

2과목	C++ 프로그래밍	(36~60)
출제위원 : 방송대 이병래		
출제범위 : 교재 3장~8장 (해당 멀티미디어강의 포함)		

※ C++ 코드 중 문장 이해에 문제가 없는 부분은으로 생략하였음

※ (36~37) 다음은 함수 호출 문장의 int형 실 매개변수의 값을 서로 바꾸는 함수 swapValues이다. 다음 질문에 답하라.

```
(㉠) swapValues( (㉡) ) {  
    int tmp = x;  
    x = y;  
    y = tmp;  
}
```

36. (㉠)에 넣을 적절한 단어는? (2점)

- ① int
- ② double
- ③ int&
- ④ void

37. (㉡)에 작성할 형식 매개변수 선언문으로 적절한 것은? (2점)

- ① int x, y;
- ② int &x, int &y
- ③ int *x, int *y
- ④ const int &x, const int &y

38. 함수의 디폴트 인수에 대한 올바른 설명은? (2점)

- ① 호출 시 디폴트 값을 인수로 사용할 경우 생략할 수 있다.
- ② 디폴트 인수는 다른 인수의 앞에 있어야 한다.
- ③ 디폴트 인수는 상수만 전달할 수 있다.
- ④ 디폴트 값과 다른 값을 인수로 전달할 수 없다.

39. 다음과 같이 원형이 선언된 함수에 대한 올바른 설명은? (2점)

```
void func(const Coord2D& c, int x, int y=10);
```

- ① return 명령으로 임의의 자료형의 값을 반환할 수 있다.
- ② y의 값은 항상 10이 전달된다.
- ③ 함수에서 x의 값을 변경하면 해당되는 실매개변수의 값이 함께 변경된다.
- ④ 함수 안에서 c의 내용을 수정할 수 없다.

40. 다음 함수의 원형 (㉠)~(㉤) 중 아래 문장 (a)에 의해 호출되는 함수는? (3점)

```
int f(int x, int y); // (㉠)  
int f(float x, float y); // (㉡)  
int f(int x, double y); // (㉢)  
void f(int x); // (㉣)  
int a = f(10, 2.0); // (a)
```

- ① (㉠)
- ② (㉡)
- ③ (㉢)
- ④ (㉣)

※ (41~43) 다음은 계수를 지시하면 1씩 감소하는 하향계수기 클래스의 선언문이다. 질문에 답하라.

```
class DnCounter {  
    const int max; // 계수기의 최댓값  
    int value; // 계수기의 현재 값.  
public:  
    (㉠)  
    void reset() { value = max; }  
    void count() { value = value ? value-1 : max; }  
    int getValue() const { return value; }  
};
```

41. (㉠)에 넣을 DnCounter 클래스의 생성자 선언 문장으로 올바른 것은? (4점)

- ① DnCounter(int m) : max(m), value(m) {}
- ② int DnCounter(int m) { max = value = m; }
- ③ void DnCounter(int m) : value(m) { max = m; }
- ④ DnCounter(int m) : max(m) { return value = m; }

42. 문제 41의 생성자가 선언된 경우 DnCounter 클래스의 객체를 선언하는 올바른 문장은? (3점)

- ① DnCounter c;
- ② DnCounter c(9);
- ③ DnCounter c = 9;
- ④ DnCounter(9) c;

43. 다음 중 아래의 함수 g의 몸체 블록 (㉡)에서 사용할 수 있는 문장은 무엇인가? (가시성과 const에 유의하라.) (2점)

```
void g(const DnCounter& cnt) {  
    (㉡)  
}
```

- ① cnt.count();
- ② cnt.reset();
- ③ int n = cnt.value;
- ④ cout << cnt.getValue();

44. 다음 중 생성자에 대한 설명으로 올바른 것은? (4점)

- ① return 명령으로 값을 반환할 수 있다.
- ② 생성자는 매개변수를 포함할 수 없다.
- ③ 생성자는 다중정의를 할 수 있다.
- ④ 생성자에서는 데이터 멤버를 수정할 수 없다.

45. 다음 중 소멸자에 대한 설명으로 올바른 것은? (4점)

- ① 소멸자의 이름은 클래스 이름 앞에 ~를 붙인 것이다.
- ② return 명령으로 값을 반환할 수 있다.
- ③ 프로그래머의 호출 명령이 있어야 동작한다.
- ④ 매개변수를 통해 인수를 전달받을 수 있다.

※ (46~48) 다음 클래스에 대한 질문에 답하라.

ClassA.h	ClassA.cpp
<pre>class ClassA { int a double b; static int c; public: int f1(); static int f2(); };</pre>	<pre>#include "ClassA.h" int (㉠) f1() { } int (㉡) f2() { //static (㉢) } int (㉣) c = 0;</pre>

46. 클래스 ClassA의 멤버를 정의하기 위해 (㉠)에 필요한 내용은 무엇인가? (3점)

- ① this->
- ② ClassA->
- ③ ClassA::
- ④ public:

47. 데이터 멤버 c에 대한 올바른 설명은? (3점)

- ① 각각의 객체마다 각자의 c가 있다.
- ② 객체가 만들어져야 c에 데이터를 저장할 수 있다.
- ③ c에 저장된 값을 변경할 수 없다.
- ④ 클래스에 속하는 모든 객체들이 공유하는 데이터 멤버이다.

48. 다음 중 static 멤버함수의 몸체인 (㉡)에 포함될 수 있는 문장은? (2점)

- ① c = f1();
- ② c = 0;
- ③ return a * a;
- ④ a = static_cast<int>(b);

※ (49~50) 다음은 연필의 개수를 12자루 묶음의 타와 날개로 표현하는 클래스이다. 다음 질문에 답하라.

```
class Pencils {
    int dozens, np;           // 타, 날개
public:
    .....                    // 생성자(생략)
    Pencils operator ++ ();
    Pencils operator ++ (int);
    Pencils operator + (const Pencils& p);
    Pencils operator * (int n);
    .....

};



---


(7) {
    if (++np >= 12) ++dozens, np = 0;
    return *this;
}
```

49. 다음 중 (가)에 넣을 적절한 구문은 무엇인가? (2점)

- ① Pencils Pencils::operator ++ ()
- ② Pencils Pencils::operator ++ (int)
- ③ Pencils Pencils::operator + (const Pencils& p)
- ④ Pencils Pencils::operator * (int n)

50. 위 내용만으로 볼 때 다음 중 Pencils 객체 p1과 p2를 대상으로 사용할 수 있는 연산에 해당되지 않는 것은? (2점)

- | | |
|-----------|----------|
| ① p1++ | ② ++p1 |
| ③ p1 + p2 | ④ 5 * p1 |

51. 다음 중 클래스에 대한 설명으로 올바른 것은? (4점)

- ① 파생클래스는 기초클래스의 속성과 메소드를 이어받는다.
- ② 가시성을 지시하지 않은 멤버는 public 멤버이다.
- ③ 기초클래스의 멤버함수에서 파생클래스의 public 멤버를 액세스할 수 있다.
- ④ 기초클래스의 protected 멤버는 파생클래스의 멤버함수에서 액세스할 수 없다.

52. 파생클래스 객체가 만들어지고 제거될 때 파생클래스와 기초 클래스 생성자 및 소멸자의 동작 순서를 올바르게 설명한 것은?
(3점)

- ① 생성자와 소멸자 모두 기초클래스의 것이 먼저 동작한다.
- ② 생성자와 소멸자 모두 파생클래스의 것이 먼저 동작한다.
- ③ 파생클래스의 소멸자 동작 후 기초클래스의 소멸자 동작
- ④ 파생클래스의 생성자 동작 후 기초클래스의 생성자 동작

※ (53~55) Person 클래스와 Person의 파생클래스인 Student에 대한 다음 질문에 답하라.

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

class Student : (㉡) Person {
    string school;
public:
    Student(const string &n, const string &s)
        (㉢) // 생성자
    string getSchool() const { return school; }
    void print() const {
        cout << school << "에 다니는 " << getName();
    }
};
```

53. Person 클래스의 public 멤버가 Student의 public 멤버로 상속 되도록 하기 위해 (㉠)에 넣을 내용은? (3점)

- ① friend ② protected
③ public ④ 공란

54. Student 클래스의 생성자를 선언하기 위해 (ㄷ)에 넣을 적절한 구문은? (3점)

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

55. 다음 문장이 student의 print를 호출하도록 하려면 (가)에 어떤 단어를 넣어야 하는가? (3점)

```
Person *p = new Student("김철수", "방송대");
p->print();
```

- ① friend ② virtual
③ public ④ 공란

56. 추상클래스와 상세클래스에 대한 올바른 설명은? (4점)

- ① 추상클래스로 직접 객체를 정의할 수 있다.
- ② 상세클래스는 순수가상함수를 포함하지 않아야 한다.
- ③ 상세클래스는 객체를 정의할 수 없다.
- ④ 상세클래스는 그 클래스의 파생클래스들이 반드시 가지고 있어야 할 행위를 누락하지 않도록 하는 역할을 한다.

57. 다음은 템플릿 매개변수에 지정된 자료형과 개수에 따라 데이터를 저장하는 클래스 템플릿이다. double형 값을 50개 저장하는 컨테이너 a를 정의하는 올바른 문장은? (2점)

```
template <class T, int size> class ArrT {
    T arr[size];
    .....
};
```

- ① ArrT<double, 50> a; ② ArrT<double> a[50];
③ ArrT a(double)[50]; ④ ArrT a[50];

※ (58~59) 다음 프로그램에 대한 질문에 답하라.

```
vector<int> a(10);
vector<int>::iterator it;
int i=0;
for ( _____ (7) _____ )
    *it = i++;
```

58. a에 대한 설명으로 올바른 것은? (3점)

- ① 연상 컨테이너에 해당된다.
- ② 원하는 위치에 직접 접근할 수 없다.
- ③ 키를 이용한 검색 기능을 제공한다.
- ④ 배열과 유사하나 실행 중에 필요하다면 a의 용량을 확장할 수 있다.

59. a의 선두에서 시작하여 차례로 0부터 9까지의 값이 저장되게 하려고 한다. (㉠)에 넣을 적절한 내용은? (2점)

- ① (i = 0 ; i < 10 ; i++)
- ② (it = a[0] ; it < a[10] ; it++)
- ③ (it = a.begin() ; it < a.end() ; it++)
- ④ (; i < 10 ;)

60. 다음 중 프로그램 실행 중 예외 발생하였음을 알리기 위해 사용하는 명령은? (3점)

- ① friend ② throw
③ try ④ switch