

C++ 프로그래밍, 멀티미디어시스템

2010학년도 2 학기

1 학년 3 교시

※ 1. 정답 하나만을 골라 반드시 컴퓨터용 사인펜으로 OMR 답안지에 표기할 것.	학 과		감독관	⑨
2. 답안정정은 일절 불가하니 각별히 유의할 것.	학 번		성 명	

1과목	C++ 프로그래밍 (1~25)
출제위원	방송대 이병래
출제범위	교재 3장~8장(해당강의 포함)

※ C++ 코드 중 문장 이해에 문제가 없는 부분은으로 생략하였음.

※ (1~4) 배열에 저장된 값들의 절대값의 합을 구하는 함수인 absSum에 대한 질문에 답하라.

```

..... // 생략
(ㄱ)
int main() {
    int a[10];
    ..... // 생략
    cout << absSum(a, 10) << endl; // (a)
    return 0;
}
int absSum(const int x[], int n) { // (b)
    int sum=0;
    for (int i=0 ; i < n ; i++) {
        if (.....) sum += x[i];
        else sum -= x[i];
    }
    (ㄷ)
}

```

1. 이 프로그램의 (a)에서 함수 absSum을 사용하기 위해 (ㄱ)에 넣을 적절한 문장은? (2점)

- ① int absSum()
- ② absSum()
- ③ void absSum(const int [], int);
- ④ int absSum(const int x[], int n);

2. sum에 x[i]의 절대값을 더해 넣기 위해 (ㄴ)에 넣을 내용은? (2점)

- ① x > 0 ② x[i] >= 0
- ③ x[i] == n ④ x[i] < 0;

3. (ㄷ)에 넣을 문장은 무엇인가? (2점)

- ① absSum = sum; ② return absSum;
- ③ return; ④ return sum;

4. (b)에서 'const'를 사용한 것에 대한 올바른 설명은? (2점)

- ① 함수 안에서 배열 x 안의 값을 수정하지 못한다.
- ② 특정 배열만 x에 전달할 수 있게 한다.
- ③ x에 전달되는 배열의 크기가 고정된다.
- ④ const로 지정된 배열만 x에 전달할 수 있다.

5. 다음과 같이 함수 f가 정의되었을 때, (b)의 호출 결과 x와 y에 저장되어 있는 값은? (3점)

void f(int a, int& b) { a *= 2; b += a; }	void g() { int x=5, y=10; f(x, y); // (b) // 생략 }
--	---

- ① x=5, y=20 ② x=10, y=20
- ③ x=10, y=10 ④ x=5, y=10

6. 클래스와 객체에 대한 올바른 설명은? (4점)

- ① 클래스 안에 생성자는 1개만 포함한다.
- ② 소멸자를 다중정의할 수 있다.
- ③ 가시성을 지시하지 않은 경우 protected로 지정된다.
- ④ 객체의 멤버를 프로그램 어느 곳에서든 사용할 수 있게 하려면 가시성을 public으로 지정한다.

7. 객체에서 this가 나타내는 것은? (2점)

- ① 객체 자신을 가리키는 포인터이다.
- ② 멤버 함수에 전달된 인수를 가리킨다.
- ③ 객체가 속한 클래스를 나타낸다.
- ④ 생성자를 나타낸다.

※ (8~10) 다음은 상향계수와 하향계수를 할 수 있는 계수기 클래스를 선언한 것이다. 생성자는 계수기의 최대값을 나타내는 값 m과 계수기의 초기값 v(디폴트는 0)를 지정하여 객체를 만든다. reset은 지정된 값(디폴트는 0)으로 초기화, cntUp과 cntDn은 상향 및 하향 계수를 하는 멤버함수이다.

```

class UDCntr {
    const int MAX; // 카운터의 최대값 (const에 주의)
    int val;
public:
    ..... (ㄱ)
    void reset(int v=0) { val = v; }
    void cntUp() { val = (val+1) % (MAX+1); }
    void cntDn() { val = (val+MAX) % (MAX+1); }
    int getValue() const { return val; }
};

```

8. 생성자를 선언하기 위해 (ㄱ)에 넣을 내용은? (3점)

- ① int UDCntr(int m, int v=0) : MAX(m), val(v) {}
- ② UDCntr(int m, int v=0) { MAX = m; val = v; }
- ③ UDCntr(int m, int v=0) : MAX(m), val(v) {}
- ④ ~UDCntr(int m, int v=0) { MAX = m; val = v; }

9. UDCntr 객체를 만드는 구문이 올바르게 사용된 것은? (3점)

- ① UDCntr c;
- ② UDCntr c(99, 0, 1);
- ③ UDCntr* pCntr = new UDCntr(99);
- ④ UDCntr c = new UDCntr;

10. c가 UDCntr 클래스의 객체라 할 때, 이 객체에 대해 사용할 수 있는 명령은 무엇인가? (3점)

- ① c.cntUp(1); ② c.reset(9);
- ③ cin >> c.val; ④ c.MAX = 99;

11. 클래스의 static 데이터 멤버에 대한 올바른 설명은? (2점)

- ① 클래스에 속한 객체마다 각각 하나씩 존재한다.
- ② static 데이터 멤버는 가시성을 public으로 지정해야 한다.
- ③ 일반 멤버함수에서는 static 데이터 멤버를 사용할 수 없다.
- ④ static 멤버함수에서 사용할 수 있는 데이터 멤버는 static 데이터 멤버이다.

12. 다음에 나열된 클래스들을 계층구조로 표현하려고 한다. 다음 중 다른 클래스들의 기초 클래스가 되어야 할 것은? (4점)

- ① 교육기관 ② 중학교
- ③ 고등학교 ④ 초등학교

※ (13~14) 시간을 표현하는 클래스 TimeC에 연산자 다중정의를 하고자 한다. 다음 질문에 답하라.

```
class TimeC {
    int  hours, minutes;
public:
    TimeC(int h, int m) : hours(h), minutes(m) {}

    _____ (㉠)
    TimeC operator + (int m) const;    // (㉡)
    ..... // 생략
};
```

13. 전위 표기 ++ 연산자를 정의하고자 한다. (㉠)에 넣을 연산자 다중정의 원형은? (2점)

- ① TimeC operator ++ (int) const;
- ② TimeC operator ++ (int);
- ③ TimeC operator ++ ();
- ④ TimeC operator ++ (TimeC& t);

14. (㉡)과 같이 + 연산자가 다중정의 되었을 때 사용할 수 있는 연산은? (tObj1과 tObj2는 TimeC 객체, n은 int형 변수임) (2점)

- ① tObj1 + tObj2 ② tObj1 + n
- ③ n + tObj1 ④ 40 + tObj1

15. 기초클래스와 파생클래스에 대하여 올바르게 기술한 것은? (4점)

- ① 파생클래스의 멤버함수에서는 기초클래스의 모든 멤버를 액세스할 수 있다.
- ② 기초클래스는 파생클래스의 속성과 메소드를 이어받는다.
- ③ 기초클래스의 멤버함수에서는 파생클래스의 모든 멤버를 액세스할 수 있다.
- ④ 기초클래스는 일반적 개념, 파생클래스는 이 일반적 개념에 속하면서 고유한 속성 및 행위를 갖는 대상을 표현한다.

※ (16~19) Person 클래스와 이의 파생클래스인 Athlete 클래스가 다음과 같을 때, 이에 대한 질문에 답하라.

```
class Person {
    string name;
public:
    Person(string n) : name(n) {}
    void who() const { cout << name ; }    // (㉠)
    string getName() const { return name; }
};

class _____ (㉡) {
    string team;
public:
    Athlete(string n, string t) _____ (㉢)
    void who() const
        { cout << team << "팀 " << getName(); }
};

void f() {
    Athlete* pa = new Athlete("김철수", "최강");
    Person* pp = pa;
    pa->who();
    cout << " ";
    pp->who();
}
```

16. 파생클래스 Athlete을 선언하기 위해 (㉡)에 넣을 내용은? (3점)

- ① Person(Athlete) ② Athlete : public Person
- ③ Athlete ④ public Athlete

17. Athlete의 생성자를 선언하기 위해 (㉢)에 넣을 내용은? (3점)

- ① : name(n), team(t) {} ② : Person(n), team(t) {}
- ③ { team = t; } ④ { name = n; team = t; }

18. 함수 f를 실행하였을 때 출력되는 내용은? (3점)

- ① 김철수 김철수
- ② 김철수
- ③ 최강팀 김철수 김철수
- ④ 최강팀 김철수 최강팀 김철수

19. (㉠)의 문장을 아래와 같이 수정하였을 때 함수 f에 의해 출력되는 내용은? (3점)

```
virtual void who() const { cout << name ; } // (㉠)
```

- ① 김철수 김철수
- ② 김철수
- ③ 최강팀 김철수 김철수
- ④ 최강팀 김철수 최강팀 김철수

20. 다음 클래스에 대한 설명이 올바른 것은? (4점)

```
class Shape {
    ..... // 생략
public:
    virtual int draw() = 0;
    ..... // 생략
};
```

- ① 이 클래스로는 객체를 만들 수 없다.
- ② 이 클래스에 대한 파생클래스를 만들 수 없다.
- ③ 함수 draw는 0을 반환한다.
- ④ Shape의 파생클래스는 draw 함수를 재정의할 필요가 없다.

21. 가상 기초클래스를 사용하는 목적은 무엇인가? (2점)

- ① 다중 상속을 금지하기 위한 것이다.
- ② 중복 상속이 발생했을 때 기초클래스가 한 번만 나타나게 한다.
- ③ 파생클래스를 선언할 수 없게 하기 위한 것이다.
- ④ 기초클래스에서 파생클래스의 멤버를 사용할 수 있도록 하기 위한 것이다.

22. 생성자와 소멸자에 대한 설명이 올바른 것은? (4점)

- ① 기초클래스의 생성자는 파생클래스의 생성자가 동작을 마친 후 실행된다.
- ② 파생클래스에 소멸자가 있으므로 기초클래스에는 소멸자가 필요없다.
- ③ 파생클래스의 소멸자가 동작한 후 기초클래스 소멸자가 동작한다.
- ④ 기초클래스의 소멸자는 일반 함수로, 파생클래스의 소멸자는 가상함수로 선언하는 것이 일반적이다.

23. STL의 컨테이너 중 연상 컨테이너에 해당되는 것은? (2점)

- ① vector ② list
- ③ map ④ deque

24. 다음과 같이 생성된 벡터에 대한 올바른 설명은? (3점)

```
vector<int> intVec(10);
```

- ① 벡터의 내용이 10으로 초기화된다.
- ② 10개의 int 값을 저장할 수 있으며, 값을 추가하면 크기가 확장될 수 있다.
- ③ 저장된 값은 앞에서부터 순차적으로만 읽을 수 있다.
- ④ 반복자를 사용할 수 없다.

25. C++의 예외처리에 대한 올바른 설명은? (3점)

- ① 예외의 발생을 알리기 위해 throw 명령을 사용한다.
- ② 예외가 발생할 가능성이 있는 부분은 catch 블록에 넣는다.
- ③ 예외 발생은 int형 값으로 알려야 한다.
- ④ try 블록에서는 예외의 유형에 따라 적절한 처리를 한다.