

문제 번호		내용
1	문제	C++ 언어에 대한 설명으로 잘못된 것은 ?
1	1	C++11 이후 나오는 문법을 "Modern C++" 이라고 한다.
1	2	파일 확장자는 관례적으로 .cpp 를 사용한다.
1	3	객체지향 프로그래밍 언어 이다.
1	4	C 언어와는 호환 되지 않으므로 C++ 에서는 C 함수를 호출할수 없다.
2	문제	다음 설명 중 틀린 것은?
2	1	C++에서는 함수 중간에 변수를 선언할 수 있다.
2	2	C++에서는 구조체형 변수를 사용할 때 반드시 struct 를 붙여야 한다.
2	3	C++11에서 추가된 auto를 사용하면 등호의 우변식으로 변수의 타입을 추론할 수 있다.
2	4	{ }로 초기화 하면 일반 변수, 구조체, 배열 등을 모두 동일한 방식으로 초기화 할 수 있다.
3	문제	C++11에서 추가된 포인터 0을 의미하는 것은 무엇인가?
3	1	empty
3	2	NULL
3	3	nullptr
3	4	zero
4	문제	다음중 C++ 표준 출력에 사용되는 객체는 ?
4	1	std::cout
4	2	std::cin
4	3	std::ostream
4	4	std::printf
5	문제	다음 중 디폴트 인자를 바르게 지정한 것은?
5	1	void foo(int a = 0, int b = 0, int c);
5	2	void foo(int a = 0, int b, int c = 0);
5	3	void foo(int a, int b = 0, int c = 0);
5	4	void foo(int a, int b = 0, int c);
6	문제	다음 중 레퍼런스에 대한 설명으로 잘못된 것은 ?
6	1	구조체 등의 사용자 정의 타입을 인자로 전달할때는 call by value 보다는 const 참조가 좋다.
6	2	기존 메모리에 또 다른 이름을 부여하는 문법이다.
6	3	함수가 값을 리턴하면 임시객체가 생성되지만 참조를 리턴하면 임시객체가 생성되지 않는다.
6	4	NULL포인터 처럼 NULL 참조도 만들수 있다.
7	문제	다음 중 C++의 객체지향 문법에 대한 설명 중 잘못된 것은?
7	1	C++은 구조체를 만들때 함수를 포함할 수 있다.
7	2	접근지정자를 사용하면 외부의 잘못된 사용으로부터 멤버 data가 잘못된 값을 가지는 것을 막을 수 있다.
7	3	일반적으로 객체를 초기화 하기 위해 생성자를 사용한다.
7	4	클래스를 만들때는 반드시 클래스 선언부 안에 멤버 함수의 구현도 포함해야 한다.

8	문제	다음중 우변의 표현식으로 좌변 변수의 타입을 자동으로 결정되게 하는 키워드는 ?
8	1	auto
8	2	decltype
8	3	template
8	4	var

9	문제	다음 중 복사 생성자가 호출되는 경우가 아닌 것은?
9	1	자기 자신의 타입의 객체로 초기화 될 때
9	2	함수 인자로 객체가 값으로 전달될 때
9	3	함수가 객체를 값으로 리턴 할 때
9	4	함수가 const reference를 사용해서 객체를 전달 받을 때

10	문제	다음 중 복사 생성자에 대한 설명으로 잘못된 것은?
10	1	객체 생성시 자신과 동일한 타입으로 초기화 할 때 호출된다.
10	2	사용자가 만들지 않으면 컴파일러가 제공해 준다.
10	3	컴파일러가 만들어 주는 복사 생성자는 아무 일도 하지 않는다.
10	4	복사 생성자를 제공한 경우, 컴파일러는 디폴트 생성자를 제공하지 않는다.

11	문제	객체의 복사에 대한 설명 중 잘못 된 것은?
11	1	클래스 안에 포인터 멤버가 있을 때 디폴트 복사 생성자는 깊은 복사 현상을 발생 시킨다.
11	2	자원을 공유 하면서, 몇명의 객체가 자원을 사용하는지 갯수를 관리하는 기술을 reference counting 이라고 한다.
11	3	복사를 금지하기 위해서는 복사생성자를 삭제 하거나, private 영역에 선언만 제공하면 된다.
11	4	복사생성자를 만드는 경우 보통은 대입연산자도 같이 만들어야 한다.

12	문제	다음 중 연산자 재정의 대한 설명 중 잘못된 것은?
12	1	+연산자는 반드시 멤버 함수로만 구현해야 한다.
12	2	+연산자의 함수 이름은 operator+() 이다.
12	3	이항 연산자 함수를 멤버로 구현 할 때는 인자를 한 개 가져야 한다.
12	4	연산자 재정의할 때 인자중 한개 이상은 반드시 사용자 정의 타입이 되어야 한다.

13	문제	표준 입출력에 대한 설명으로 잘못된 것은?
13	1	endl 은 함수이다.
13	2	cout은 operator<< 연산자 재정의를 사용해서 만들어졌다.
13	3	cout의 operator<< 연산자의 리턴 값은 자기 자신의 참조 타입이다.
13	4	사용자 정의 타입을 cout으로 출력 할 수 는 없다.

14	문제	다음중 예외를 던질때 사용하는 키워드는 ?
14	1	catch
14	2	raise
14	3	throw
14	4	try

15	문제	소멸자에 대한 설명으로 잘못된 것은 ?
15	1	사용자가 만들지 않으면 컴파일러가 만들어 준다.
15	2	자원해지등이 필요한 경우에만 만들면 된다.
15	3	인자가 있을수도 있고 없을수도 있다.
15	4	객체가 파괴될때 자동으로 호출된다.

16	지문	<pre> class Test { int a; static int b; public: void goo() {} static void foo() { a = 10; // 1 b = 10; // 2 cout << this; // 3 goo(); // 4 } }; </pre>
16	문제	다음 코드 중 컴파일 에러가 아닌 것은?
16	1	1
16	2	2
16	3	3
16	4	4
17	문제	다음 중 잘못된 것은?
17	1	OCP 란, 개방 폐쇄의 원칙으로, 나중에 기능이 추가되어도 기존에 있던 코드는 수정 되지 않도록 설계 하자는 개념이다.
17	2	확장성을 고려 한다면 약한 결합(loosely coupling)이 좋다.
17	3	인터페이스는 파생 클래스에서 특정 함수를 반드시 만들라고 지시하는 것이다.
17	4	C++ 에서 인터페이스를 만들때는 interface 키워드를 사용한다.
18	문제	다음 중 STL 컨테이너에 대한 설명으로 잘못된 것은?
18	1	list, stack queue, tree, hash 등 다양한 자료구조를 제공한다.
18	2	요소를 제거하는 함수와 꺼내는 함수가 분리되어 있다.
18	3	vector에 앞에 요소를 추가할때는 push_front() 함수를 사용한다.
18	4	템플릿 기반으로 되어 있다.
18	문제	s가 list <int> 일때, 반복자를 꺼내는 코드가 아닌 것은?
18	1	list<int>::iterator p = s.begin();
18	2	list<int>::iterator p = begin(s);
18	3	auto p = begin(s);
18	4	auto p = s.front();
19	문제	다음 중 STL의 알고리즘에 대한 설명으로 잘못된 것은?
19	1	컨테이너에 있는 요소를 조작하기 위한 다양한 멤버함수를 알고리즘이라고 부른다.
19	2	알고리즘이 사용하는 정책을 변경하려면, 함수, 함수객체, 람다 표현식등을 알고리즘의 인자로 전달하면 된다.
19	3	알고리즘이 실패 할 때는 past the end 반복자를 리턴한다.
19	4	STL에는 선형검색, 이진검색, 정렬등의 다양한 알고리즘이 제공된다.
20	문제	다음중 예외를 잡을 때 사용하는 키워드는 ?
20	1	try
20	2	throw
20	3	raise

20	4	catch