 가시적인 모든 변수들을 그 변수가 선언된 프로그램 단위와 함께 나열하라.

10. 다음 C 프로그램을 생각해보자.

```
void fun(void) {
    int a, b, c; /* 정의 1 */
    ...
    while (...) {
        int b, c, d; /* 정의 2 */
        ... <----- 1
        while (...) {
            int c, d, e; /* 정의 3 */
            ... <----- 2
        }
        ... <----- 3
    }
    ... <----- 4
}
```

위 함수에서 화살표로 지시된 4개 지점 각각에 대해, 각 가시적인 변수와 이 변수를 정의하는 정의문의 번호를 함께 나열하라.

11. 다음 C 골격 프로그램을 생각해보자.

```
void fun1(void); /* 원형 */
void fun2(void); /* 원형 */
void fun3(void); /* 원형 */
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;
    ...
}

```

호출 순서가 다음과 같이 주어져 있고 동적-영역 규칙이 사용된다고 가정하면, 마지막으로 호출된 함수가 실행되는 동안에 어떤 변수가 가시적인가? 각 가시적인 변수에 대해 그 변수가 정의된 함수 이름을 포함시켜라.

- a. main은 fun1을 호출; fun1은 fun2를 호출; fun2는 fun3를 호출.
- b. main은 fun1을 호출; fun1은 fun3를 호출.
- c. main은 fun2를 호출; fun2는 fun3를 호출; fun3는 fun1을 호출.
- d. main은 fun3를 호출; fun3는 fun1을 호출.
- e. main은 fun1을 호출; fun1은 fun3를 호출; fun3는 fun2를 호출.
- f. main은 fun3를 호출; fun3는 fun2를 호출; fun2는 fun1을 호출.

12. JavaScript-유사 구문으로 작성된 다음 프로그램을 생각해보자.

```

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

```

}

호출 순서가 다음과 같이 주어져 있고 동적-영역 규칙이 사용된다고 가정하면, 마지막 활성화된 부프로그램이 실행되는 동안에 어떤 변수가 가시적인가? 각 가시적인 변수에 대해 그 변수가 정의된 함수 이름을 포함시켜라.

- a. main은 sub1을 호출; sub1은 sub2를 호출; sub2는 sub3를 호출.
- b. main은 sub1을 호출; sub1은 sub3를 호출.
- c. main은 sub2를 호출; sub2는 sub3를 호출; sub3는 sub1을 호출.
- d. main은 sub3를 호출; sub3는 sub1을 호출.
- e. main은 sub1을 호출; sub1은 sub3를 호출; sub3는 sub2를 호출.
- f. main은 sub3를 호출; sub3는 sub2를 호출; sub2는 sub1을 호출.