세종사이버대학교

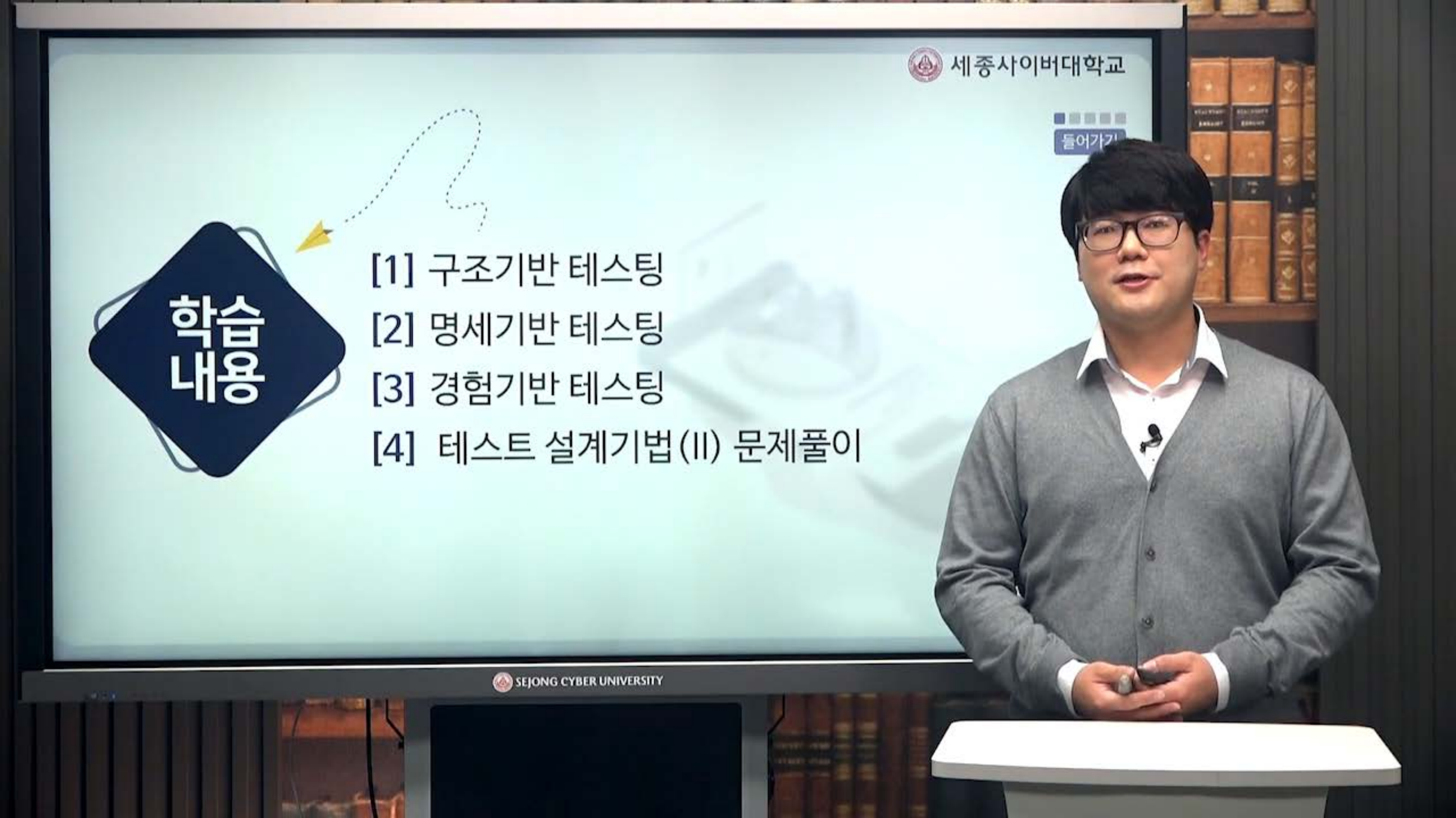
소프트웨어 테스트 전문가(CSTS) 특강

03

테스트 설계기법(II)

소프트웨어공학과
윤동환 교수

SEJONG CYBER UNIVERSITY

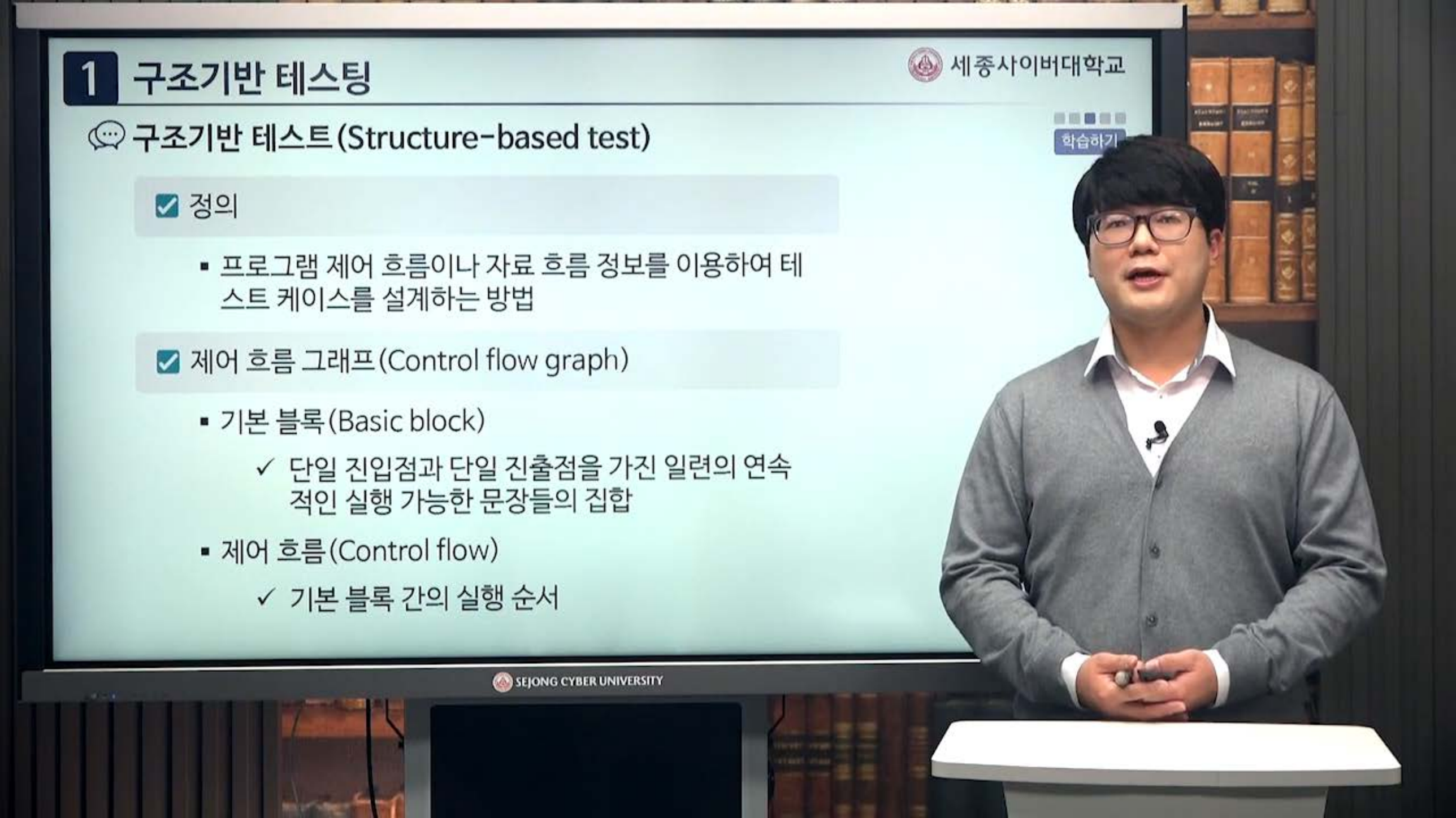

세종사이버대학교
학습 내용
[1] 구조기반 테스팅
[2] 명세기반 테스팅
[3] 경험기반 테스팅
[4] 테스트 설계기법(II) 문제풀이
SEJONG CYBER UNIVERSITY

세종사이버대학교

01

구조기반 테스팅

SEJONG CYBER UNIVERSITY


1 구조기반 테스팅
세종사이버대학교
구조기반 테스트(Structure-based test)
학습하기
정의
프로그램 제어 흐름이나 자료 흐름 정보를 이용하여 테스트 케이스를 설계하는 방법
제어 흐름 그래프(Control flow graph)
기본 블록(Basic block)
단일 진입점과 단일 진출점을 가진 일련의 연속적인 실행 가능한 문장들의 집합
제어 흐름(Control flow)
기본 블록 간의 실행 순서
SEJONG CYBER UNIVERSITY

1 구조기반 테스팅

세종사이버대학교

제어 흐름 그래프 (예제)

```
void findVinArray(int a[], int n, int v) {
   1:   int i = 0;
   2:   int count = 0;
   3:   while (i < n) {
   4:      if (a[i] == v)
   5:         count++;
   6:    i++;
         }
   7:   printf("number %d", count);
}
```

SEJONG CYBER UNIVERSITY

1 구조기반 테스팅

세종사이버대학교

제어 흐름 그래프 (예제)

학습하기

```
void findVinArray(int a[], int n, int v) {
  1:  int i = 0;
  2:  int count = 0;
  3:  while (i < n) {
  4:    if (a[i] == v)
  5:      count++;
  6:  i++;
      }
  7:  printf("number %d", count);
}
```

기본 블록

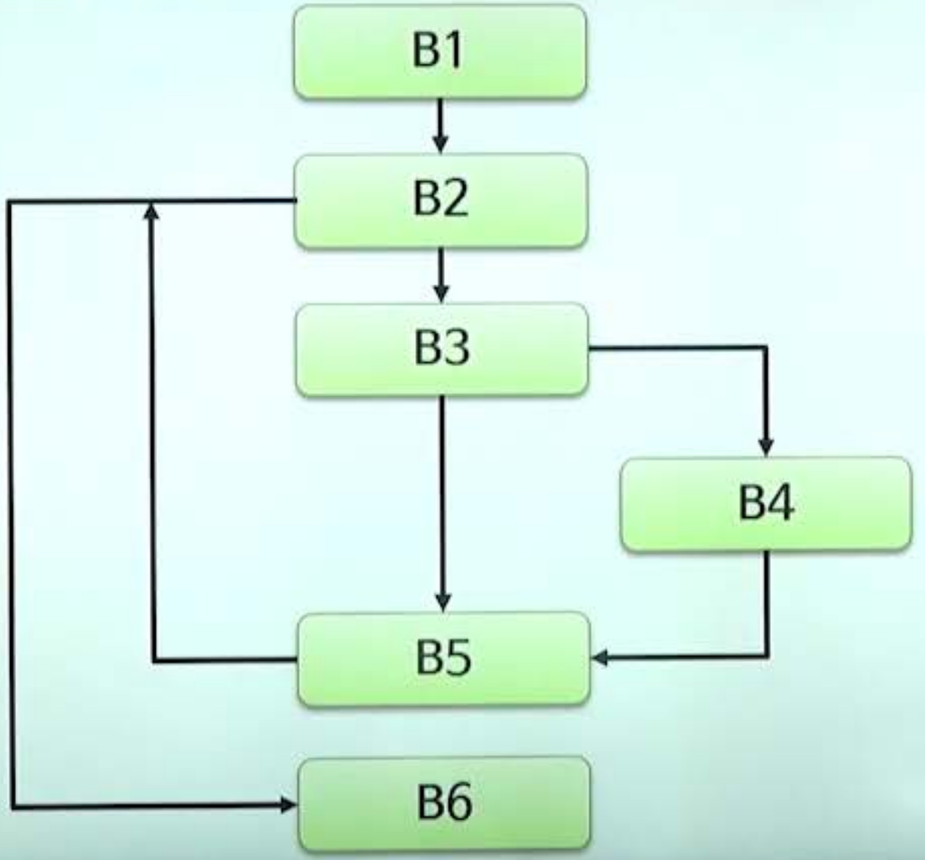
기본 블록	문장	진입점	진출점
B1	{1, 2}	1	2
B2	{3}	3	3
B3	{4}	4	4
B4	{5}	5	5
B5	{6}	6	6
B6	{7}	7	7

1 구조기반 테스팅

세종사이버대학교

학습하기

제어 흐름 그래프 (예제)



SEJONG CYBER UNIVERSITY

1 구조기반 테스팅

세종사이버대학교

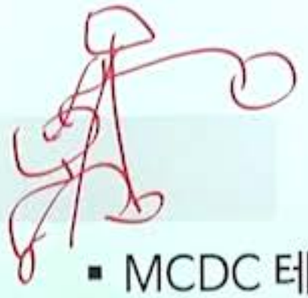
구조기반 테스트(Structure-based test)

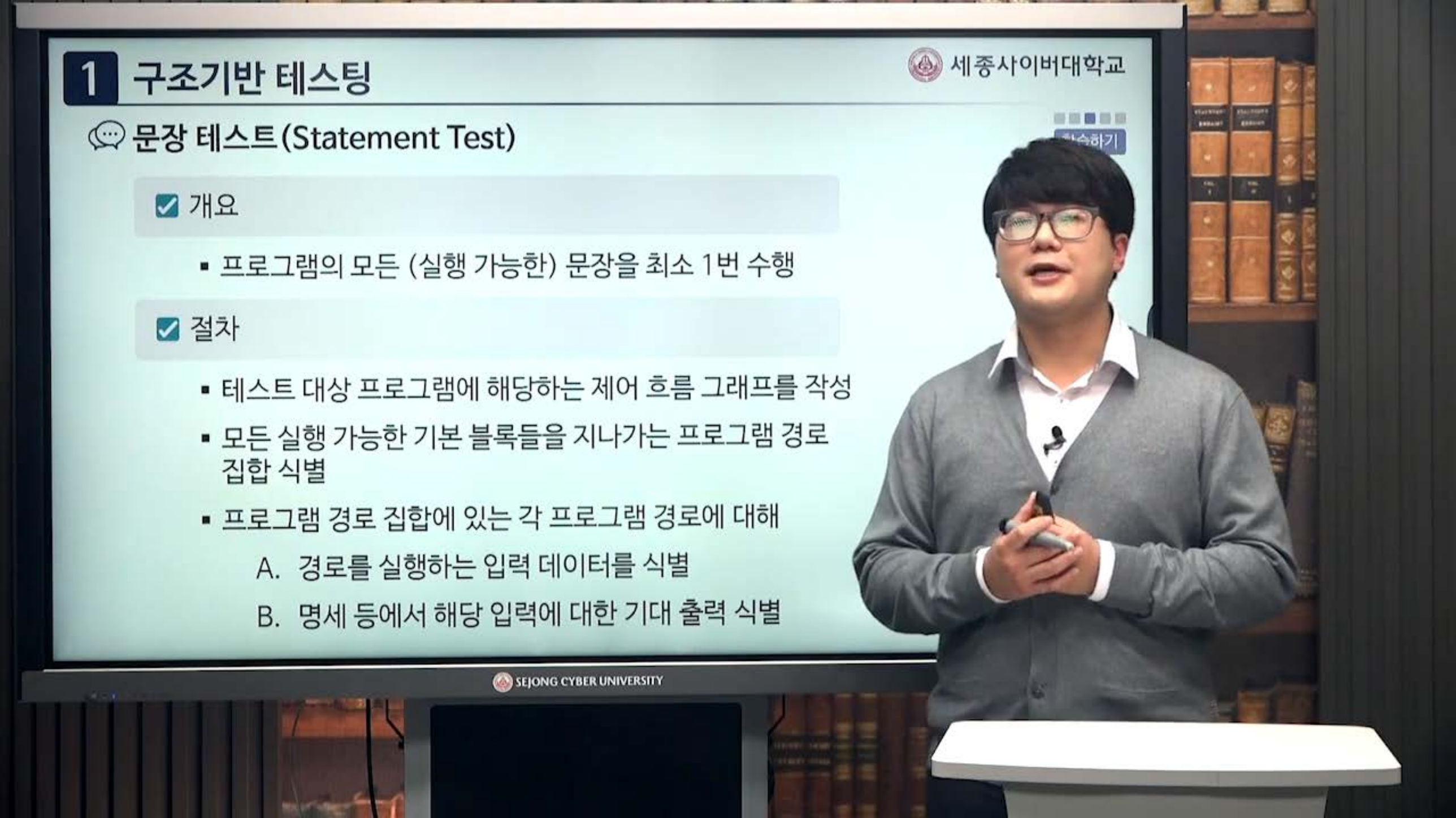
특징

- 가장 이상적인 구조 기반 테스트는 프로그램의 모든 경로를 최소 1번 실행하여 테스트 하는 것
- But, 현실적으로 불가능 → 2^n

종류

- 문장 테스트(Statement test)
- 분기/결정 테스트(Branch/Decision test)
- 조건 테스트(Condition test)
- 다중 조건 테스트(Multiple Condition test)
- MCDC 테스트
- 기본 경로 테스트




1 구조기반 테스팅
세종사이버대학교
문장 테스트(Statement Test)
개요
프로그램의 모든 (실행 가능한) 문장을 최소 1번 수행
절차
테스트 대상 프로그램에 해당하는 제어 흐름 그래프를 작성
모든 실행 가능한 기본 블록들을 지나가는 프로그램 경로 집합 식별
프로그램 경로 집합에 있는 각 프로그램 경로에 대해
A. 경로를 실행하는 입력 데이터를 식별
B. 명세 등에서 해당 입력에 대한 기대 출력 식별
SEJONG CYBER UNIVERSITY

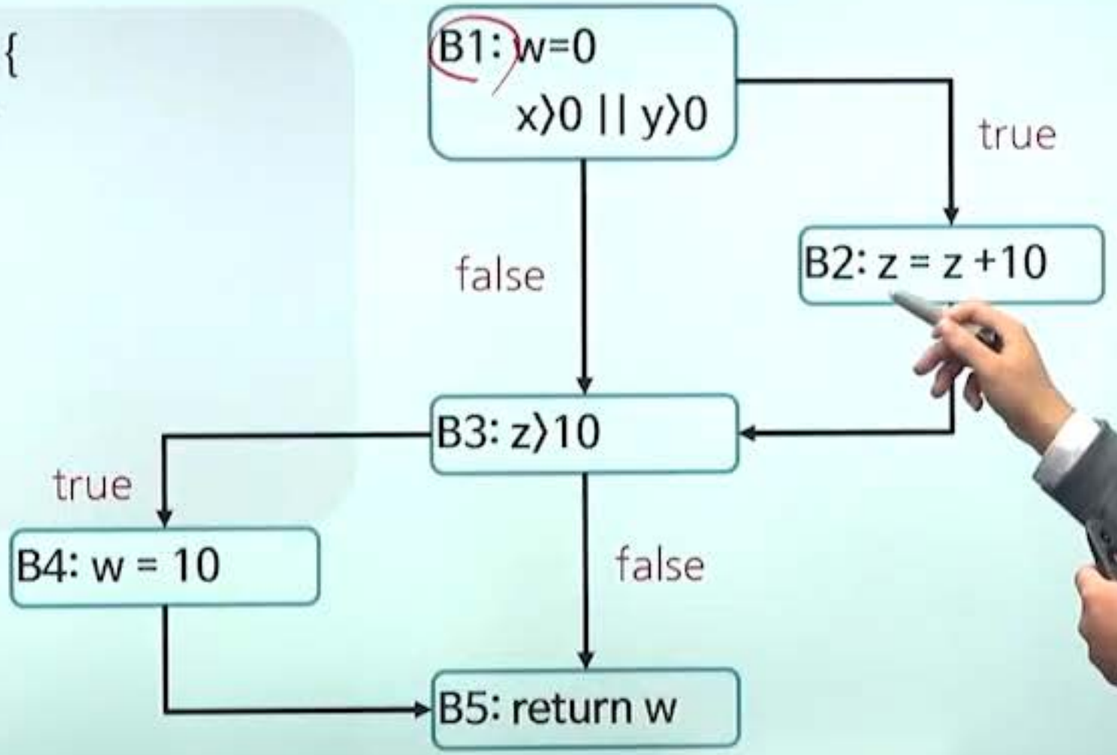
1 구조기반 테스팅

세종사이버대학교

문장 테스트(Statement Test) 예시

학습하기

```
int foo(int x, int y, int z) {
    int w = 0;
    if (x > 0 || y > 0)
        z = z + 10;
    if (z > 10)
        w = 10;
    return w;
}
```

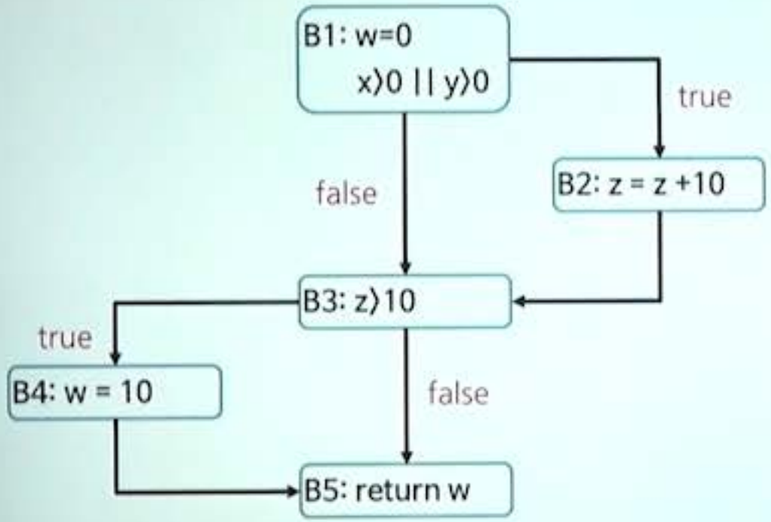


1 구조기반 테스팅

세종사이버대학교

학습하기

문장 테스트(Statement Test) 예시



프로그램 경로 집합

TS1 = {<B1, B2, B3, B5>, <B1, B3, B4, B5>}

TS2 = {<B1, B2, B3, B4, B5>}

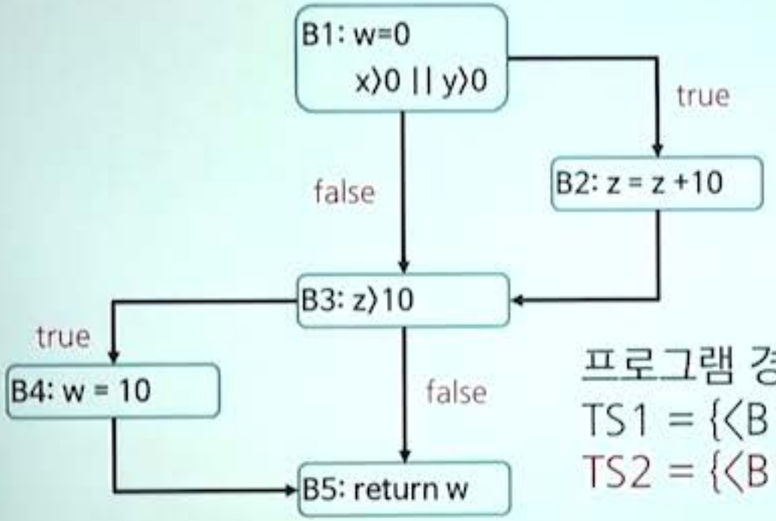
SEJONG CYBER UNIVERSITY

1 구조기반 테스팅

세종사이버대학교

문장 테스트(Statement Test) 예시

학습하기



프로그램 경로 집합

TS1 = {〈B1, B2, B3, B5〉, 〈B1, B3, B4, B5〉}

TS2 = {〈B1, B2, B3, B4, B5〉}

테스트 케이스	입력			기대출력	실행된 블록
	x	y	z		
1	10	10	10	10	B1, B2, B3, B4, B5

SEJONG CYBER UNIVERSITY

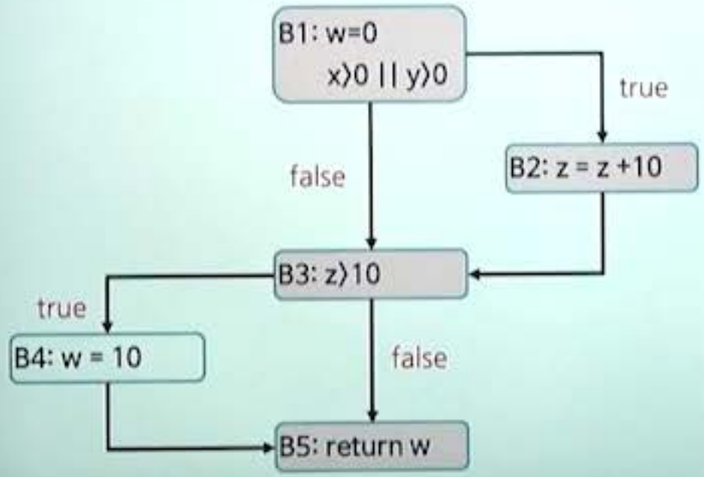
세종사이버대학교

문장 테스트(Statement Test)

학습하기

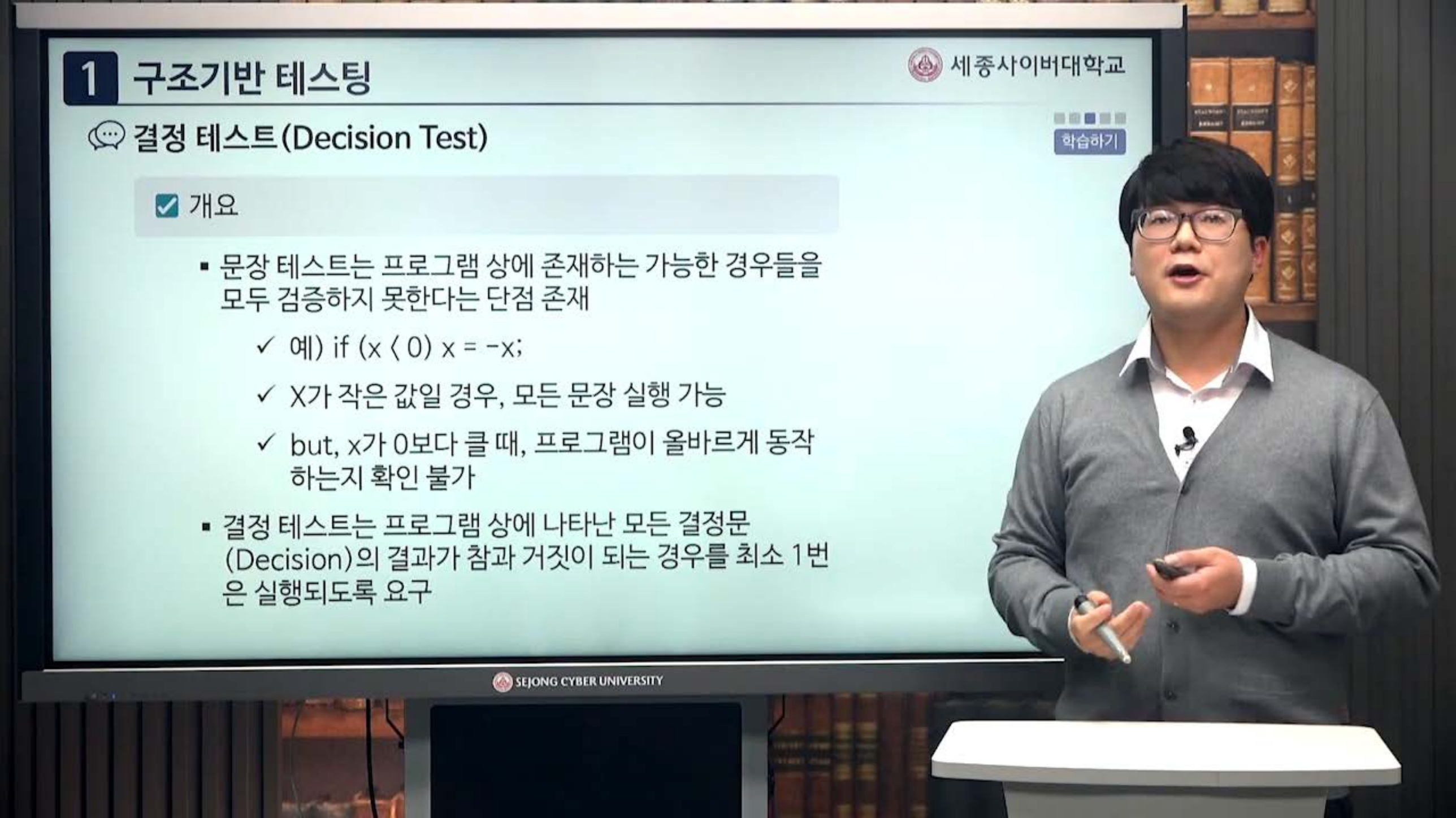
☑ 문장 커버리지(Statement Coverage)

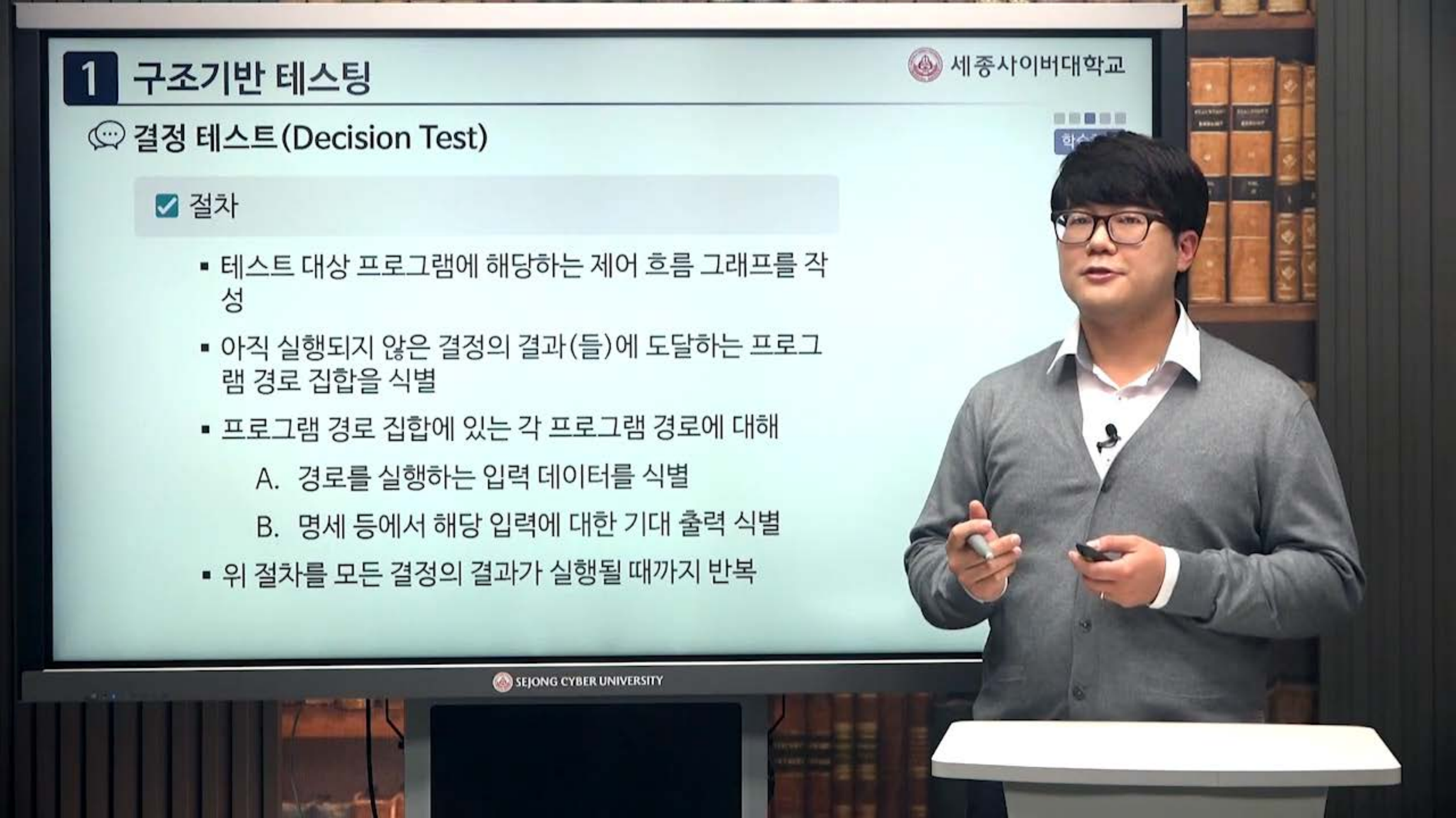
- 문장 커버리지(%) = $\frac{\text{테스트 케이스 집합에 의해 실행된 문장의 수}}{\text{전체 실행 가능한 프로그램 문장의 수}} \times 100$



입력: x = 10, y = 10, z = 0

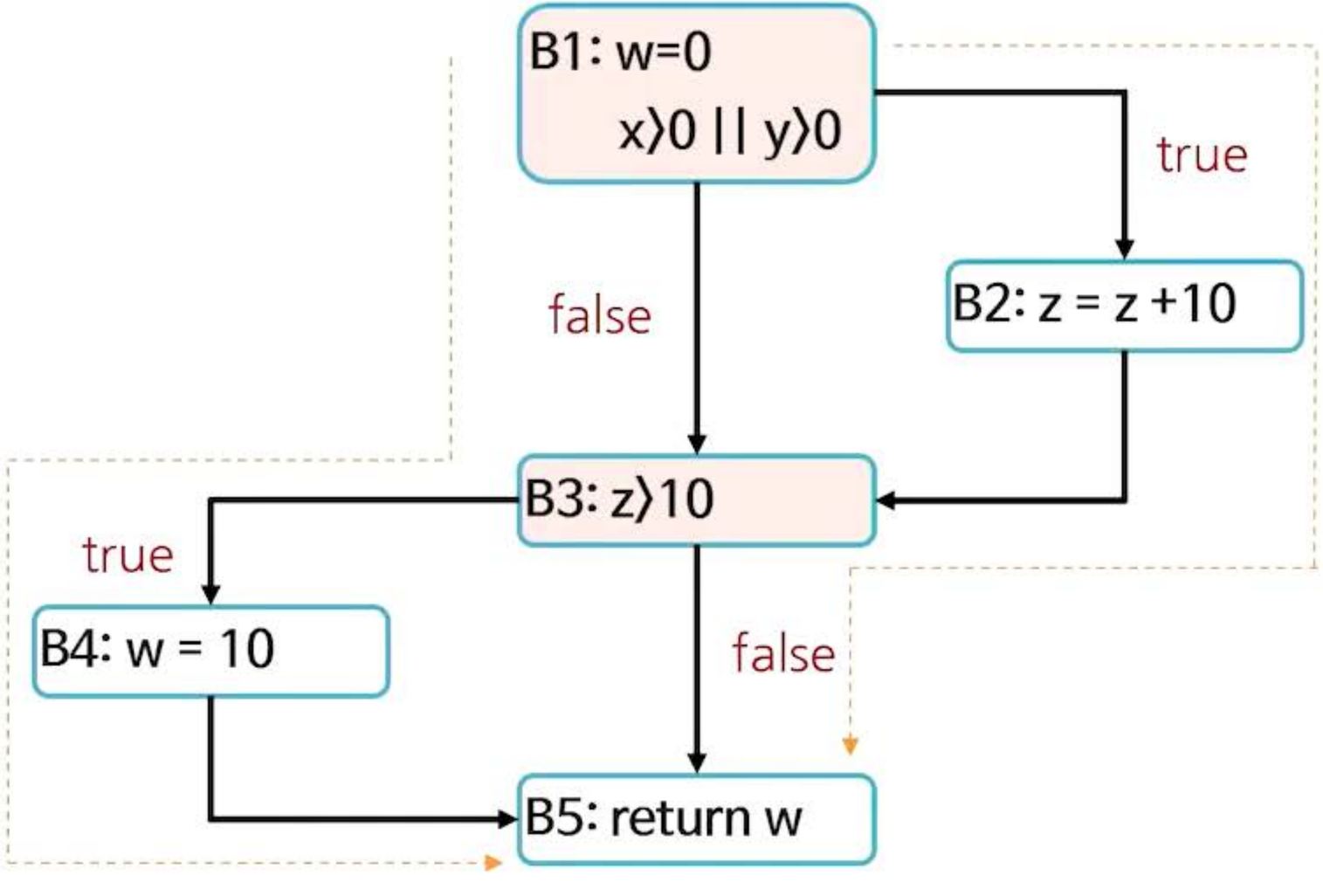
문장 커버리지
= 6문장 / 7문장 x 100 = 86%


1 구조기반 테스팅
세종사이버대학교
결정 테스트(Decision Test)
학습하기
개요
문장 테스트는 프로그램 상에 존재하는 가능한 경우들을 모두 검증하지 못한다는 단점 존재
예) if (x < 0) x = -x;
X가 작은 값일 경우, 모든 문장 실행 가능
but, x가 0보다 클 때, 프로그램이 올바르게 동작하는지 확인 불가
결정 테스트는 프로그램 상에 나타난 모든 결정문(Decision)의 결과가 참과 거짓이 되는 경우를 최소 1번은 실행되도록 요구
SEJONG CYBER UNIVERSITY


1 구조기반 테스팅
세종사이버대학교
결정 테스트(Decision Test)
절차
테스트 대상 프로그램에 해당하는 제어 흐름 그래프를 작성
아직 실행되지 않은 결정의 결과(들)에 도달하는 프로그램 경로 집합을 식별
프로그램 경로 집합에 있는 각 프로그램 경로에 대해
A. 경로를 실행하는 입력 데이터를 식별
B. 명세 등에서 해당 입력에 대한 기대 출력 식별
위 절차를 모든 결정의 결과가 실행될 때까지 반복
SEJONG CYBER UNIVERSITY

결정 테스트(Decision Test) 예시

학습하기

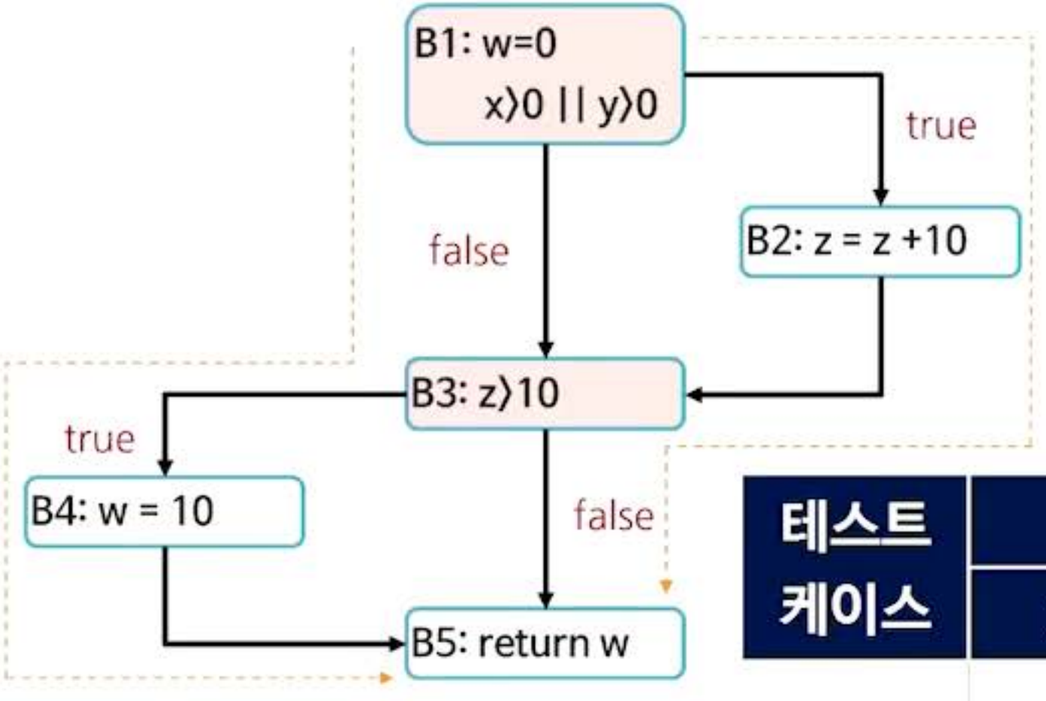


2개의 결정문 식별

2개의 경로집합 식별
TS =
{ <B1, B2, B3, B5>,
<B1, B3, B4, B5>}

결정 테스트(Decision Test) 예시

학습하기



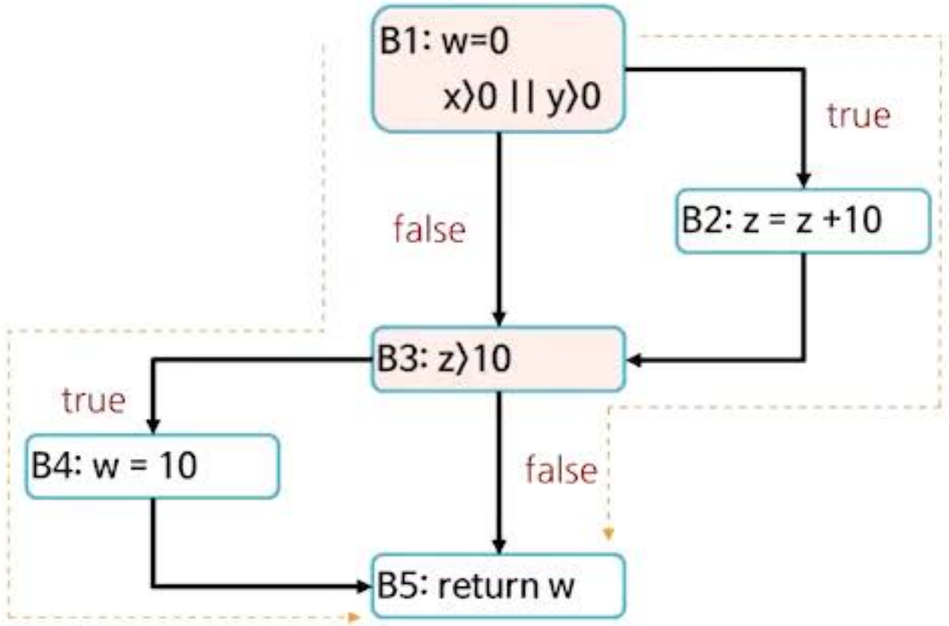
테스트 케이스	입력			기대 출력	결정 결과
	x	y	z		
1	10	10	-10	0	x>0||y>0 : true z>10 : false
2	-10	-10	20	10	x>0 || y>0 : false z > 10 : true

결정 테스트(Decision Test)

학습하기

☑ 결정 커버리지(Decision Coverage)

- 결정 커버리지(%) $= \frac{\text{테스트 케이스 집합에 의해 실행된 결정문의 수}}{\text{전체 실행 가능한 프로그램 결정문의 수}} \times 100$



표의 TC1 실행결과

결정 커버리지
= 2개 / 4개 x 100 = 50%

1 구조기반 테스팅

세종사이버대학교

조건 테스트(Condition Test)

학습하기

개요

- 조건은 논리 연산자 “and”나 “or”를 포함하지 않은 Boolean 식
- 결정은 이러한 조건들이 논리 연산자를 사용하여 구성된 Boolean 식을 의미
- 예시

```
if (x>0 && y<= -3) {
        x = y + 4;
        y = y  -  1;
}
```

SEJONG CYBER UNIVERSITY

1 구조기반 테스팅

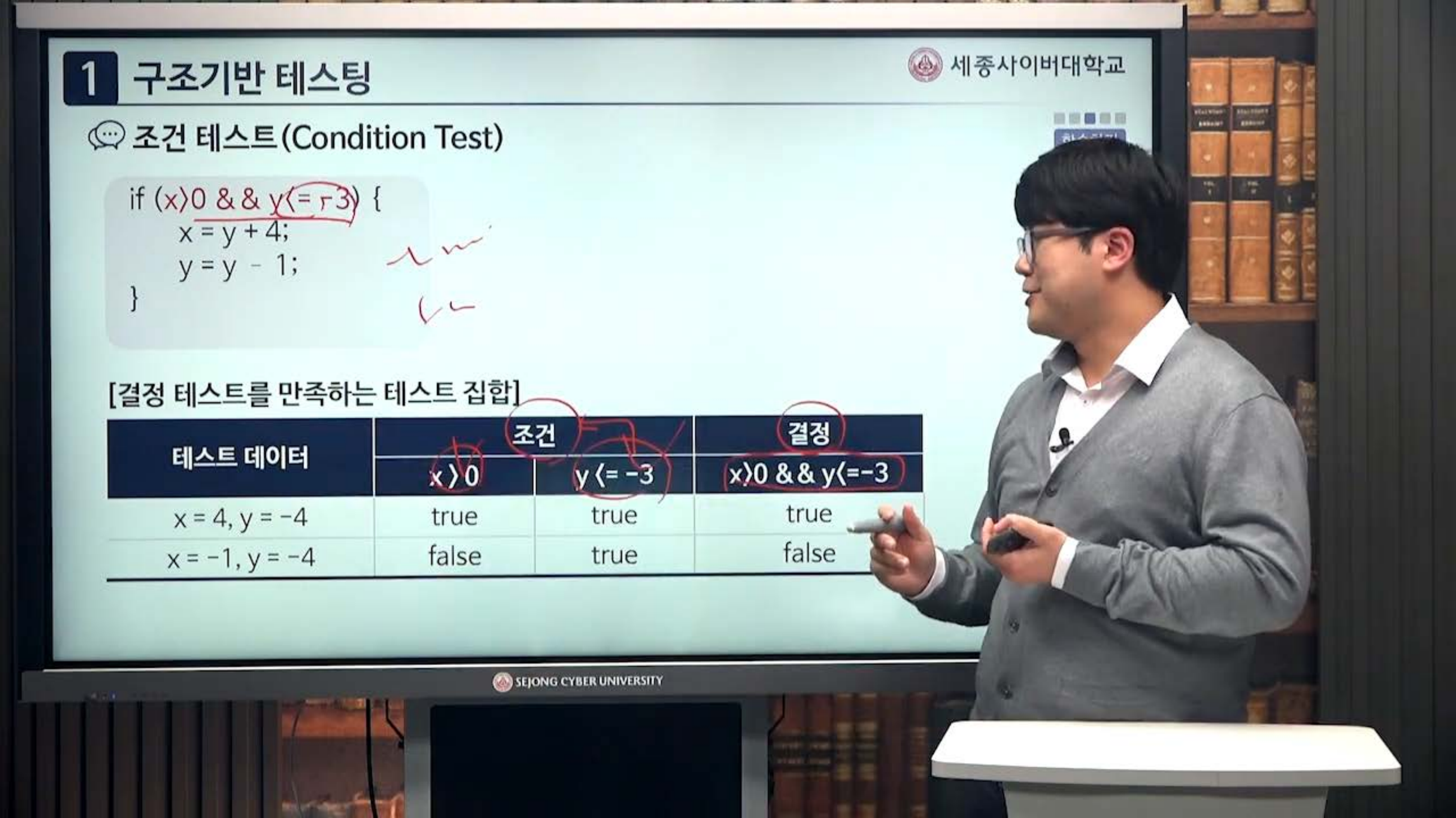
세종사이버대학교

조건 테스트(Condition Test)

```
if (x>0 && y<=-3) {
    x = y + 4;
    y = y - 1;
}
```

[결정 테스트를 만족하는 테스트 집합]

테스트 데이터	조건		결정
	x > 0	y <= -3	x>0 && y<=-3
x = 4, y = -4	true	true	true
x = -1, y = -4	false	true	false



1 구조기반 테스팅

세종사이버대학교

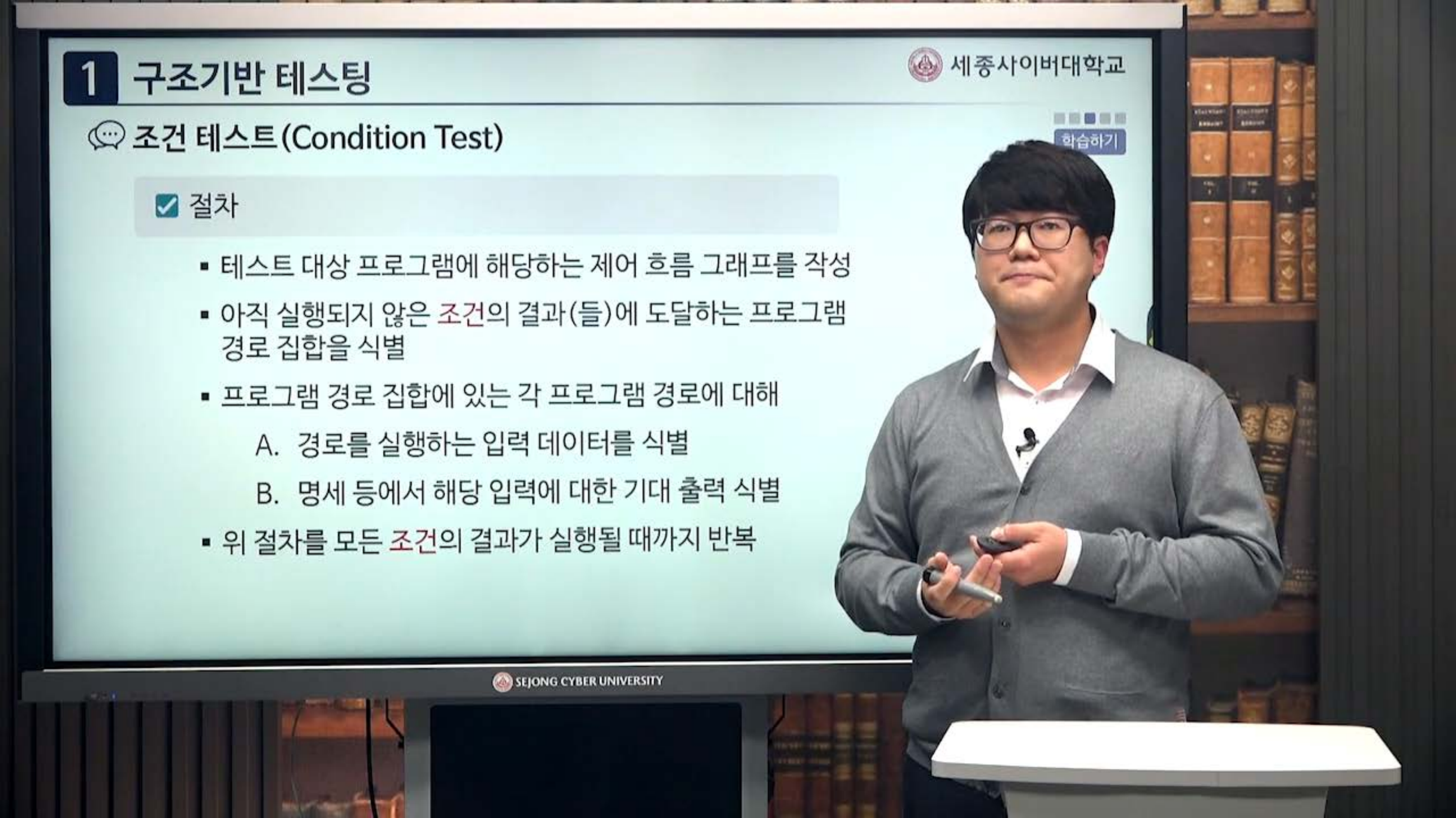
조건 테스트(Condition Test)

학습하기

```
if (x>0 && y<= -3) {
    x = y + 4;
    y = y - 1;
}
```

[조건 테스트를 만족하는 테스트 집합]

테스트 데이터	조건		결정
	x > 0	y <= -3	x>0 && y<=-3
x = 4, y = -4	true	true	true
x = -1, y = -4	false	true	false
x = -1, y = -2	false	false	false


1 구조기반 테스팅
세종사이버대학교
조건 테스트(Condition Test)
학습하기
절차
테스트 대상 프로그램에 해당하는 제어 흐름 그래프를 작성
아직 실행되지 않은 조건의 결과(들)에 도달하는 프로그램 경로 집합을 식별
프로그램 경로 집합에 있는 각 프로그램 경로에 대해
A. 경로를 실행하는 입력 데이터를 식별
B. 명세 등에서 해당 입력에 대한 기대 출력 식별
위 절차를 모든 조건의 결과가 실행될 때까지 반복
SEJONG CYBER UNIVERSITY

조건 테스트(Decision Test)

학습하기

☑ 조건 커버리지(Condition Coverage)

- 조건 커버리지(%) = $\frac{\text{테스트 케이스 집합에 의해 실행된 개별 조건의 결과수}}{\text{전체 프로그램 개별 조건의 결과 수}} \times 100$

테스트 데이터	조건	
	x 〉 0	y 〈= -3
x = 4, y = -4	true	true
x = -1, y = -4	false	true
x = -1, y = -2	false	false

표의 테스트 데이터 1행 실행결과

조건 커버리지
= 2개 / 4개 x 100 = 50%