cpp_08_프렌드와연산자중복

□ 개념 확인 학습

1. 다음 질문에 O, X로 답하세요.

A. friend로 선언되어 있으면 클래스의 모든 멤버를 접근할 수 있는 권한이 부여된다. ( )

B. friend 함수는 멤버 함수 이므로 상속이 가능하다. ( )

C. 전역함수만 friend 함수로 사용할 수 있다. ( )

D. 모든 연산자를 중복할 수 있다. ( )

E. 연산자 함수를 이용해 연산자에 새로운 의미를 정의할 수 있다. ( )

F. 연산자 함수 중복 시 디폴트 매개변수는 사용할 수 없다. ( )

G. 피 연산자의 개수를 변경하여 연산자 함수를 중복할 수 있다. ( )

H. 연산자 함수는 멤버 함수로도 외부 함수로도 작성할 수 있다. ( )

I. 객체 a의 모든 요소에 2을 더해 객체 b로 저장하는 b=2+a 연산은 외부함수로 작성해 friend 로 사용한다. ( )

2. Exam 클래스의 주석에 해당하는 프렌드 선언문을 작성하세요.

```
bool isCheck(int i, Exam e);

class ExamMember {
   void em_method(Exam e);
}
class Exam {
   //외부함수를 isCheck()를 프렌드로 선언

   //ExamMember 클래스의 멤버 함수 em_method()를 프렌드로 선언

   //ExamMember 클래스 전체를 프렌드로 선언
}
```

3. Power 객체가 다음과 같을 경우, Power 객체 각 멤버에 대한 다음 연산을 위한 연산자 함수를 작성하세요.

슬라이드에 있음.

```
class Power { //에너지를 표현하는 파워 클래스
    int kick; //발로 차는 힘
    int punch; //주먹으로 치는 힘
public:
    Power(int kick=0, int punch=0) {
        this->kick = kick;
        this->punch = punch;
    }
};
```

A. c = a + b; //a, b, c는 Power 객체 //멤버 함수로 작성 P.20

B. c = a + b; //a, b, c는 Power 객체 //프렌드로 사용할 연산자 함수로 작성 P.40

C. b = a + 2; //a, b는 Power 객체 //멤버 함수로 작성 P.21

D. b = 2 + a; //a, b는 Power 객체 //프렌드로 사용할 연산자 함수로 작성 P.38

E. a == b; //a, b는 Power 객체 //멤버 함수로 작성 P.23

F. a += b; //a, b, c는 Power 객체 //멤버 함수로 작성 P.25

G. a = b; //a, b는 Power 객체 //멤버 함수로 작성 P.26, 27

H. p = createObject(); //a는 Power 객체 //멤버 함수로 작성 이동대입 P.28

I. ++a; //a는 Power 객체 //멤버 함수로 작성 P.32

J. ++a; //a는 Power 객체 //프렌드로 사용할 연산자 함수로 작성 P.42

K. a++; //a는 Power 객체 //멤버 함수로 작성 P.34

L. a++; //a는 Power 객체 //프렌드로 사용할 연산자 함수로 작성 P.42

M. ! //a는 Power 객체, 모든 멤버 값이 0이면 true리턴 //멤버 함수로 작성 P.35

4. Power 객체가 다음과 같을 경우, Power 객체 각 멤버에 대한 *연산이 가능하도록 연산자 함수를 멤버함수와 외부 함수로 각각 제시하세요.

```
class Power { //에너지를 표현하는 파워 클래스
    int kick; //발로 차는 힘
    int punch; //주먹으로 치는 힘
public:
    Power(int kick=0, int punch=0) {
        this->kick = kick;
        this->punch = punch;
    }
    void show(string obj);
};
void Power::show(string obj) {
    cout << obj << ") kick=" << kick << ',' << "punch=" << punch << endl;
}
int main(){
    Power a(3,5), b(4,6), c;
    c = a * b;
    a.show("a");
    b.show("b");
    c.show("c");
}
```

```
a) kick=3,punch=5
b) kick=4,punch=6
c) kick=12,punch=30
```

5. Power 객체의 kick과 punch에 2^n 곱하기 연산을 수행하는 << 연산자를 멤버 함수로 작성하세요.

```
int main() {
    Power a(2,3);
    a << 3;
    a.show("a");

    Power b(1,5);
    b << 1;
    b.show("b");
}
```

```
a) kick=16,punch=24
b) kick=2,punch=10
```

□ 응용 프로그래밍

6. Book 객체에 대하여 다음과 같은 연산을 수행할 수 있도록 연산자 함수를 멤버 함수로 구현하고 프로그램을 완성 하세요.

```
class Book {
```

```
    string title;
    int price;
public:
    Book(string title = "", int price = 0);
    void show(string obj);
    string getTitle();  //title 반환
};

int main() {
    Book a("청춘", 20000), b("미래", 30000);

    a += 500;  //책 a의 가격 500원 증가
    b -= 500; //책 b의 가격 500원 감소
    a.show("a");
    b.show("b");

    Book c("명품 C++", 30000), d("고품 C++", 30000);

    if (c == 30000) cout << "명품 C++ 정가 30000원" << endl; //price 비교
    if (c == "명품 C++") cout << "명품 C++ 입니다." << endl; //책 title 비교
    if (c == d) cout << "두 책이 같은 책입니다." << endl; // title, price 모두 비교
    else cout << "두 책이 다른 책입니다." << endl;
}
```

```
a) title=청춘, price=20500
b) title=미래, price=29500
명품 C++ 정가 30000원
명품 C++ 입니다.
두 책이 다른 책입니다.
```

7. Matrix 클래스에 대하여 다음과 같은 연산이 가능하도록 연산자 함수를 멤버 함수로 구현하고 프로그램을 완성하세요.

```
class Matrix {
    int ar[4];
public:
    Matrix(int a1 = 0, int a2 = 0, int b1 = 0, int b2 = 0);
    void show(string name);
};
int main() {
    Matrix a(1, 2, 3, 4), b(2, 3, 4, 5), c;
    c = a + b;
    a.show("a");
    b.show("b");
    c.show("c");

    a += b;
    a.show("a");

    int x[4], y[4] = {5, 6, 7, 8};
    a >> x; // a의 각 원소를 배열 x에 복사.
    b << y; // 배열 y의 원소 값을 b의 각 원소에 설정

    cout << "x = { ";
    for (int i = 0; i < 4; i++)
```

```
a = { 1 2 3 4 }
b = { 2 3 4 5 }
c = { 3 5 7 9 }
a = { 3 5 7 9 }
x = { 3 5 7 9 }
b = { 5 6 7 8 }
```

```
        cout << x[i] << ' '; // x[] 출력
    cout << "}" << endl;
    b.show("b");
}
```

8. Circle 클래스에 대하여 다음과 같은 연산이 가능하도록 연산자 함수를 멤버 함수로 구현하고 프로그램을 완성하세요.

```
class Circle {
   int radius;
public:
   Circle(int radius = 0);
   void show(string name);
};

int main() {
    Circle a(5), b(4);
    a.show("a");
    b.show("b");

   ++a; // 반지름을 1 증가 시킨다.
    a.show("a");

    b = a++; // 반지름을 1 증가 시킨다.
    a.show("a");
    b.show("b");

    b = a + 3; // b의 반지름을 a의 반지름에 3을 더한 것으로 변경
    b.show("b");
}
```

9. [ ]연산자를 멤버 함수로 정의하여 제시된 결과처럼 실행하는 프로그램을 작성하세요.

```
class Array {
    double *ptr;
    int size;
public:
    Array(int size);
    ~Array();
    void show(string name);
    //[] 연산자 중복
    //= 연산자 중복
};

int main() {
    int size;
    cout << "array size ? ";
    cin >> size;
```

```
    Array arr(size), brr(size);

    for (int i = 0; i < size; i++) {
        cout << i << ") input>> ";
        cin >> arr[i];
    }
    arr.show("arr");

    brr = arr;
    brr.show("brr");

    brr[2] = 34.5;
    brr[4] = 56.3;
    arr.show("arr");
    brr.show("brr");

    return 0;
}
```

```
PS E:\lecture_src\cpp_src> ./a
array size ? 4
0) input>> 6
1) input>> 5
2) input>> 4
3) input>> 3
arr = { 6  5  4  3  }
brr = { 6  5  4  3  }
인덱스 범위 초과 오류
PS E:\lecture_src\cpp_src> ./a
array size ? 6
0) input>> 7
1) input>> 6
2) input>> 5
3) input>> 4
4) input>> 3
5) input>> 2
arr = { 7  6  5  4  3  2  }
brr = { 7  6  5  4  3  2  }
arr = { 7  6  5  4  3  2  }
brr = { 7  6  34.5  4  56.3  2  }
```

10. 정수 배열을 항상 증가 순으로 유지하는 StoredArray 클래스를 작성하려고 합니다. 아래의 main() 함수가 동작할 수 있도록 SortedArray 클래스를 작성하고 +와 =연산자도 작성하세요.

```
class SortedArray {
    int size;      //현재 배열의 크기
    int *p;        //정수 배열에 대한 포인터
    void sort(); //정수 배열을 오름차순으로 정렬
public:
    SortedArray();                          //p는 nullptr로 size는 0으로 초기화
    SortedArray(SortedArray &arr);  //복사 생성자
    SortedArray(int arr[], int size);   //생성자. 정수 배열과 크기를 전달받아 p에 저장 후 sort() 호출
    ~SortedArray();                         //소멸자
    SortedArray operator+(SortedArray &b);
    SortedArray &operator=(const SortedArray &b); //현재 배열에 b 배열을 복사
    void show(string name);                                 //배열의 원소 출력
};

void SortedArray::sort() { //오름차순 버블 정렬
    if (p == nullptr || size == 0)
        return;

    for (int i = 0; i < size - 1; i++) {
        for (int j = i; j < size - 1; j++) {
            if (p[j] > p[j + 1]) {
                int temp = p[j];
                p[j] = p[j + 1];
                p[j + 1] = temp;
            }
        }
    }
```

```
}

int main() {
    int n[] = {2, 20, 6};
    int m[] = {10, 7, 8, 30};
    SortedArray a(n, 3), b(m, 4), c;

    c = a + b;

    a.show("a");
    b.show("b");
    c.show("c");
}
```

```
a = { 2 6 20 }
b = { 7 8 10 30 }
c = { 2 6 7 8 10 20 30 }
```