

DEPT: \_\_\_\_\_ SID: \_\_\_\_\_ Name: \_\_\_\_\_

**이론 시험문제(100점,40분):**

1. 아래 명제들의 참, 거짓을 판단하시오 (각각 정답 2점, 오답 -2점, 무응답 0점).(20 pts)

- (1) ( ) Virtual 함수는 이후 상속되는 모든 계층에서 virtual 함수이다.
- (2) ( ) Function template와 class template의 특수화(specialization)는 프로그램의 실행시 (runtime) 이루어진다.
- (3) ( ) 기본 클래스의 어떤 멤버 함수를 파생 클래스에서 다시 정의하면 기본 클래스의 멤버 함수는 파생 클래스의 객체에서는 더 이상 사용할 수 없다.
- (4) ( ) 추상 클래스도 객체를(인스턴스를) 생성할 수 있다.
- (5) ( ) 기본 클래스의 생성자(constructor) 및 소멸자(destructor)도 파생 클래스(derived class)에 상속된다.
- (6) ( ) 클래스에서 private 멤버는 외부에서 접근할 수 없다
- (7) ( ) 함수의 인자(parameter)의 개수와 함수호출시 argument의 개수는 반드시 동일해야 한다.
- (8) ( ) 클래스 정의내에 선언된 변수를 데이터 멤버라고 한다.
- (9) ( ) 소멸자의 소멸순서는 항상 생성자의 생성순서와 반대로 수행된다.
- (10)( ) const 데이터멤버는 일반 멤버와 동일한 방식으로 초기화 할 수 있다.

2. Basic concept (20 pts = 4pts x 5):

- (1) Template을 사용했을때의 장점에 대해 서술하시오.
- (2) 클래스에서 다형성 기능을 부여했을때의 장점을 서술하고, 실제 다형성을 어떻게 구현하는지에 대해서 설명하시오.
- (3) 상속계층에서 파생클래스가 기본클래스를 public, protected, private으로 상속받았을 때 public, protected, private 멤버들에 대한 접근성에 대해 설명하시오.
- (4) 기본클래스에서 Pure virtual 과 virtual 함수의 차이를 설명하시오.

2021 Final Exam of Applications of OOP (ICE2013)

DEPT: \_\_\_\_\_ SID: \_\_\_\_\_ Name: \_\_\_\_\_

3. Fill in the blanks in each of the following statements (20 pts = 4 pts x 5):

- (1) 클래스의 다형성(Polymorphism)은 키워드 \_\_\_\_\_을(를) 사용하여 구현된다.
- (2) a와 b가 정수형 변수이고 합  $a+b$ 를 계산한다고 가정하고, c와 d가 부동 소수형 변수이고 합  $c+d$ 를 계산한다고 가정하자. 여기서 "+"연산자는 분명히 다른 용도로 사용되고 있다. 이러한 기능을 \_\_\_\_\_(이)라고 부른다.
- (3) \_\_\_\_\_연산자는 지정된 데이터형의 오브젝트에 대한 메모리를 동적으로 할당하고 그 데이터에 대한 \_\_\_\_\_를 반환한다.
- (4) 클래스 멤버 오브젝트에 대해 멤버초기화 값을 제공하지 않으면 오브젝트의 \_\_\_\_\_이(가) 호출된다.
- (5) 오브젝트의 비정적 멤버함수는 \_\_\_\_\_포인터라는 오브젝트 자체를 가리키는 포인터를 액세스할 수 있다.

4. 객체지향프로그래밍(OOP)의 4대 특성으로 추상화(abstraction), 캡슐화(encapsulation), 상속(inheritance), 다형성(polymorphism)등이 있다. 아래의 C++ 언어의 특징이 각각 OOP의 어느 특성에 해당하는지 대응시키시오. (10pts)

- A. Pure virtual function →
- B. 클래스 재사용성(reusability) →
- C. 'is-a' 관계 →
- D. 동일한 기본 클래스(base class)로부터 상속을 받는 서로 다른 클래스의 객체들에 대한 동일한 interface →
- E. Data hiding →

2021 Final Exam of Applications of OOP (ICE2013)

DEPT: \_\_\_\_\_ SID: \_\_\_\_\_ Name: \_\_\_\_\_

5. Single selection (10pts):

1) (\_\_\_\_\_) Which of the following is not a kind of inheritance in C++?

- A. public
- B. protected
- C. static.
- D. private

2) (\_\_\_\_\_) Given the class definition:

```
class CreateDestroy
{
public:
    CreateDestroy() { cout << "constructor called, "; }
    ~CreateDestroy() { cout << "destructor called, "; }
};
```

What will the following program output?

```
int main()
{
    CreateDestroy c1;
    CreateDestroy c2;
    return 0;
}
```

- A. constructor called, destructor called, constructor called, destructor called, .
- B. constructor called, constructor called, destructor called, destructor called, .
- C. constructor called, destructor called, .
- D. constructor called, constructor called, .

DEPT: \_\_\_\_\_ SID: \_\_\_\_\_ Name: \_\_\_\_\_

3) ( ) Given the class definition:

```
class base
{
public:
    base() { cout << "constructor base called, "; }
    ~base() { cout << "destructor base called, "; }
};

class derived:public base
{
public:
    derived() { cout << "constructor derived called, "; }
    ~derived() { cout << "destructor derived called, "; }
};
```

What will the following program output?

```
int main()
{
    base *c1 = new base ;
    base *c2 = new derived;
    delete c1;
    delete c2;
    return 0;
}
```

- A. constructor base called, constructor base called, constructor derived called, destructor base called, destructor base called,
- B. constructor base called, constructor base called, constructor derived called, destructor base called, destructor base called, destructor derived called, destructor derived called
- C. constructor base called, constructor base called, destructor base called, destructor derived called, destructor derived called
- D. constructor derived called, constructor base called, constructor derived called, destructor base called, destructor base called,

4) ( ) 다음 중 어느 경우 컴파일 오류(compilation error)를 발생하는가?

- A. Assigning the address of a base-class object to a base-class pointer.
- B. Assigning the address of a base-class object to a derived-class pointer.
- C. Assigning the address of a derived-class object to a base-class pointer.
- D. Assigning the address of a derived-class object to a derived-class pointer.

DEPT: \_\_\_\_\_ SID: \_\_\_\_\_ Name: \_\_\_\_\_

6. 오른쪽 그림과 같은 클래스 상속(class inheritance) 구조를 가지는 클래스 A, B, C에 대하여 아래 물음에 답하시오. 클래스 A가 가장 상위 클래스이다. (20점)

(1) A의 객체 생성시 생성자와 소멸자의 call, execution, return의 순서를 시간순으로 올바르게 정렬하시오. (5pts)

(2) B의 객체 생성시 생성자와 소멸자의 call, execution, return의 순서를 시간순으로 올바르게 정렬하시오.(5pts)

(3) C의 객체 생성시 생성자와 소멸자의 call, execution, return의 순서를 시간순으로 올바르게 정렬하시오.(10pts)

