객체지향프로그래밍 퀴즈 #1 / E13 해답

강대기

2007년 9월 27일

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

1. 자갈치 시장의 아주머니를 돕기 위한 물고기에 대한 구조체 선언을 고려해 보자.

   우선 이 구조체에는 물고기의 종류(kind)를 나타내는 20 바이트 정도 길이의 문자열, 정수(int)로 된 그램(gram) 단위의 무게(weight), 소수부(double)가 있는 센티미터(cm) 단위의 길이 (length)를 정의해야 한다. 즉 이 구조체는 kind, weight, length라는 세 개의 멤버 변수를 갖는다.

   이제 마음 속으로 구상한 구조체를 C++ 형식으로 선언하는 데, 바로 구조체의 배열을 정의하면서 그 내용도 '도미', '우럭', '광어', '도다리' 그리고 '장어'에 대해 초기화하라. 각각의 고기의 무게와 길이는 다음에 나오는 표를 참조한다.

   즉, 지금까지 언급한 모든 게 하나의 명령문(statement)에서 이루어져야 한다. 더 구체적으로 말하자면, 구조체 내에서 멤버 변수들을 정의하는 경우들을 제외하고는, 하나의 세미콜론만 쓰게 된다는 것이다.

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

답으로 작성할 코드의 한 예은 다음과 같다.

```
struct Fish
{
  char kind[20]; // 물고기의 종류
  int weight; // 그램 단위의 무게
```

```
  double length; // 센티미터 단위의 길이
} fish[5] =
{
  {"도미",500,30.0},
  {"우럭",400,20.0},
  {"광어",1000,60.0},
  {"도다리",490,30.0},
  {"장어",600,60.0},
};
```

2. 위에서 선언한 구조체 형을 가지고, new 명령으로 동적 구조체를 하나 생성하라. 그 다음에 그 생성된 동적 구조체에 "농어"와 그에 대한 정보를 집어넣는다. 농어의 무게와 길이는 적당한 값을 알아서 넣도록 한다. #include <cstring> 이나 #include <string> 은 이미 있다고 가정한다.

   답으로 작성할 코드의 한 예은 다음과 같다.

```
Fish *nong = new Fish;
strcpy(nong->kind, "농어");
nong->weight = 100;
nong->length = 10.0;
```

3. 다음의 코드는 위험하다. 어째서인가?

```
long* fellow;
*fellow = 233233;
```

   fellow라는 long 형의 개체를 위한 포인터를 정의하였으나, 그 포인터가 가리키는 주소를 초기화하지 않았다. 이 상태에서는 fellow 포인터가 어디를 가리킬지 모른다. 그런데, 그 fellow 포인터가 가리키는 곳에 데이터를 대입하고 있다. 이런 명령은 운영체제, 또는 다른 프로그램이나 그 코드를 실행하는 프로그램 자체에 나쁜 영향을 미친다.

4. ted는 double 형 변수이다. ted를 가리키는 포인터를 정의하고 그 포인터를 사용하여 ted 값을 출력하라.

   답으로 작성할 코드의 한 예은 다음과 같다.

```
double* pointer = &ted;
cout << *pointer << endl;
```

5. treacle은 10 개의 double 형 원소들을 가지는 배열이다. treacle의 첫번째 원소를 가리키는 포인터를 정의하고, 그 포인터를 사용하여 treacle 배열의 첫번째 원소와 마지막 원소를 출력하라.

   답으로 작성할 코드의 한 예은 다음과 같다.

```
double* pointer = treacle;
cout << *pointer << endl;
cout << *(pointer+10-1) << endl;
```

6. 코드 88에 해당하는 문자를 C++에서 출력하는 방법들 중 네가지를 보여라.

   네가지 방법은 다음과 같다.

   - `char c=88;`
     `cout << c << endl;`
   - `cout.put(char(88));`
   - `cout << char(88) << endl;`
   - `cout << (char)88 << endl;`

7. 다음의 코드를 보자. 실행한다면 어떤 문자열을 출력할까?

```
unsigned x = 100;
if (x>-2)
  std::cout << "크다." << std::endl;
else
  std::cout << "작다." << std::endl;
```

   "작다"가 출력된다. -2는 signed int이고 x는 unsigned int이다. 이 둘을 비교하는 경우, signed int가 unsigned int로 변환된다. 따라서 -2는 그에 해당하는 unsigned int값인 큰 양수값으로 비뀌고(비쥬얼 스튜디오에선 4294967294), 100인 x의 값과 비교되므로, "작다"가 출력된다.

8. 다음의 명령문들은 무엇을 출력하는가?

```
int x = 19.99 + 11.99;
std::cout << x << std::endl;
int y = (int)19.99 + 11.99;
std::cout << y << std::endl;
int z = (int)19.99 + (int)11.99;
std::cout << z << std::endl;
```

   31, 30, 30 순서로 한 행씩 출력된다. x의 경우 19.99와 11.99는 같은 형(type)이므로 그대로 더해져 31.98이 되고 정수형으로 변환되면서 31이 된다. y의 경우, 19.99는 19로 변환되고, 11.99와 더해져서 30.99가 된다. 그리고 나서 y에 대입되면서 0.99가 버려져, 31이 된다. z의 경우, 19.99는 19가 되고, 11.99는 11로 변환되어 더해지므로 30이 된다.

9. 다음의 변수에 대해 표준 입력 (cin)에서 한 행을 입력받으려면 어떻게 해야 하는가? 직접 실제로 수행할 수 있는 C++ 명령문들을 써라.

   (a) char cname[20]; → `cin.getline(cname, 20);`

   (b) string sname; /* #include <string> 이 이미 있다고 가정 */ → `getline(cin,sname);`

10. #define 대신 const를 사용할 때의 장점을 세가지 적어라.

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