

2개월차 시험지

총점 70/100 ?

2개월 수업 종료 이후에 치루는 시험지입니다.

이름을 한글로 작성하세요.(미작성 시 제출 불가) *

150기 강명호

✕ 1. 다음의 출력 결과로 올바른 것을 고르시오.

0/5

1. **int** **a = 10;**
2. **float** **b = 3.14f;**
3.
4. **cout << a * b << endl;**

- ☐ 1 31.4
☒ 2 31
☐ 3 314
☐ 4 30
☐ 5 3.140

정답

- ☒ 1 31.4

✕



✕ 2. 다음의 설명 중 틀린 것을 모두 고르시오.(3개)

0/5

- ☐ 1 C++에서 레퍼런스의 크기는 4byte이다.
- ☒ 2 C언어에서 레퍼런스의 크기는 4byte이다. ✓
- ☒ 3 레퍼런스는 초기 참조 이후 다른 공간을 참조하게 변경할 수 없다. ✕
- ☒ 4 레퍼런스도 포인터와 마찬가지로 NULL로 초기화가 가능하다. ✓
- ☐ 5 레퍼런스는 C++부터 등장한 문법이다.

정답

- ☒ 1 C++에서 레퍼런스의 크기는 4byte이다.
- ☒ 2 C언어에서 레퍼런스의 크기는 4byte이다.
- ☒ 4 레퍼런스도 포인터와 마찬가지로 NULL로 초기화가 가능하다.

✕ 3. 다음 중 C++의 함수 호출 규약이 아닌 것은?

0/5

- ☐ 1 __stdcall
- ☐ 2 __cdecl
- ☒ 3 __fastcall ✕
- ☐ 4 __lastcall
- ☐ 5 __thiscall

정답

- ☒ 4 __lastcall



✓ 4. stl에서 vector의 멤버 함수 중 메모리 공간을 예약하는 함수의 이름을 고 5/5
르시오.

- ☐ 1 push_back
- ☐ 2 insert
- ☒ 3 reserve
- ☐ 4 size
- ☐ 5 reverse



✕ 5. 다음 코드의 출력 결과를 올바른 것을 고르시오.

0/5

```
1.      int      a = 7;  
2.      int      b = 9;  
3.      int      c = 3;  
4.  
5.      c &= a;  
6.      c |= b;  
7.      cout << c << endl;
```

☒ 1 00001011

✕

☐ 2 11

☐ 3 1011

☐ 4 00000111

☐ 5 13

정답

☒ 2 11



✓ 6. 다음 코드의 출력 결과로 옳은 것을 고르시오.

5/5

```
1.      map<int, int, greater<int>>  IntMap;
2.
3.      IntMap.insert(pair<int, int>(1,10));
4.      IntMap.insert(pair<int, int>(2,20));
5.      IntMap.insert(pair<int, int>(3,30));
6.      IntMap.insert(pair<int, int>(1,40));
7.
8.      map<int, int, greater<int>>::iterator iter;
9.      iter = IntMap.find(1);
10.
11.     cout << iter->second << endl;
12.
13.     for(iter = IntMap.begin();iter !=IntMap.end(); ++iter)
14.     {
15.         cout << iter->second << endl;
16.     }
```

- ☐ 1 10 - 10 - 20 - 30
- ☐ 2 40 - 30 - 20 - 10
- ☒ 3 10 - 30 - 20 - 10
- ☐ 4 10 - 40 - 30 - 20
- ☐ 5 40 - 10 - 20 - 30



✕ 7. 클래스의 4대 속성이 아닌 것을 고르시오.

0/5

- ☐ 1 은닉화
- ☒ 2 상속성
- ☐ 3 추상화
- ☐ 4 다형성
- ☐ 5 캡슐화

✕

정답

- ☒ 3 추상화



✓ 8. 다음의 코드를 보고 잘못된 부분을 올바르게 지적한 답을 고르시오.

5/5

```
class CObj
{
Private:
    int* p;
public:
    CObj() { p = new int; }
    ~CObj() { delete p; }
};
class CPlayer : public CObj
{
Private:
    int* p2;
public:
    CPlayer() { p2 = new int; }
    ~CPlayer() { delete p2; }
};
void main(void)
{
    CObj* Player = new CPlayer;
    delete Player;
    Player = NULL;
}
```

- ☐ 1 멤버 변수의 이름으로 헝가리안 표기법을 사용하지 않아 컴파일 에러가 발생한다.
- ☐ 2 생성자와 소멸자의 몸체를 클래스 내부에서 정의할 수 없다.
- ☐ 3 부모 타입의 포인터 Player이기 때문에 동적할당을 new CObj로 하는 것이 옳다.
- ☐ 4 Player 포인터를 delete한 이후 NULL을 채워넣는 것은 매우 위험한 행동이다.
- ☒ 5 부모 소멸자 앞에 virtual 키워드가 없어서 자식 소멸자가 호출되지 않는다. ✓



✓ 9. 클래스를 생성했을 때, 컴파일러가 자동적으로 생성하는 것이 아닌 것을 5/5
고르시오.

- ☐ 1 default 생성자
- ☐ 2 default 복사 생성자
- ☐ 3 default 대입 연산자
- ☒ 4 default 복사 소멸자
- ☐ 5 default 소멸자



✗ 10. 다음은 vector의 멤버 함수 기능을 설명한 문장이다. 올바르지 않은 것을 0/5
모두 고르시오.

- ☐ 1 empty()함수는 vector가 비었는지 조사한다.
- ☒ 2 capacity(n)는 n개의 vector원소를 저장할 공간을 예약한다.
- ☒ 3 pop_front()함수를 통해 vector의 맨 앞 원소를 제거할 수 있다.
- ☒ 4.erase 함수는 vector의 원소를 제거한 뒤, 다음 원소를 가리킨다.
- ☐ 5 size() 함수는 vector 원소의 개수를 반환한다



정답

- ☒ 2 capacity(n)는 n개의 vector원소를 저장할 공간을 예약한다.
- ☒ 3 pop_front()함수를 통해 vector의 맨 앞 원소를 제거할 수 있다.



✓ 11. 다음 코드의 출력 결과로 올바른 것을 고르시오.

5/5

```
class CInfo
{
public:
    int    m_iX;
public:
    CInfo(int m_iX) {          this->m_iX = m_iX;    }
};
void main(void)
{
    CInfo  Info = 20;
    cout << Info.m_iX << endl;
}
```

- ☐ 1 쓰레기값
- ☒ 2 20
- ☐ 3 0
- ☐ 4 -20
- ☐ 5 컴파일 에러



✓ 12. 출력 결과로 올바른 것을 고르시오.

5/5

```
list<int>          intList;

intList.push_back(10);
intList.push_back(20);
intList.push_back(30);

list<int>::iterator      iter;

iter = intList.begin();
++iter;

iter = intList.erase(iter);

cout << *iter << endl;
```

- ☐ 1 10
- ☐ 2 20
- ☐ 3 25
- ☐ 4 29
- ☒ 5 30



✓ 13. 다음 설명 중 괄호 안에 들어갈 답을 고르시오.

5/5

()는 포인터와 비슷하게 동작한다. ()는 컨테이너에 저장된 원소를 순회하고 접근하는 일반화된 방법을 제공하며, 컨테이너와 알고리즘이 하나로 동작하게 묶어주는 인터페이스 역할을 한다.

()덕에 알고리즘은 특정 컨테이너에 종속적이지 않고 독립적이면서도 언제든지 컨테이너와 결합하여 동작할 수 있다.

- ☐ 1 레퍼런스
- ☐ 2 상수
- ☐ 3 매크로
- ☒ 4 반복자
- ☐ 5 조건자



✓ 14. 다음 코드의 출력 결과로 올바른 것을 고르시오.

5/5

```
class A
{
public:
    virtual void f() { cout << 1<< endl; }
    void g() { f(); }
};
class B : public A
{
public:
    virtual void f() { cout << 2 << endl; }
    void g() { A::f(); }
};
void main()
{
    A a;
    B b;
    A* c = &b;
    a.f();
    a.g();
    b.f();
    b.g();
    c->f();
    c->g();
}
```

- ☒ 1 1, 1, 2, 1, 2, 2
- ☐ 2 1, 2, 1, 1, 2, 2
- ☐ 3 1, 2, 2, 1, 1, 2
- ☐ 4 1, 1, 2, 1, 1, 2
- ☐ 5 1, 1, 1, 2, 2, 2



✓ 15. 다음 중 stl 컨테이너가 아닌 것을 고르시오.

5/5

- ☐ 1 vector
- ☐ 2 map
- ☐ 3 list
- ☒ 4 sort
- ☐ 5 string



✓ 16. stl에서 list의 멤버 함수 중 원소를 정렬하는 함수를 고르시오.

5/5

- ☐ 1 push_back
- ☒ 2 sort
- ☐ 3 emplace
- ☐ 4 swap
- ☐ 5 pop_front



✓ 17. 다음 중 사용자 정의 자료형이 아닌 것을 고르시오.

5/5

- ☐ 1 클래스
- ☒ 2 문자열
- ☐ 3 구조체
- ☐ 4 공용체
- ☐ 5 열거체



✓ 18. 다음 설명을 보고 알맞은 답을 고르시오.

5/5

()는 C++에서 제공하는 기본 타입이 아닌 클래스 타입,
즉 사용자 정의 타입에도 연산자를 사용할 수 있게 하는 문법이다.
()는 많은 STL 구성요소에 사용되 쉽고 더 직관적이며
일반화된 코드를 만들어낼 수 있다.

- ☐ 1 오버라이딩
- ☒ 2 연산자 오버로딩
- ☐ 3 디폴트 매개변수
- ☐ 4 연산자 축약형
- ☐ 5 반복자



✓ 19. 다음 설명을 보고 알맞은 답을 고르시오.

5/5

()컨테이너는 연관 컨테이너 중 자주 사용하는 컨테이너로
원소를 KEY와 VALUE 쌍으로 저장하며
원소의 KEY는 중복 저장될 수 없다.

- ☐ 1 queue
- ☐ 2 stack
- ☒ 3 map
- ☐ 4. list
- ☐ 5 vector



✓ 20. 다음 중 메모리 함수가 아닌 것을 고르시오.

5/5

- ☐ 1 memcpy
- ☐ 2 memmove
- ☐ 3 memset
- ☐ 4 memcmp
- ☒ 5 memremove



이 콘텐츠는 Google이 만들거나 승인하지 않았습니다. [악용사례 신고](#) - [서비스 약관](#) - [개인정보처리방침](#)

Google 설문지



