

REPORT

보고서 작성 서약서

1. 나는 타학생의 보고서를 복사(Copy)하지 않았습니다.
2. 나는 타학생의 보고서를 인터넷에서 다운로드 하여 대체하지 않았습니다.
3. 나는 타인에게 보고서 제출 전에 보고서를 보여주지 않았습니다.
4. 보고서 제출 기한을 준수하였습니다.

나는 보고서 작성시 위법 행위를 하지 않고,
성.균.인으로서 나의 명예를 지킬 것을 약속합니다.

과 목 : 프로그래밍 기초와 실습

과 제 명 : 실습 1 : Operator

담당교수 : 민 형 복 교수

학 과 : 자연과학계열

학 년 : 1학년

학 번 : 2014310407

이 름 : 이준혁

제 출 일 : 2014년 9월 28일

1, Introduction

이 실습에서는 operator에 대해 공부한다. 이 operator들은 일반적으로 사용하는 사칙연산 같은 것도 있지만, 그렇지 않은 경우도 많다. 그렇기 때문에 이 operator들이 어떻게 사용되는지 알아야 할 필요성이 있다. operator가 복잡하게 결합되면 어떤 결과가 나오는지, 즉 어느 것을 먼저 할 지(precedence)나 어느 방향에서 먼저 할 지(associativity) 등이다.

이 실습에서는 여러 operator가 복잡하게 사용되었을 때, 어떤 연산을 할 지 예측하고, 그 결과가 옳은지 직접 프로그래밍을 통해 알아볼 것이다.

2. Problem Statement

(A) Describe what is the problem

문제 1) $a+++b$ 라는 다소 애매한 expression이 어떤 결과가 나오는 지 해석하는 것이다. 다음 세 가지로 유추해 볼 수 있겠다.

1] $(a++ + b)$

2] $(a + ++b)$

3] $(a + + + b)$

이 중 어떤 결과가 나오는 지 알아내는 것이다.

문제 2) B와 C를 0으로 두고, 네 개의 식

$$A = ++B + ++C$$

$$A = B++ + C++$$

$$A = ++B + C++$$

$$A = B-- + --C$$

를 차례대로 계산하고, 각 과정마다 A, B, C를 출력하면 된다. 이 값들을 예측하고, 프로그래밍을 통해 확인하면 된다.

문제 3) $x = (y = 2) + (z = 3)$; 이라는 실행문에서, 괄호가 둘 다 없을 경우, 첫 번째 것만 없을 경우, 두 번째 것만 없을 경우, 둘 다 있을 경우 네 가지 경우에서 오류가 발생하는지, 발생하지 않는다면 변수의 값이 어떻게 출력되는지 알아보는 것이다.

1] $x = y = 2 + z = 3$: 괄호가 둘 다 없음

2] $x = y = 2 + (z = 3)$: 첫 번째 것만 없음

3] $x = (y = 2) + z = 3$: 두 번째 것만 없음

4] $x = (y = 2) + (z = 3)$: 둘 다 있음

즉, 이 네 가지 경우를 살펴보는 것이다.

(B) Describe how you solve the problem.

문제 1) $a+++b$ 라는 식을 바로 이해하긴 쉽지 않았다. 그래서 유추한 세 개의 식을 계산했을 때 나오는 값과 $a+++b$ 를 계산했을 때 나오는 값과 비교해 보면

될 것이라 생각했다.

1) $(a++ + b)$

이 식은 수업시간에 배운 대로 ++가 postfix이므로, a와b를 먼저 더하고, a에 1을 더한다고 볼 수 있다. 즉, $a+b$ 가 expression의 값이 된다. 또, a의 값은 1 증가한다. b의 값은 변화가 없다.

2) $(a+ ++b)$

이 식 역시 수업시간에 배운 대로 ++가 prefix이므로, b에 1을 먼저 더하고, $a+b$ 를 계산한다고 볼 수 있다. 이 때 $a+(b+1)$ expression의 값이 된다.(이 때 이 b는 처음 주어진 b값이다.) a의 값은 변화가 없다.

3) $(a + + + b)$

이 식은 해석하기 조금 어려웠다. 처음에는 그냥 $a + b$ 일 것이라 생각했다. 그러나 +를 unary operator로 쓸 수도 있다고 생각하니 해석할 수 있었다. Unary operator는 right associative하므로 왼쪽에서부터 연산하면 $a+(+(+b))$ 라고 해석할 수 있겠다. 다만 가장 왼쪽의 +는 두 수 사이에 있으므로 binary operator라 해석했다. 이 때 expression의 값은 $a+b$ 일 것이고, a와 b는 변화가 없다.

이 세가지를 해석했을 때 나오는 값과 $a+++b$ 를 계산했을 때 나오는 값을 프로 그래밍을 통해 비교해 보면 풀 수 있을 것이라 생각했다.

문제 2) 주어진 네 개의 식은 prefix와 postfix의 연산 순서만 잘 안다면 쉽게 풀 수 있을 것 같았고, 실제로 그러했다.

$$1] A = ++B + ++C$$

B, C 모두 prefix 이므로 $B = B + 1$, $C = C + 1$, $A = B + C$ 로 계산될 것이라 생각했다.

$$2] A = B++ + C++$$

B, C 모두 postfix 이므로 $A = B + C$, $B = B + 1$, $C = C + 1$ 로 계산될 것이라 생각했다.

$$3] A = ++B + C++$$

B는 prefix, C는 postfix 이므로 $B = B + 1$, $A = B + C$, $C = C + 1$ 로 계산될 것이라 생각했다.

$$4] A = B-- + --C$$

C는 prefix, B는 postfix 이므로 $C = C - 1$, $A = B + C$, $B = B - 1$ 로 계산될 것이라 생각했다.

문제 3) $x = (y = 2) + (z = 3)$ 에서 괄호가 둘 다 없을 경우, 첫 번째 것만 없을 경우, 두 번째 것만 없을 경우, 둘 다 있을 경우 네 가지가 있는데, 어떤 경우는 정상 작동하고, 어떤 경우는 제대로 작동하지 않을 것 같았다. 일단 경우를 나누

어 생각해 보았다.

1] $x = y = 2 + z = 3$: 괄호가 둘 다 없음

assignment operator는 right associative 하므로, 위 식을 괄호로 묶으면

$x = (y = (2 + z = 3))$ 가 될 것이다. 그런데 assignment operator의 왼쪽에는 단일 변수만 올 수 있으므로, 구문 오류가 생길 것이라 추측할 수 있었다. 이 정보를 바탕으로 다음 문제들을 해석해 보았다.

2] $x = y = 2 + (z = 3)$: 첫 번째 것만 없음

괄호로 묶으면 $x = (y = 2 + (z = 3))$ 이다. 이 경우에는 assignment operator의 왼쪽에 단일 변수가 왔으므로, 정상적으로 변수의 값들이 출력될 것이라 추측했다.

3] $x = (y = 2) + z = 3$: 두 번째 것만 없음

괄호로 묶으면 $x = ((y = 2) + z = 3)$ 이다. 이 경우에는 assignment operator의 왼쪽에 단일 변수가 아닌 expression이 왔으므로, 오류가 생길 것이라 추측했다.

4] $x = (y = 2) + (z = 3)$: 둘 다 있음

괄호로 묶으면 $x = ((y = 2) + (z = 3))$ 이다. 이 경우에는 assignment operator의 왼쪽에 단일 변수가 왔으므로, 정상적으로 변수의 값들이 출력될 것이라 추측했다.

3. Implementation

문제 1) 변수 a, b는 자료에 있는 것과 같은 것을 사용했다. a와 b값이 어떻게 변하는 지만 보면 되므로, 초기값은 별로 중요하지 않다고 생각했다. 따라서 계산하기 쉬운 수 $a=1, b=1$ 로 뒀다.

또 이 연산들을 저장할 변수 하나가 필요했다. 그래서 '합'이라는 의미를 가져 알아보기 쉬운 변수 sum을 정의했다.

이제 $(a++b)$ 를 어떻게 해석하느냐에 따라 a, b, sum값이 어떻게 나올 지 생각해 보았다.

1] $sum = a++ + b$

$sum = a + b, a = a + 1$ 이므로, $a = 2, b = 1, sum = 2$ 가 될 것이다.

2) $sum = a+ ++b$

$b = b + 1, sum = a + b$ 이므로, $a = 1, b = 2, sum = 3$ 이 될 것이다.

3) $sum = a + + + b$

$sum = a + b$ 이므로, $a = 1, b = 1, sum = 2$ 이 될 것이다

이를 표로 나타내면 다음과 같다.

표 1. 세 개의 표현식을 실행했을 때 예상되는 a, b, sum 값(a, b=1)

표현식	실행 후 a 값	실행 후 b 값	실행 후 sum 값
a++ + b	2	1	2
a+ ++b	1	2	3
a + + + b	1	1	2

위 세 예측의 a, b, sum값과 sum = a+++b로 두었을 때 a, b, sum 값을 비교해 a+++b가 어떻게 해석되는지 알아보기 위해 다음과 같이 프로그램 코드를 입력했다.

```
sum = a+++b, printf("sum = a+++b 일 때 a=%d b=%d sum=%d이다.\n",
a,b,sum);
```

이렇게 설정한 후 앞에서 살펴본 세 결과를 비교해 보았다.

문제 2) 변수와 초기값은 별다른 의미가 없기 때문에 자료에서 주어진 것과 같은 것으로 사용했다.

이제 각 단계를 통해 A, B, C값이 단계에 걸쳐 어떻게 변할 지 생각해 보았다.

1] A = ++B + ++C, B = 0, C = 0 이므로, B = B + 1, C = C + 1, A = B + C 가 되어 B = 1, C = 1, A = 2가 될 것이다.

2] $A = B++ + C++$, $B = 1$, $C = 1$ 이므로, $A = B + C$, $B = B + 1$, $C = C + 1$ 이 되어 $A = 2$, $B = 2$, $C = 2$ 가 될 것이다. (처음에 $B = 1$, $C = 1$ 이라 둔 것은 1]의 계산 결과에서 비롯된 것이다.)

3] $A = ++B + C++$, $B = 2$, $C = 2$ 이므로 $B = B + 1$, $A = B + C$, $C = C + 1$ 이 되어 $B = 3$, $A = 5$, $C = 3$ 가 될 것이다.

4] $A = B-- + --C$, $B = 3$, $C = 3$, $C = C - 1$, $A = B + C$, $B = B - 1$ 이 되어 $C = 2$, $A = 5$, $B = 2$ 가 될 것이다

이를 표로 나타내면 다음과 같다.

표 2. 네 개의 실행문을 차례대로 실행했을 때 예상되는 A, B, C 값

표현식	A 값	B 값	C 값
$A = ++B + ++C$	2	1	1
$A = B++ + C++$	2	2	2
$A = ++B + C++$	5	3	3
$A = B-- + --C$	5	2	2

이 결과를 프로그래밍을 통해 나온 결과랑 비교해 보면 된다. 이 문제에서는 코드를 어떻게 짜는 지를 물어 보는 것이 아니라, 예상한 값을 실제

프로그래밍을 통한 결과와 비교해 보는 것이다. 따라서 코드는 자료에 있는 것에서 표현의 변화만 주었을 뿐, 사실상 자료와 똑같이 작성했다. 그러므로 여기에 코드를 적지는 않겠다. 그리고 예상 값과 프로그래밍을 통해 얻은 값을 비교해 보았다.

문제 3) 이 문제 역시 변수가 큰 의미를 가지지 않기 때문에 자료에 나온 것과 같은 x, y, z 로 설정했다. 그리고 프로그래밍 도중에 assignment operator를 이용해 변수에 값을 할당하므로, 초기값은 설정하지 않았다.

이제 네 가지 경우로 나누어 생각해 보았다.

$$1] x = y = 2 + z = 3$$

앞에서 예상한 대로, 위 식을 괄호로 묶으면 $x = (y = (2 + z = 3))$ 가 될 것이다. 우선 assignment operator의 왼쪽에는 단일 변수만 올 수 있으므로, $2+z = 3$ 이라는 것이 정의되지 않아 오류가 생길 것이다. 이 부분은 주석 처리 하기로 했다.

$$2] x = y = 2 + (z = 3)$$

괄호로 묶으면 $x = (y = 2 + (z = 3))$ 이다. 이 경우에는 우선 z 에 3을 assign했다. 그 후에 y 에 $2+z$, 즉 5를 assign 했다. 마지막으로 x 에 y 를 assign 했으므로 x 는 5가 된다. 따라서 $x = 5, y = 5, z = 3$ 이 된다.

$$3] x = (y = 2) + z = 3$$

앞에서 예상한 대로, 괄호로 묶으면 $x = (y = 2) + z = 3$ 이다. assignment operator의 왼쪽에는 단일 변수만 올 수 있으므로, $(y = 2) + z = 3$ 이라는 것이 정의되지 않아 오류가 생길 것이다. 이 부분 역시 주석 처리 할 것이다.

4] $x = (y = 2) + (z = 3)$

괄호로 묶으면 $x = ((y = 2) + (z = 3))$ 이다. 이때 y에 2를 assign, z에 3을 assign 하고 x에 $y + z$, 즉 5를 assign했다. 따라서 $x = 5, y = 2, z = 3$ 이 된다.

이를 잘 표현할 수 있게 다음과 같이 코드를 짰다.

```
x = y = 2 + z = 3;
```

```
printf("괄호가 둘 다 없을 경우 x=%d, y=%d, z=%d", x, y, z);
```

```
x = y = 2 + (z = 3);
```

```
printf("첫 번째 괄호가 없을 경우 x=%d, y=%d, z=%d", x, y, z);
```

```
x = (y = 2) + z = 3;
```

```
printf("두 번째 괄호가 없을 경우 x=%d, y=%d, z=%d", x, y, z);
```

```
x = (y = 2) + (z = 3);
```

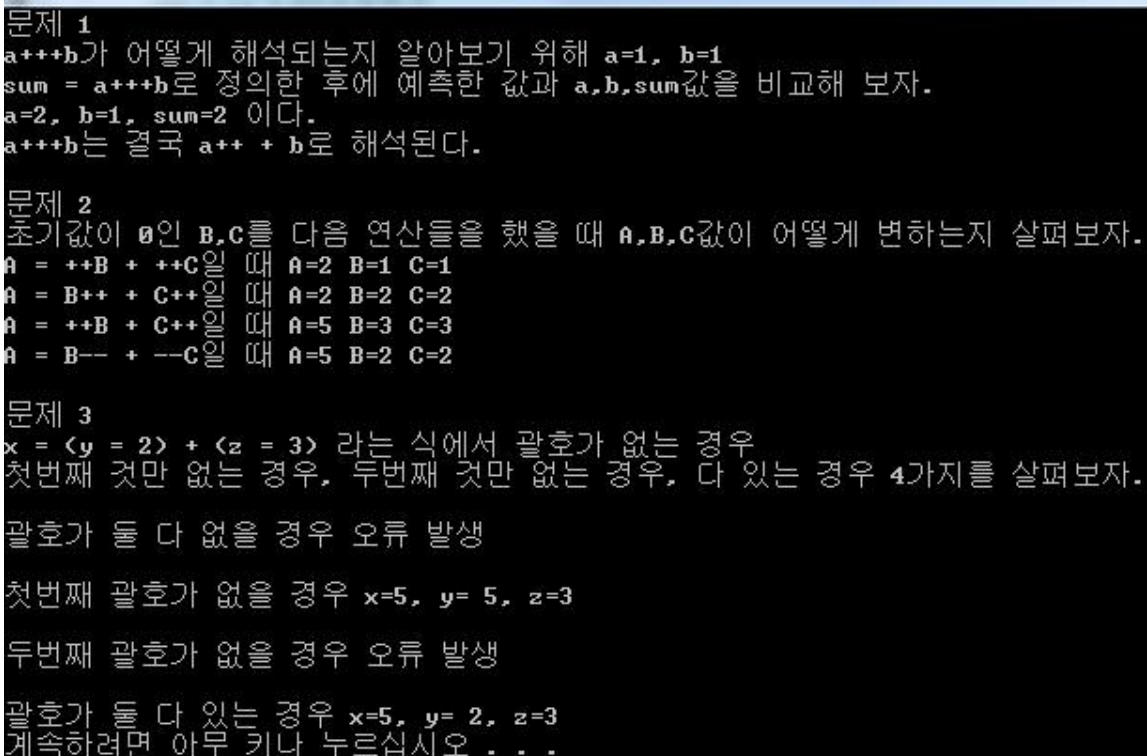
```
printf("괄호가 둘 다 있는 경우 x=%d, y=%d, z=%d", x, y, z);
```

괄호가 둘 다 없는 경우와 두 번째 괄호가 없는 경우는 오류가 날 것임을 예측했기 때문에 이 부분은 컴파일 되지 않을 경우 주석처리를 하고, 다음과 같은 문구를 각각 입력하기로 했다.

```
printf("괄호가 둘 다 없을 경우 오류 발생\n\n");
```

```
printf("두 번째 괄호가 없을 경우 오류 발생\n\n");
```

4. Result



```
문제 1
a+++b가 어떻게 해석되는지 알아보기 위해 a=1, b=1
sum = a+++b로 정의한 후에 예측한 값과 a,b,sum값을 비교해 보자.
a=2, b=1, sum=2 이다.
a+++b는 결국 a++ + b로 해석된다.

문제 2
초기값이 0인 B,C를 다음 연산들을 했을 때 A,B,C값이 어떻게 변하는지 살펴보자.
A = ++B + ++C일 때 A=2 B=1 C=1
A = B++ + C++일 때 A=2 B=2 C=2
A = ++B + C++일 때 A=5 B=3 C=3
A = B-- + --C일 때 A=5 B=2 C=2

문제 3
x = <y = 2> + <z = 3> 라는 식에서 괄호가 없는 경우
첫번째 것만 없는 경우, 두번째 것만 없는 경우, 다 있는 경우 4가지를 살펴보자.
괄호가 둘 다 없을 경우 오류 발생
첫번째 괄호가 없을 경우 x=5, y= 5, z=3
두번째 괄호가 없을 경우 오류 발생
괄호가 둘 다 있는 경우 x=5, y= 2, z=3
계속하려면 아무 키나 누르십시오 . . .
```

문제 1) 위 화면의 결과로 알 수 있듯이, (a+++b)와 (a++ + b)의 a, b, sum 값이 같다. 즉, (a+++b)는 (a++ + b)로 해석됨을 알 수 있다.

문제 2) 예측한 값과 정확히 맞아떨어졌다. Prefix는 연산보다 먼저, postfix는 연산보다 나중에 시행된다는 사실을 재확인 할 수 있다.

문제 3) 첫 번째 괄호가 없는 경우와 괄호가 둘 다 있는 경우는 예측 값과 정확히 같았다. 또 컴파일 해 보았을 때 괄호가 둘 다 없는 경우, 두 번째 괄호만 없는 경우는 '=' : left operand must be l-value 라는 경고 문구가 나왔다.

Assignment operator의 왼쪽에 단일 변수가 오지 않아 오류가 난다는 예측과 정확히 맞아떨어졌다. 이 부분은 오류가 났기 때문에, 주석 처리 후 오류 발생이라는 문구가 출력되게 했다.

5. Conclusion & Evaluation

실습 1에서는 operator간의 다양한 활용법에 대해 알아보았다. 특히 increment/decrement는 생소한 것이라 헷갈리는 부분이 많았고, assignment operator 역시 수학에서의 등식과 다르다는 점에서 약간 당황했다. 그러나 이번 실습을 통해 여러 operator들을 더 잘 다루게 된 것 같다. 첫 실습이라 조금 헤맸지만, 한번 해보니 다음에 어떻게 할 지 대충 감이 온다.

문제 1에서는 $(a+++b)$ 라는 생소한 expression을 어떻게 해석할 지를 프로그래밍을 통해 알아보았다. 처음엔 어떻게 해석할 지 감조차 오지 않았다. 프로그램을 짜고 해석하면서, $(a+++b)$ 가 $(a++ + b)$ 로 해석된다는 사실을 알 수

있었다. 이 결과에 처음에는 의문이 들었다. 그래서 왜 이렇게 되는지 곰곰이 생각해 보니 대충 이런 것 같았다. 교수님께서 컴파일러는 최대한 길게 해석한다고 하셨는데, $a+++b$ 를 왼쪽에서부터 해석해보면, a 는 변수로 해석하고, 그 다음에 오는 $+$ 를 하나의 operator로 해석할 수도 있겠지만, 최대한 길게 해석해야 하니 뒤에 있는 $+$ 까지 고려해 $++$ 로 해석한다. 더 길게 해석할 수는 없으니(즉, $+++$ 는 없으니) $++$ 에서 끊고, 그 다음에 operator $+$, 마지막으로 변수 b 를 해석하면 $(a++ + b)$ 라는 결과가 나올 것이다. 이렇게 생각했으나, 틀린 부분이 있을 수도 있으니 질문 해 볼 수도 있을 것 같다.

문제 2에서는 increment/decrement와 다른 연산(여기서는 binary operator $+$ 가 되겠다)이 섞여 있을 때 어떻게 해석되는지 물어보는 것이었다. 수업 시간에 배운 내용이었지만, 완전히 이해하지는 못한 부분이었다. 퀴즈에서도 이런 류의 문제를 하나 틀렸던 것 같다. 그러나 이 문제를 통해서 increment/decrement를 어떻게 해석할 지 완전히 알아냈다.

문제 3에서는 assignment operator에 대해 물어봤다. 이 부분도 완벽히 이해한 부분은 아니었다. 수업시간에 $a = (b = 0)$ 라는 식을 해석할 때 $b = 0$ 이라는 식의 값은 왼쪽 변수 b 라는 말을 잘 이해하지 못했다. 그러나 이 문제를 풀면서 식의 값이 왼쪽에 오는 변수라는 사실을 이해할 수 있었다. 또 여기에서 깨달은 것이 assignment operator의 왼쪽에는 값을 할당 받는 변수가 와야 하므로, 변수가 하나만 와야 한다는 것을 정확하게 알게 되었다. 결국 이러한 발견이 괄호가 둘 다 없을 때와 두 번째 것만 있을 때 왼쪽에 변수 하나가 오지 않아서 오류가 생긴다

는 점도 생각하게 해 준 것이다. 이 문제 덕에 assignment operator를 자세히 알 수 있었고, 어떻게 하면 오류가 나는지도 알게 되었다.

6. 참고 문헌

[1] 민형복, 보고서 예시 p1.pdf.

[2] 프로그래밍 기초와 실습 사이트, <http://class.icc.skku.ac.kr/~min/C/> , 2014 년 9 월.