



C++ Study



C++ Study

연산자와 제어문, 함수

울산대학교 | 박예찬 | Github:devpyc | yechan6855@naver.com



C++ Study



산술 연산자



산술 연산자

연산자	설명
+	더하기
-	빼기
*	곱하기
/	나누기
%	나머지 (정수만 가능)



7/2는 왜 3일까?



나눗셈의 함정

피연산자가 둘다 정수형이라면, 결과도 정수로 출력된다

```
cout<<7/2<<"\n";
```



나눗셈의 함정

피연산자가 하나라도 실수형 이라면?
결과는 실수 나눗셈이 된다.

```
cout<<7/2.0<<"\n";
```



타입 변환을 이용한 해결법

정수형인 피연산자로 실수 나눗셈을 하고싶다면

```
int a=7;
```

```
int b=2;
```

```
cout<<(double)a/b<<"\n";
```



자주 하는 실수

```
int a=7;  
int b=2;  
double avg=sum/cnt; // avg=3.0  
double avg=(double)sum/cnt; // avg=3.5
```



C++ Study



비교 연산자



비교 연산자

연산자	의미	($a = 3, b = 5$)
$==$	같다	$a == 3 \rightarrow \text{true}$
$!=$	다르다	$a != b \rightarrow \text{true}$
$<$	보다 작다	$a < b \rightarrow \text{true}$
$>$	보다 크다	$a > b \rightarrow \text{false}$
\leq	작거나 같다	$a \leq 3 \rightarrow \text{true}$
\geq	크거나 같다	$b \geq 3 \rightarrow \text{true}$



조건문에서의 비교 연산자

```
int a=10;  
if(a==10){  
    cout<<"a is 10\n";
```

```
}
```

```
if(a!=5){  
    cout<<"a is not 5\n";  
}
```



최악의 실수 (1)

`x=5` 는 x에 5를 대입한다는 의미임
해당 코드의 결과는 x는 0이 아니므로 항상 true가 됨

```
int x=3;
```

```
if(x=5){  
    cout<<"Yes\n";  
}
```



최악의 실수 (2)

파이썬에선 가능하지만 C++에선 불가능하니 주의!

```
int score=95;
```

```
if(90<=score<=100) {}
```

```
if(90<=score&&score<=100){}
```



C++ Study



논리 연산자



논리 연산자

연산자	이름	의미	예시
&&	논리 AND	둘 다 참이면 참	$a > 0 \&\& a < 10$
	논리 OR	둘 중 하나라도 참이면 참	$\neg a < 0$
!	논리 NOT	참 \leftrightarrow 거짓 반전	$\neg a$



나이와 성별 구별하기

```
int age;  
char gender; // M 또는 F  
cin>>age>>gender;  
if(age>=20&&gender=='M') cout<<"성인 남자";  
if(age<0||age>150) cout<<"잘못된 나이"
```



논리 NOT

isLogin이 **false**일때 **true** 반환

```
bool isLogin=false;
```

```
if(!isLogin) cout<<"로그인이 필요합니다";
```



Short-Circuit

&& : 왼쪽이 false라면, 전체가 **false**

|| : 왼쪽이 true라면, 전체가 **true**

```
int x=0;
```

```
int y=10;
```

```
if(x!=0 && y/x>1) {}
```

x!=0이 **false**이므로 **y/x>1**은 계산 안함



C++ Study



비트 연산자



비트 연산자

연산자	이름	의미	예시 ($a = 6(110)$,	결과
&	AND	둘 다 1이면 1	$a \& b$	2 (010)
	OR	하나라도 1이면 1	`a	b`
^	XOR	다르면 1	$a ^ b$	5 (101)
~	NOT	비트 반전	$\sim a$	001 → 음수 주의
<<	Left Shift	비트를 왼쪽으로	$a << 1$	12
>>	Right Shift	비트를 오른쪽으	$a >> 1$	3



비트 연산자

```
int a = 6; // 110
```

```
int b = 3; // 011
```

```
cout << (a & b) << '\n'; // 2
```

```
cout << (a | b) << '\n'; // 7
```

```
cout << (a ^ b) << '\n'; // 5
```

```
cout << (a << 1) << '\n'; // 12
```

```
cout << (a >> 1) << '\n'; // 3
```



시프트 연산자

일반적인 곱셈연산보다는 시프트연산이 훨씬 빠르다!

$a << n$ 은 a를 2^n 과 곱한것과 같음

$a >> n$ 은 a를 2^n 으로 나눈것과 같음



C++ Study



조건문



조건문

조건식의 결과가 **true**일때만 실행

```
int age=20;  
if(age>=20) cout<<"성인입니다\n";
```



조건에 따른 출력

중첩도 가능하지만, 너무 중첩되면 가독성이 떨어짐

```
if(age>=20){  
    if(gender=='M'){  
        cout<<"성인 남자";  
    }  
}
```



여러 조건 중 하나 선택

첫 조건이 참이면 아래는 검사 X

```
int score=92;  
if(score>=90) cout<<'A';  
else if(score>=80) cout<<'B';  
else cout<<'C';
```



C++ Study



반복문



반복 횟수가 명확한 상황

for문을 사용하여 1~5까지 출력하기

```
for(int i=1; i<=5; i++){  
    cout<<i<<" ";  
}
```



조건을 기반으로 반복하는 상황

while문을 사용하여 cnt가 0일때 반복 종료

```
int cnt=5;  
while(cnt>0){  
    cout<<"Hello\n";  
    cnt--;  
}
```



EOF 기반 입력 처리

입력이 끝날때까지 출력하기

```
int n;  
while(cin>>n){  
    cout<<n<<"\n";  
}
```



C++ Study



흐름 제어



break

조건을 만족하면 즉시 반복 종료

```
for(int i=1; i<=10; i++){
    if(i==5) break;
}
```



continue

조건을 만족하면 다음으로 건너뛰기

```
for(int i=1; i<=10; i++){
    if(i%2==0) continue;
    cout<<i<<" ";
}
```



C++ Study



함수 기초



함수의 정의

특정 작업을 수행하기 위해 설계된 **재사용 가능한** 코드의 집합



예시

두 정수를 인자로 받아 합을 출력하는 함수

```
int add(int a, int b){  
    return a+b;  
}
```



함수의 구성 요소

요소	설명	예시
반환형	return할 자료형	void, int, bool, string
함수명	알아보기 쉬운 이름	fibonacci, isValid, check
매개변수	입력값	int a, int b
함수식	실행될 코드	return a+b



피보나치 함수 구현

이전 두 수의 합이 다음 수가 되는 수열

```
int fibo(int n){  
    if(n<=1) return n;  
    return fibo(n-1) + fibo(n-2);  
}
```



return 0;

울산대학교 | 박예찬 | Github:devpyc | yechan6855@naver.com