



Object-Oriented Programming Report

Assignment 3-3

Professor	Donggyu Sim
Department	Computer engineering
Student ID	2022202061
Name	Seoeun Yang
Class (Design / Laboratory)	1 / B (미수강시 0로 표기)
Submission Date	2023. 5. 19

Program 1

□ 문제 설명

클래스 상속을 사용하는 문제이다. virtual function, 즉 가상함수를 사용하는데 파생 클래스의 객체를 참조하면 해당 객체의 가상 함수를 호출하고 실행한다. 이는 프로그램이 실행되면서 결정된다. 두 물건의 가격을 입력 받고 할인율을 입력 받는다. 할인율이 적용된 두 가격의 가격을 비교해 더 싼 물건과 할인율이 적용된 두 물품의 가격 차를 출력한다.

□ 결과 화면

```
Microsoft Visual Studio 디버그 콘솔
=====
Price Compare Program
=====
Insert item1 price: $16
Insert item2 price: $12
-----
Insert discount percent: 25%
-----
Result:
Discount of item1 is cheaper.
Saving discount price is $3.0
```

문제지 예시

```
Microsoft Visual Studio 디버그 콘솔
=====
Price Compare Program
=====
Insert item1 price: $10
Insert item2 price: $15
-----
Insert discount percent: 10%
-----
Result:
Discount of item1 is cheaper.
Saving discount price is $4.5
C:\Users\W82108\source\repos\OOP_20
```

할인율 적용 가격: item1 - \$9, item2 - \$13.5
따라서 item1 을 구입하고 \$4.5 를 아낄 수 있다.

```
Microsoft Visual Studio 디버그 콘솔
=====
Price Compare Program
=====
Insert item1 price: $4
Insert item2 price: $4
Same Price
C:\Users\82108\source\repos\OOP_예외 처리
```

예외 처리

□ 고찰

virtual function 의 개념이 너무 생소해 이해하기가 어려웠다. Bill() 함수가 parent class 에서 virtual 로 선언되어 있기 때문에 derived class 에서 Bill() 함수를 정의해줘야 한다. 또한 Saving(), operator<에서는 할인율이 적용된 가격을 사용해야 하기 때문에 Bill()함수에서 연산한 할인율이 적용된 가격을 불러와 함수를 실행한다. 또한 두 물건의 가격이 동일하면 어떤 할인율을 적용해도 가격이 동일하기 때문에 예외 처리를 해줬다.

✚ Program 2

□ 문제 설명

클래스 상속을 사용하는 문제이다. parent class 로 Person class 를 생성해 이름과 나이를 저장한다. derived class 로는 professor class 와 student class 를 생성해 각각 필요한 정보들을 저장한다. Person class 에 순수가상함수 Say()가 존재한다. derived class 에서 Say()함수를 정의해주어 호출한다. 이 문제에서는 저장한 정보들을 출력하는 데에 사용된다.

□ 결과 화면

```
Enter student information:
Name: Seoeun YANG
Age: 21
StudentNum: 2022202061
Major: Computer Engineering
SchoolYear: 2

Enter professor information:
Name: Minho KIM
Age: 38
ProfessorNum: 1234
Major: Computer Engineering

<Student Information>
I'm a student of KW University.
My name is Seoeun YANG.
My student number is 2022202061.
I'm 21 years old and I'm a sophomore.
I'm majoring in Computer Engineering.

<Professor Information>
I'm a professor of KW University.
My name is Minho KIM.
My professor number is 1234.
I'm 38 years old.
I'm majoring in Computer Engineering.
```

학생, 교수 정보 출력

```
cout << "I'm " << getAge() << " years old and I'm a ";
if (getSchoolYear() == 1)
    cout << "freshman." << endl;
else if (getSchoolYear() == 2)
    cout << "sophomore." << endl;
else if (getSchoolYear() == 3)
    cout << "junior." << endl;
else
    cout << "senior." << endl;
```

학년에 따라 다르게 출력됨.

□ 고찰

부모 클래스의 Say()함수는 순수 가상 함수이기 때문에 추상 클래스가 되어 main()함수에서 직접적으로 객체를 생성할 수 없다. 따라서 Say()함수를 자식 클래스에서 overriding을 꼭 해야 한다. main()함수에서 공백 포함 입력을 받기 위해 getline()함수를 사용했는데 엔터키로 구분하기 때문에 오류가 발생했다. 따라서 'wn'를 예외처리를 해주면서 입력을 받도록 수정했다.

Program 3

□ 문제 설명

3*3 matrix 에서 +=, -=, * operator 을 구현하는 문제이다. 이는 operator overloading 을 사용하는데 이는 사용자가 연산자의 의미를 재정의하는 것을 의미한다. 따라서 같은 연산자에 2 가지 이상의 의미가 내포된다. 컴파일러는 피연산자의 형식을 검사하여 연산자의 의미를 구분한다. matrix 의 숫자는 double 형으로 실수를 포함해 계산한다.

□ 결과 화면

```
Microsoft Visual Studio - C++ 콘솔
enter number of matrix1 :
1.5 4 5
2 1 3
1 1 0

enter number of matrix2 :
2 5 7.8
1 6 5
2 2 3

result of += operator
3.5 9 12.8
3 7 8
3 3 3

result of -= operator
-0.5 -1 -2.8
1 -5 -2
-1 -1 -3

result of * operator
3 20 39
2 6 15
2 2 0
```

matrix1, matrix2 입력 받아 연산자 실행 결과 출력

□ 고찰

operator 을 사용자가 직접 정의할 수 있다는 게 가장 신기하고 재미있었던 부분인 것 같다. 행렬을 계산할 때 이중 for 문을 사용해 일일이 계산해줘야 했다면 재정의하고 나서는 main()함수에서 호출만 하면 바로 계산해주는 점이 편리한 것 같다. 사용자로 하여금 좀 더 편할 수 있도록 해주는 기능인 것 같다. 실제 프로젝트나 스스로 공부할 때 적극적으로 사용하면 좋을 것 같다.

Program 4

□ 문제 설명

더블 링크드 리스트를 구현하는 문제이다. 이중 연결 리스트로 포인터 2 개를 사용해 노드 앞뒤를 연결한다. 이 문제는 사용자로부터 점수들을 입력 받아 그 평균을 계산해 연결 리스트에 저장하는 문제이다. 노드를 삽입할 때 오름차순으로 저장하는데 출력할 땐 사용자로부터 입력을 받아 오름차순 혹은 내림차순으로 출력한다.

□ 결과 화면

```
Microsoft Visual Studio 디버그 콘솔
1. insert 2. print 3. exit : 1 23 54 34 76 45 87
1. insert 2. print 3. exit : 1 65 34 12 68 98 56
1. insert 2. print 3. exit : 2
1. ascending 2. descending : 1
53.1667 55.5
1. insert 2. print 3. exit : 2
1. ascending 2. descending : 2
55.5 53.1667
1. insert 2. print 3. exit : 1 54 23 12 11 9 32
1. insert 2. print 3. exit : 2
1. ascending 2. descending : 1
23.5 53.1667 55.5
1. insert 2. print 3. exit : 2
1. ascending 2. descending : 2
55.5 53.1667 23.5
1. insert 2. print 3. exit : 3
```

3 개의 평균 저장

□ 고찰

이전 과제들을 풀면서 충분히 연결 리스트에 대해 익숙해진 상태라 구현이 어렵지는 않았다. 평균을 구할 때 몇 개의 점수가 입력될 지 모르기 때문에 char 형으로 입력을 받고 공백을 기준으로 문자열을 잘라 double 형으로 형변환을 해주었다. 이를 통해 평균을 계산하고 연결 리스트에 삽입했다. 오름차순으로 저장해야 하기 때문에 중간 삽입 기능 또한 구현했다.