

객체지향프로그래밍 퀴즈 #1 / E12 해답

강대기

2007년 11월 2일

다음 질문들에 대해 서술형으로 필요한 내용을 충분하면서도 명확하고 군더더기 없이 답하시오.

1. 다음의 프로그램은 무엇을 출력하는가?

```
#include <cstdio>
int main ()
{
    int i(0),a[3];

    a[i] = i++;
    printf ("%d",a[i]);
}
```

이러한 문제는 C/C++ 표준에는 정의되어 있지 않으므로 근본적으로 어떤 값이 출력될지는 예측불허이다. Visual C++ 8.0 의 경우, i가 처음에 0으로 초기화되었으나, a[0]에 0을 집어넣은 후, i값이 하나 증가한다. 출력하려는 a[1]의 값은 초기화되어 있지 않으므로 예측불허가 된다.

2. #define 대신 const를 사용할 때의 장점을 세 가지 적어라.

- 데이터 형을 명시적으로 저장할 수 있음
- C++의 사용 범위 규칙을 사용하여 그 정의를 특정 함수나 파일에 서만 사용하게 제한 가능함
- 배열/구조체같은 복잡한 데이터 형에도 사용 가능함

3. 정수형 (int)으로 숫자를 하나 입력받아 10진수, 16진수, 그리고 8진수로 출력하는 프로그램을 작성하라.

답으로 작성할 코드의 한 예은 다음과 같다.

```
#include <iostream>
using namespace std;
```

```
int main()
{
    cout << "Enter a number :";
    int number;
    cin >> number;
    cout << dec << number << endl;
    cout << hex << number << endl;
    cout << oct << number << endl;
    return 0;
}
```

4. 정수(int) 200 개를 원소로 가지는 배열을 선언하되, (선언하는 동시에) 처음 5 개의 원소는 1을 가지고, 나머지는 0으로 초기화 하라. 반드시 한 문장의 명령문으로 이 일을 수행하라.

답으로 작성할 코드의 한 예은 다음과 같다.

```
int numbers[200] = {1,1,1,1,1};
```

5. 프로그램의 한 부분에서, 다음과 같이 이름을 문자열로 입력받는 경우, 발생할 수 있는 문제점을 세 가지 적어라.

```
...
const int ArSize = 20;
char name[ArSize];
cout << "이름을 입력하세요 :";
cin >> name;
...
```

- cin 자체는 화이트 스페이스 (white space)를 만나면 입력을 멈추므로, 예를 들어 공백이나 탭이 들어간 입력이 불가능하다.
 - 입력이 복사될 배열 name의 크기는 20 바이트에 불과하지만, cin은 화이트 스페이스가 없는 동안 계속 입력을 받아서 복사한다. 따라서 name 배열의 범위를 넘겨서 문자열이 복사될 수도 있다. 이러한 buffer overflow 문제는 심각한 보안 상의 결함을 야기할 수도 있다.
 - 만일 바로 앞에서 숫자를 cin으로 입력 받았을 경우, 개행 문자가 남아있으므로, 개행 문자가 입력되고 바로 입력이 끝난다. 입력 명령 전에 cin.clear()를 수행하고, 언제나 입력을 검사하는 루프를 만들어야 한다.
6. 다음의 각각의 C++ 코드들 중 위험한 것은 “위험하다”, 위험하지 않은 것은 “괜찮다”라고 답하고, 위험한 경우 그 이유를 서술하라.

각각의 코드에 필요한 헤더 파일들(iostream, string, cstring)은 이미 위에서 #include 문으로 포함되었다고 가정한다. 또한 각각의 코드들은 서로

독립적이다. (즉, 같은 프로그램 내의 코드들이 아니라, 서로가 다른 프로그램 내의 코드들이다.)

- (a)

```
long* fellow = new long;
*fellow = 233233;
std::cout << *fellow << std::endl;
```

아무 문제 없다. 다만 fellow가 가리키는 객체를 new로 생성했으므로 나중에 delete fellow;를 해야 할 것이다.
- (b)

```
long* fellow = new long[10];
*fellow = 233233;
cout << *fellow << endl;
```

아무 문제 없다. fellow는 10 개의 배열의 가장 처음 원소를 가리키고 있다. new로 생성했으므로 나중에 delete[] fellow;를 해야 할 것이다.
- (c)

```
char bird[5]="sparrow";
```

위험하다. bird는 5 바이트까지를 넣을 수 있는 배열이다.
- (d)

```
std::cout << "H" "e" "l" "l" "o" << std::endl;
```

아무 문제 없다. 이런 경우 문자열 상수들끼리는 서로 자동으로 연결된다.
- (e)

```
int index(10);
std::cout << index << std::endl;
```

아무 문제 없다. C++에서 초기화하는 방법 중 하나이다. 10이 출력된다.
- (f)

```
string str;
str = "Doneseo";
std::cout << str << std::endl;
```

아무 문제 없다. string 객체는 =연산자가 오버로딩되어 있어 대입이 가능하다.
- (g)

```
char* str;
str = "Doneseo";
std::cout << str << std::endl;
```

아무 문제 없다. 문자열 상수 "Doneseo"의 시작 주소가 str 포인터에 대입된다.
- (h)

```
char* str;
strcpy(str,"Doneseo");
std::cout << str << std::endl;
```

위험하다. 초기화되지 않은 포인터는 어떤 번지를 가리키고 있는지 알 수 없음에도, 거기에 문자열을 strcpy로 복사하려 하고 있다.
- (i)

```
string str;
strcpy(str,"Doneseo");
std::cout << str << std::endl;
```

위험하다. strcpy 함수는 char *가 오기를 기대하므로, string 객체는 받아들이지 않는다. 아예 컴파일조차 되지 않는다.

```
(j) cout << "Dongseo"[4] << endl;
```

아무 문제 없다. “Doneseo”의 네번째 원소인 ‘s’가 출력된다.

7. 다음 중 문제가 발생할 수 있는 행은 어느 행인가? 문제가 발생할 소지가 있는 행이 여러 개인가? 문제가 있는 행마다 그 행 번호(line number)를 쓰고 이유를 간단히 서술하라. 물론 왼쪽의 숫자는 프로그램 코드가 아니라 행 번호이다.

```
01: int* ps = new int;
02: delete ps;
03: delete ps;
04: int jugs = 5;
05: int* pi = &jugs;
06: delete pi;
```

3행에서 delete를 두번 수행했다. 그 결과는 예측 불허이다. 또한 6행에서는 new로 할당되지 않은 메모리를 delete로 해제하려 했다. 이것 역시 허가되지 않은 행동으로 그 결과는 예측불허이다.

8. 다음과 같은 명령문이 하는 역할은 무엇인가?

```
using namespace std;
```

std 이름 공간에 정의되어 있는 이름들을 프로그램이 사용할 수 있게 허용한다.

9. 물고기에 대한 구조체 형(struct type)을 선언하라. 이 구조체 형에는 20 바이트의 C 스타일의 문자 배열(char kind[20])로 된 물고기의 종류(kind), 정수(int)로 된 그램(gram) 단위의 무게(weight), 소수부(double)가 있는 센티미터(cm) 단위의 길이(length)로 정의해야 한다. 즉 구조체 형의 멤버 변수의 개수는 세 개이다.

답으로 작성할 코드의 한 예은 다음과 같다.

```
struct Fish
{
    char kind[20]; // 물 고 기의 종 류
    int weight; // 그 램 단위의 무 게
    double length; // 센티 미터 단위의 길 이
};
```

10. 위에서 정의한 구조체 선언을 이용하여, 동적 구조체 배열을 생성하라. (정적 구조체도 아니고, 동적 구조체이며, 동적 구조체 변수 하나가 아니라, 동적 구조체의 배열이다.) 생성한 동적 구조체 배열은 다섯 개의 구조체를 원소로 가지게 한다. 각 원소는 ‘도미’, ‘우럭’, ‘광어’, ‘도다리’ 그리고 ‘장어’로 초기화하라. 무게와 길이는 다음의 표를 참조한다.

답으로 작성할 코드의 한 예은 다음과 같다.

```

Fish *fish = new Fish[5];

strcpy(fish[0].kind,"도미");
fish[0].weight = 500; // fish[0] == (*fish) == fish->
fish[0].length = 30.0;

strcpy(fish[1].kind,"우럭");
fish[1].weight = 400;
fish[1].length = 20.0;

strcpy(fish[2].kind,"광어");
fish[2].weight = 1000;
fish[2].length = 60.0;

strcpy(fish[3].kind,"도다리");
fish[3].weight = 490;
fish[3].length = 30.0;

strcpy(fish[4].kind,"장어");
fish[4].weight = 600; // fish[4] == (*(fish+4)) == (fish+4)->
fish[4].length = 60.0;

```

주석문에서 보인 바와 같이 fish[0]는 (*fish), 또는 fish->으로 바꿔 쓸 수 있다. 더 일반화하면 fish[i] 는 (*(fish+i)), 또는 (fish+i)->으로 바꿔 쓸 수 있다.

표 1: 물고기 구조체를 위한 표		
종류	무게 (gram)	길이 (cm)
도미	500	30
우럭	400	20
광어	1000	60
도다리	490	30
장어	600	60