

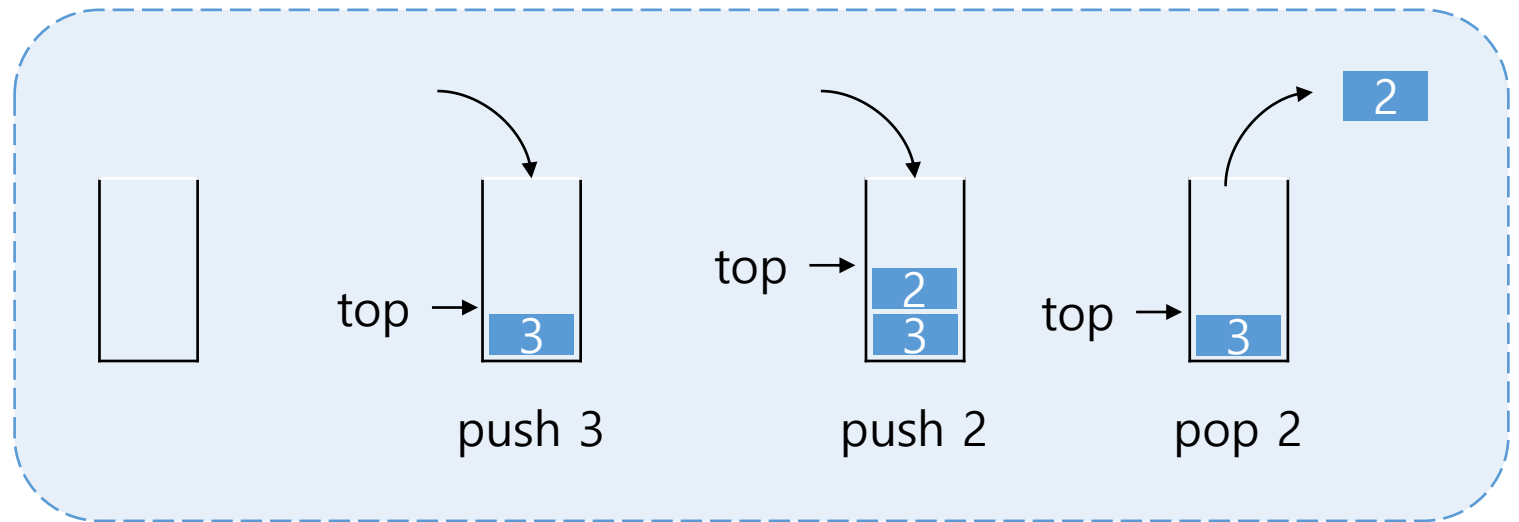
# 윈도우프로그래밍 Lab 03

스택을 활용한 괄호 검사 프로그램

김한슬  
uo3359@sookmyung.ac.kr

# 스택을 활용한 괄호 검사 프로그램

- **스택(stack)** 자료구조를 이용 (LIFO: Last-in First-out)



- push: 스택에 데이터를 저장
- pop: 스택에서 데이터를 꺼내는 작업
- top: 마지막 데이터 다음 빈칸을 가리킴(교재 P.14 참고)

# 스택을 활용한 괄호 검사 프로그램

## 표준 입력 함수 - cin.get()

- cin.get(char c)
  - 입력 스트림상의 한 글자를 읽어 c에 저장
  - Whitespace도 읽을 수 있음
- cin.get(char s[], int len)
  - 하나의 행을 읽어 배열 s에 (배열의 첫 번째 주소값에) 문자열 형식으로 저장
  - len : 입력될 데이터의 최대 크기
  - 'wn'은 읽지 않음

# 스택을 활용한 괄호 검사 프로그램

## 표준 입력 함수 - cin.get()

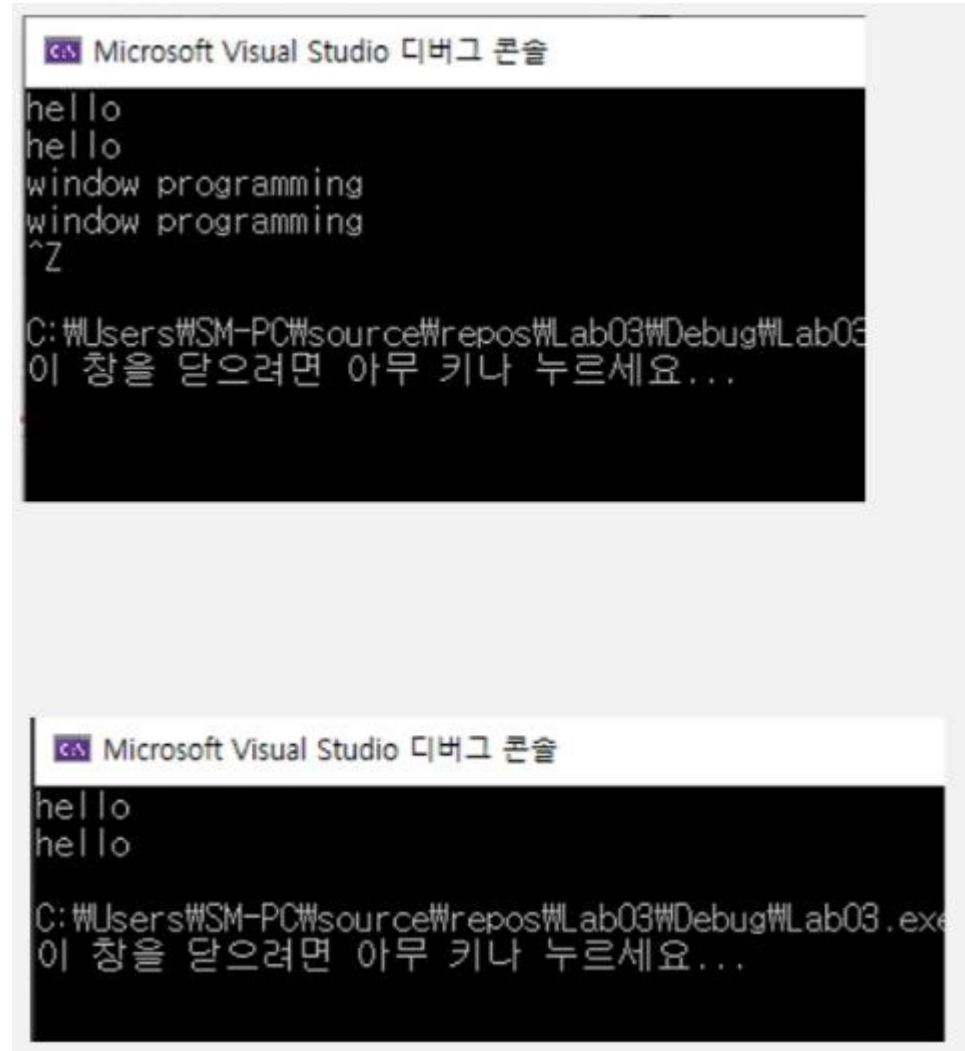
```
#include <iostream>

using namespace std;

#define LINE_SIZE 1000

int main()
{
    char line[LINE_SIZE];

    while(cin.get(line, LINE_SIZE))
    {
        cout << line << endl;
        char c;
        cin.get(c); // '₩n'을 읽고 지나감
    }
}
```



Microsoft Visual Studio 디버그 콘솔

```
hello
hello
window programming
window programming
^Z

C:\₩Users₩SM-PC₩source₩repos₩Lab03₩Debug₩Lab03.exe
이 창을 닫으려면 아무 키나 누르세요...
```

Microsoft Visual Studio 디버그 콘솔

```
hello
hello

C:\₩Users₩SM-PC₩source₩repos₩Lab03₩Debug₩Lab03.exe
이 창을 닫으려면 아무 키나 누르세요...
```

# 스택을 활용한 괄호 검사 프로그램

## 과제 : **balanced.cpp** 작성

<실행결과 예시>

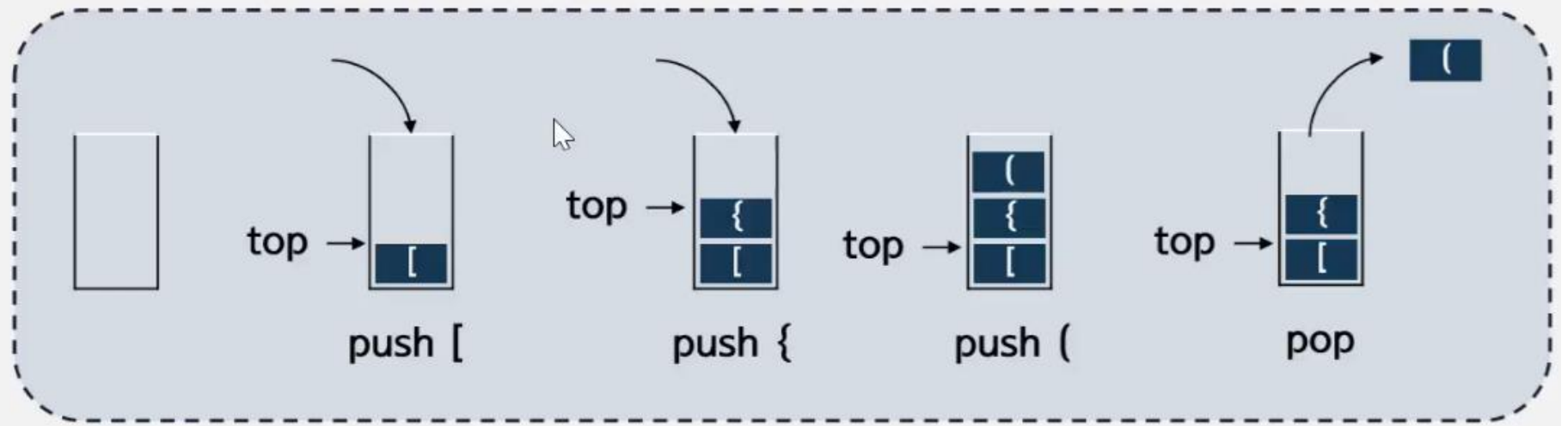
```
if (isopeningparen(c))  
if (isopeningparen(c)) : BALANCED!!  
void check_balance(char expr[])  
void check_balance(char expr[]) : BALANCED!!  
else if (isclosingparen(c) && !matching(pop(s), c)) {  
else if (isclosingparen(c) && !matching(pop(s), c)) { : MISMATCHING!!  
([({})])()  
([({})])() : BALANCED!!  
([({})])()  
([({})])() : MISMATCHING!!  
([({})])]  
([({})])]] : MISMATCHING!!  
^Z  
계속하려면 아무 키나 누르십시오 ...
```

# 스택을 활용한 괄호 검사 프로그램

## 과제 : `balanced.cpp` 작성

- 괄호들이 올바르게 짝지어지는지 확인하기 위해 스택 구현을 사용
  - 여는 괄호 : 스택에 저장
  - 닫는 괄호 : 스택 상단의 괄호와 짝이 되는지 검사
  - 행이 끝난 후, 스택에 여는 괄호가 남아있으면 안됨

Input : [ { ( ) } ]



# 스택을 활용한 괄호 검사 프로그램

## 과제 : `balanced.cpp` 작성

- `istream` 객체 사용
  - `istream is (char s[])`
  - `is.get(char c)`
- `istream` 객체를 사용하기 위해 `istream` 헤더를 추가할 것.

# 스택을 활용한 괄호 검사 프로그램

## 과제 : `balanced.cpp` 작성

- `check_balance()` 함수
  - 괄호 검사를 수행하면서 입력 문자들을 출력
  - 문제가 없으면 " : BALANCED!!" 출력
  - 문제가 발견될 경우, 더 이상 문자를 읽지 않고 " : MISMATCHING!!" 출력



# 스택을 활용한 괄호 검사 프로그램

## 과제 : **balanced.cpp** 작성

- check\_balance() 함수의 psedocode

```
while(there remains a next character c)
    print c
    if(c is an opening parenthesis)
        push c onto the stack
    else if (c is a closing parenthesis
            but doesn't match the top of stack)
    {
        declare MISMATCH and stop
    }

if (the stack is empty)
    declare BALANCED
else
    declare MISMATCH
```

# 과제 제출: 보고서

- 내용
  - 프로그램 소스
  - 실행 화면 캡처
- 제출 형식
  - 스노우보드에서 Lab03\_이름\_학번.doc 다운로드
  - 스노우보드에 제출: Lab03\_이름\_학번.zip
    - 솔루션 폴더 (Lab02에서와 같이 Debug폴더는 모두 제거)
    - 보고서 파일
- 제출 기한
  - 3월 30일(화) PM 11:59 까지
  - 제출 기한 이후 제출 불가(0점 처리)