

1 C++ 표준 입출력에 대한 설명으로 잘못된 것은?

- 1 표준 출력은 cout, 표준 입력은 cin을 사용한다.
- 2 헤더를 포함해야 사용 할 수 있다.
- 3 cout으로 정수를 출력할 때는 "%d", 실수를 출력할때는 "%f"를 사용한다.
- 4 개행을 할때는 'wn' 또는 endl을 사용한다.

2 다음 설명 중 틀린 것은?

- 1 C++11 부터는 0b 를 사용하면 이진수도 표기할수 있다.
- 2 C++에서는 구조체형 변수를 사용할 때 struct 를 붙여야 한다.
- 3 C++11에서 추가된 auto를 사용하면 등호의 우변식으로 변수의 타입을 추론할 수 있다.
- 4 {}로 초기화 하면 일반 변수, 구조체, 배열 등을 모두 동일한 방식으로 초기화 할 수 있다.

3 C++11에서 추가된 포인터 0 을 의미하는 것은 무엇인가?

- 1 empty
- 2 NULL
- 3 nullptr
- 4 zero

4 C++ 함수에 대한 설명으로 잘못된 것은?

- 1 함수 인자의 타입이나 갯수가 다르면 동일 이름을 가지는 함수는 2개 이상 만들 수 있다.
- 2 함수 인자의 값을 전달하지 않으면 디폴트 값을 사용할 수 있다.
- 3 함수를 삭제하는 것은 함수를 제공하지 않은 것과 동일한 효과를 나타낸다.
- 4 함수 인자의 타입만 다르고 구현이 동일한 경우 함수 템플릿을 사용할 수 있다.

5 다음 중 디폴트 인자를 바르게 지정한 것은?

- 1 void foo(int a = 0, int b = 0, int c);
- 2 void foo(int a = 0, int b, int c = 0);
- 3 void foo(int a, int b = 0, int c = 0);
- 4 void foo(int a, int b = 0, int c);

6 다음 중 레퍼런스에 대한 설명으로 잘못된 것은 ?

- 1 구조체 등의 사용자 정의 타입을 인자로 전달할때는 call by value 보다는 const reference가 좋다.
- 2 기존 메모리에 또 다른 이름을 부여하는 문법이다.
- 3 함수가 값을 리턴하면 임시객체가 생성되지만 참조를 리턴하면 임시객체가 생성되지 않는다.
- 4 NULL포인터 처럼 NULL 참조도 만들수 있다.

7 다음 중 C++의 객체지향 문법에 대한 설명 중 잘못된 것은?

- 1 C++은 구조체를 만들때 함수를 포함할 수 있다.
- 2 접근지정자를 사용하면 외부의 잘못된 사용으로부터 멤버 data가 잘못된 값을 가지는 것을 막을 수 있다.
- 3 일반적으로 객체를 초기화 하기 위해 생성자를 사용한다.
- 4 함수는 템플릿으로 만들수 있지만 class 는 템플릿으로 만들수 없다.

8 C++의 객체지향 문법에 관한 설명 중 잘못된 것은?

- 1 접근 지정자를 생략시 struct 는 public, class 는 private 이 된다.
- 2 생성자와 소멸자는 리턴타입과 인자를 가질수가 없다.
- 3 객체를 만들면 생성자가 호출되고, 객체가 파괴 될때 소멸자가 호출된다.
- 4 malloc으로 객체를 만들면 생성자가 호출되지 않지만, new로 객체를 만들면 생성자가 호출된다.

9 초기화 리스트에 대한 설명으로 잘못된 것은?

- 1 대입이 아닌 실제 초기화이다.
- 2 멤버 데이터가 사용자 정의 타입인 경우 생성자 호출 한번으로 초기화 되므로 빠르다.
- 3 디폴트 생성자가 없는 객체를 멤버로 가질때는 초기화 리스트로 초기화 해야 한다.
- 4 멤버 데이터가 놓인 순서가 아니라 초기화 리스트 코드에 놓인 순서대로 멤버가 초기화 된다.

10 다음 중 복사 생성자가 호출되는 경우가 아닌 것은?

- 1 자기 자신과 동일한 타입의 객체로 초기화 될 때
- 2 함수 인자로 사용자 정의 타입의 객체를 값으로 전달될 때
- 3 함수가 사용자 정의 타입의 객체를 값으로 리턴 할 때
- 4 함수가 const reference 를 사용해서 객체를 전달 받을 때

11 다음 중 복사 생성자에 대한 설명으로 잘못된 것은?

- 1 객체 생성시 자신과 동일한 타입으로 초기화 할 때 호출된다.
- 2 사용자가 만들지 않으면 컴파일러가 제공해 준다.
- 3 컴파일러가 만들어 주는 복사 생성자는 아무 일도 하지 않는다.
- 4 복사 생성자를 제공한 경우, 컴파일러는 디폴트 생성자를 제공하지 않는다.

12 아래 코드에서 에러가 아닌 것을 찾아 보세요

```
class Test
{
    int idata;
    static int sdata;
public:
    void foo() {}

    static void fn()
    {
        sdata = 10;           // (1)
        idata = 10;           // (2)
        foo();                 // (3)
        std::cout << this;    // (4)
    }
};

int Test::sdata = 0;
```

13 다음 중 정적 멤버(static member)에 대한 설명으로 잘못된 것은?

- 1 정적 멤버 데이터(static member data) 최초에 객체를 만들 때 생성되고, 이후에는 모든 객체가 공유하게 된다.
- 2 정적 멤버 함수(static member function)에서는 정적 멤버만 접근할 수 있다.
- 3 객체의 크기를 구할때 정적 멤버 데이터(static member data)는 포함 되지 않는다.
- 4 정적 멤버 함수(static member function)은 객체없이 클래스 이름 만으로 호출할 수 있다.

14 다음 중 상수 멤버 함수에 대한 설명으로 틀린 것은?

- 1 상수 멤버 함수에서는 모든 멤버를 상수 취급한다.
- 2 상수 객체는 상수 멤버 함수만 호출 할 수 있다.
- 3 상수 멤버 함수는 실수를 막기 위해, 안정성을 위해서만 사용하므로 반드시 필요하지는 않다.
- 4 동일 이름의 상수 멤버 함수와 비상수 멤버 함수를 동시에 만들 수 있다.

15 다음중 예외를 던질때 사용하는 키워드는 ?

- 1 try
- 2 catch
- 3 throw
- 4 noexcept

16 다음 중 연산자 재정의 대한 설명 중 잘못된 것은?

- 1 '+' 연산자는 반드시 멤버 함수로만 구현해야 한다.
- 2 '+' 연산자의 함수 이름은 operator+() 이다.
- 3 이항 연산자 함수를 멤버로 구현 할 때는 인자를 한 개 가져야 한다.
- 4 연산자 재정의할 때 인자중 한개 이상은 반드시 사용자 정의 타입이 되어야 한다.

17 가상 함수에 대한 설명으로 틀린 것은?

- 1 함수 의 호출을 실행시간에 결정한다.
- 2 기반 클래스 포인터로 파생 클래스를 가리킬 때 파생 클래스의 함수가 호출되게 하기 위해 사용한다.
- 3 가상함수를 만들때는 virtual 키워드를 사용한다.
- 4 포인터 타입을 보고 어떤 함수가 호출될지는 결정한다.

18 객체가 가지고 있는 상수성을 제거하기 위한 C++ 캐스팅 연산자는 무엇인가?

- 1 static_cast
- 2 const_cast
- 3 reinterpret_cast
- 4 dynamic_cast

19 다음 중 STL에 대한 설명으로 잘못된 것은?

- 1 컨테이너는 다양한 자료구조를 나타내는 템플릿 기반의 클래스이다.
- 2 반복자는 컨테이너와 알고리즘을 연결해주는 포인터를 추상화한 객체이다.
- 3 list와 vector는 내부 자료구조가 다르므로 list전용 find와 vector 전용 find를 따로 제공하고 있다.
- 4 C++11의 STL에는 스레드, 정규 표현식 등의 라이브러리도 제공된다.

20 다음 중 STL 컨테이너에 대한 설명으로 잘못된 것은?

- 1 list, stack queue, tree, hash 등 다양한 자료구조를 제공한다.
- 2 요소를 제거하는 함수와 꺼내는 함수가 분리되어 있다.
- 3 vector에 앞에 요소를 추가할때는 push_front() 함수를 사용한다.
- 4 템플릿 기반으로 되어 있다.