

※ (49~51) 다음 클래스에 대한 질문에 답하라. 단, (가)은 한 개의 정수를 형식 매개변수 v에 전달받아 그 값을 저장하는 객체를 생성하는 생성자의 머리가 작성될 곳이다.

```

1  class IntClass {
2      int* iPt;
3  public:
4                          (7)                     { iPt = new int;    *iPt = v; }
5                          (8)                    
6          { iPt = new int;    *iPt = *i.iPt; }
7                          (9)                     { delete iPt; }
8      const IntClass& operator = (const IntClass& i) {
9          iPt = new int;    *iPt = *i.iPt;
10         return *this;
11     }
12     IntClass operator ++ (int)
13         {                                     (10)                                     }
14     int getValue() const { return *iPt; }
15 };

```

49. 복사 생성자를 만들기 위해 (L)에 넣을 내용은? (3점)

- ① `IntClass(const IntClass& i)`
- ② `IntClass(IntClass i)`
- ③ `int IntClass(IntClass i)`
- ④ `void IntClass(const int* i)`

50. 소멸자를 작성하기 위해 (ㄷ)에 넣을 내용은? (3점)

- ① IntClass()
- ② ~IntClass(IntClass& obj)
- ③ int ~IntClass() const
- ④ ~IntClass()

51. 12~13행의 연산자 다중정의를 완성하기 위해 (ㄹ)에 넣을 문장은? (2점)

- ① (*iPt)++
- ② return IntClass((*iPt)++);
- ③ ++(*iPt)
- ④ return IntClass(++(*iPt));

52. 다음 중 클래스의 계층구조를 설계할 경우 나머지 클래스의 기초 클래스가 될 수 있는 것은? (4점)

- ① 선박 ② 화물선
③ 여객선 ④ 어선

※ (53~55) 다음 클래스에 대한 질문에 답하라.

<pre> class BClass { <u>(7)</u>: int x; public: void init(int a) { x = a; } <u>(L)</u> void print() const { cout << x; } }; </pre>	<pre> class DClass : public BClass { int y; public: void init(int a, int b) { x = a; y = b; } void print() const { cout << x << " " << y; } }; </pre>
--	---

53. BClass의 x는 타 클래스에서는 직접 사용할 수 없지만 파생클래스인 DClass의 멤버함수에서 직접 사용할 수 있게 하려고 한다. (7)에 어떠한 가시성 지시어를 사용해야 하는가? (3점)

- ① private ② public
③ static ④ protected

54. 다음 프로그램에서 (가)의 문장이 “10 20”을 출력하게 하기 위해 (ㄴ)에 넣을 단어는? (3점)

```
DClass *pD = new DClass;
BClass *pB = pD;
pD->init(10, 20);
pB->print();           // (7)
```

- ① virtual ② static
③ inline ④ protected

55. BClass의 포인터 p1을 DClass의 포인터 p2로 다운 캐스팅하려고 한다. p1이 가리키고 있는 객체가 DClass의 객체가 아닐 경우 p2에 NULL이 저장되도록 하려면 어떤 형변환 연산자를 사용해야 하는가? (2점)

- ① dynamic_cast ② static_cast
③ reinterpret_cast ④ const_cast

56. 다음 클래스에 대한 설명으로 옳바른 것은? (4점)

```
class ClassA {
    .....
    virtual int f() = 0;
};
```

- ① ClassA는 상세클래스이다.
- ② 함수 f는 항상 0을 반환한다.
- ③ ClassA의 파생클래스는 f를 재정의 할 수 없다.
- ④ ClassA의 객체를 만들 수 없다.

57. 다음은 두 매개변수의 값을 서로 교환하는 함수 템플릿을 선언한 문장이다. (㉠)에 넣을 적절한 내용은? (3점)

(\neg)

void swapValues(T& a, T& b) {

T tmp = a;
 a = b;
 b = tmp;

}

- ① template
- ② template <ANY>
- ③ template <class T>
- ④ template::

58. 다음 중 STL에서 포인터와 같은 용도로 사용되는 것은? (2점)

- ① 알고리즘
- ② 반복자(iterator)
- ③ 컨테이너(container)
- ④ deque

59. 다음 중 정렬된 데이터를 저장하는 두 컨테이너를 순서를 유지하여 하나로 합병하기 위해 사용되는 STL 알고리즘의 함수는? (2점)

- ① multiset ② sort
③ for_each ④ merge

60. 다음 프로그램에서 catch 블록이 실행되려면 함수 f()에서 어떤 명령을 실행해야 하는가? (4점)

```
try { f(); }
catch (int n) { cout << "예외 발생" << endl; }
```

- ① return 0; ② throw 1;
③ throw "예외"; ④ exit(0);

--	--