

48. (ㄹ)의 내용을 작성하는 것과 관련한 설명으로 옳바른 것은?
(2점)

- ① (ㄹ)에서 + 연산의 좌측 피연산자는 b이다.
- ② (ㄹ)에서 + 연산의 우측 피연산자는 b이다.
- ③ (ㄹ)에서 *this는 b이다.
- ④ (ㄹ)에서 *this의 내용이 + 연산의 결과로 바뀐다.

49. Class2의 private 멤버를 Class1의 멤버함수에서 액세스할 수 있도록 허용하려고 한다. ()에 넣을 적절한 단어는? (3점)

```
class Class2 {
  _____ (7) class Class1;
  .....
};
```

- ① public ② inline
③ private ④ friend

50. 클래스의 static 멤버에 대한 설명으로 올바른 것은? (2점)

- ① static 멤버함수 안에서 소속 클래스의 static이 아닌 멤버함수를 호출할 수 있다.
- ② static 멤버함수 안에서 소속 클래스의 static이 아닌 데이터 멤버의 값을 읽을 수 있다.
- ③ static 데이터 멤버는 각각의 객체마다 하나씩 존재한다.
- ④ 그 클래스의 객체가 만들어지지 않았어도 사용할 수 있다.

※ (51~54) 클래스 Person과 Person의 파생 클래스인 Athlete에
대한 질문에 답하라.

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

```
class Athlete : _____ (㉔) {
    string team;
public:
    Athlete(const string& n, const string& t)
        : Person(n), team(t) {}
    string getTeam() const { return team; }
    void print() const {
        cout << team << " 소속 " << name;    // (㉕)
    }
};
```

```
int main()
{
    Person *pP1 = new Person("홍길동");
    Athlete *pA = new Athlete("박철수", "승리구단");
    Person *pP2 = pA;

    pP1->print();          cout << "  ";
    pA->print();           cout << "  ";
    pP2->print();          cout << endl;
    delete pP1;           delete pP2;
    return 0;
}
```

51. Person의 public 멤버가 Athlete의 public 멤버로 상속되도록 하기 위해 (㉔)에 넣을 내용은? (4점)

- ① public Person ② private Person
③ public Athlete ④ static Person

52. Person의 멤버 name을 (≒)의 문장처럼 Athlete의 멤버함수에서 사용할 수 있고, main함수에서는 name을 직접 사용할 수 없도록 하려면 ()에 어떠한 내용이 필요하가? (3점)

- ① private: ② protected:
③ public: ④ default:

53. (ㄴ)이 공란일 경우 main의 출력 결과는? (3점)

- ① 홍길동 승리구단 소속 박철수 승리구단 소속 박철수
② 홍길동 박철수 박철수
③ 홍길동 승리구단 소속 박철수 박철수
④ 홍길동 박철수 승리구단 소속 박철수

54. (ㄴ)에 virtual 이라는 단어를 넣었을 때 main의 출력 결과는?
(3점)

- ① 홍길동 승리구단 소속 박철수 승리구단 소속 박철수
② 홍길동 박철수 박철수
③ 홍길동 승리구단 소속 박철수 박철수
④ 홍길동 박철수 승리구단 소속 박철수

55. 다음 클래스에 대한 설명으로 옳바른 것은? (4점)

```
class absClass {
    .....
    virtual int funcA(int a) = 0;
};
```

- ① funcA의 형식매개변수 a의 초기값은 0이다.
- ② absClass는 추상클래스로서, 객체를 직접 만들 수 없다.
- ③ absClass가 기초클래스인 파생클래스를 선언할 수 없다.
- ④ absClass의 파생클래스에서 funcA를 재정의하면 absClass는 상세클래스가 되어 객체를 만들 수 있게 된다.

56. 다음과 같이 선언된 클래스 템플릿을 이용하여 int형 값을 10개 저장하는 컨테이너 a를 정의하는 문장은? (2점)

```
template <class T, int size> class Array {
    T buf[size];
    .....
};
```

- ① `Array<int> a;` ② `Array<10> int a;`
③ `Array<int, 10> a;` ④ `Array a<10>;`

57. 다음 STL 컨테이너 중 배열과 같이 활용할 수 있으며, 필요에 따라 그 크기를 확장할 수 있는 것은? (3점)

- [illegible]

58. STL에서 반복자(iterator)의 용도는 무엇인가? (3점)

- ① 포인터 ② 데이터의 임시 저장
③ 처리결과의 반환 ④ 컨테이너의 크기 지정

※ (59~60) 다음 클래스에 대한 질문에 답하라.

```
double g(double x) {
    if (x <= 0)
        throw 0;
    return sqrt(x);
}

void f(double a) {
    (T) {
        cout << g(a) << endl;
    }
    catch ((L)) {
        cout << "예외 발생" << endl;
    }
}
```

59. 예외 처리를 위해 ()에 넣을 적절한 단어는 무엇인가? (3점)

- [illegible]

60. 함수 g()에서 예외가 발생할 경우 “예외 발생”이라고 출력하기 위해 ()에 넣을 내용은? (2점)

- [illegible]