

# 11. 객체 전달과 참조, 복사생성자

# 실습내용

---

- 실습#11 : 객체 전달과 참조, 복사생성자
- 파일명 : 실습과제#11\_2022199002\_홍길동

## [실습 11-1]

- 2개의 Circle 객체를 교환하는 swap 함수를 '참조에 의한 호출'이 되도록 작성하고 호출하는 프로그램을 작성하시오.

```
class Circle {  
private:  
    int radius;  
public:  
    Circle() { radius = 1; }  
    Circle(int radius) { this->radius = radius; }  
    double getArea() { return 3.14*radius*radius; }  
};
```

<실행 예시>

```
A의 면적 = 17.27  B의 면적 = 28.26  
A의 면적 = 28.26  B의 면적 = 17.27
```

## [실습 11-2]

- 키보드로부터 반지름 값을 읽어 Circle 객체에 반지름을 설정하는 전역함수인 `readRadius()` 함수를 구현하여 전체 프로그램을 완성하시오.

```
class Circle {  
    int radius;  
public:  
    Circle() { radius = 1; }  
    Circle(int radius) { this->radius = radius; }  
    void setRadius(int radius) { this->radius = radius; }  
    double getArea() { return 3.14*radius*radius; }  
};
```

```
int main() {  
    Circle donut;  
    readRadius(donut);  
    cout << "donut의 면적 = " << donut.getArea() << endl;  
}
```

정수 값으로 반지름을 입력하세요>>3  
donut의 면적 = 28.26

## [실습 11-3]

- 클래스 Accumulator는 add() 함수를 통해 계속 값을 누적하는 클래스로서, 다음과 같이 선언된다. Accumulator 클래스를 구현하여 전체 프로그램을 완성하시오.
- 체이닝(연쇄호출): Accumulator& add(int n) 함수는 자기 자신(객체)을 참조로 리턴 하는 함수

```
class Accumulator {  
    int value;  
public:  
    Accumulator(int value); // 매개변수 value로 멤버 value를 초기화한다.  
    Accumulator& add(int n); // value에 n을 더해 값을 누적한다.  
    int get(); // 누적된 값 value를 리턴한다.  
};
```

```
int main() {  
    Accumulator acc(10);  
    acc.add(5).add(6).add(7); // acc의 value 멤버가 28이 된다.  
    cout << acc.get() << endl; // 28 출력  
}
```

## [실습 11-4]

- 책의 이름과 가격을 저장하는 다음 Book 클래스이다. Book 클래스의 생성자, 소멸자, set() 함수를 구현하고, 실행 결과를 참조하여 깊은 복사생성자를 작성하시오.

```
class Book {  
    char *title; // 제목 문자열  
    int price; // 가격  
public:  
    Book(char* title, int price);  
    ~Book();  
    void set(char* title, int price);  
    void show() { cout << title << ' ' << price << "원" << endl; }  
};
```

```
int main() {  
    Book cpp("명품C++", 10000);  
    Book java = cpp;  
    java.set("명품자바", 12000);  
    cpp.show();  
    java.show();  
}
```

명품C++ 10000원  
명품자바 12000원

## [실습 11-5]

- 다음 코드에서 출력 결과를 쓰고, 각 출력이 왜 그렇게 되는지 메모리 동작 관점에서 설명하시오.  
또한, 이 코드의 잘못된 점이 있다면 설명하고 수정 방법을 서술하시오.

```
class Person {  
    char* name;  
  
public:  
    Person(const char* n) {  
        name = new char[strlen(n) + 1];  
        strcpy(name, n);  
        cout << "생성자 호출: " << name << endl;  
    }  
  
    Person(const Person& p) {  
        name = p.name;  
        cout << "복사 생성자 호출: " << name << " (얕은 복사)" << endl;  
    }  
  
    ~Person() {  
        cout << "소멸자 호출: " << name << endl;  
        delete[] name;  
    }  
  
    void changeName(const char* newName) {  
        strcpy(name, newName);  
    }  
  
    void show() {  
        cout << "이름: " << name << endl;  
    }  
};
```

```
void introduce(Person p) {  
    cout << "[introduce 함수] ";  
    p.show();  
    p.changeName("Alex");  
    cout << "[introduce 함수 변경 후] ";  
    p.show();  
}  
  
Person createPerson() {  
    Person temp("Temporary");  
    return temp;  
}  
  
int main() {  
    Person a("Charlie");  
    Person b = a;  
  
    cout << "[main] b 이름 변경 전: ";  
    b.show();  
    b.changeName("Bob");  
    cout << "[main] b 이름 변경 후: ";  
    b.show();  
    cout << "[main] a 이름 확인: ";  
    a.show();  
  
    introduce(a);  
  
    Person c = createPerson();  
  
    return 0;  
}
```