

02

C++ 프로그래밍의 기본

학습 목표

1. C++ 프로그램의 기본 요소를 이해한다.
2. namespace와 std:: 등 이름 공간을 사용할 수 있다.
3. 키 입력 받는 방법을 안다.
4. 문자열을 입력 받는 방법을 안다.
5. 간단한 C++ 프로그램을 작성할 수 있다.

예제 2-3 C++ 프로그램에서 키 입력 받기

3

```
#include <iostream>
using namespace std;
int main() {
    int width;
    cout << "너비를 입력하세요>>";
    cin >> width; // 키보드로부터 너비를 읽어 width에 저장

    int height;
    cout << "높이를 입력하세요>>";
    cin >> height; // 키보드로부터 높이를 읽어 height에 저장

    int area = width * height; // 사각형의 면적 계산
    cout << "면적은 " << area << "\n"; // 면적을 출력하고 줄바꿈
    return 0;
}
```

너비를 입력하세요>>3<엔터>
높이를 입력하세요>>5<엔터>
면적은 15

cin과 >> 연산자를 이용한 키 입력

4

- `cin >>` 변수;

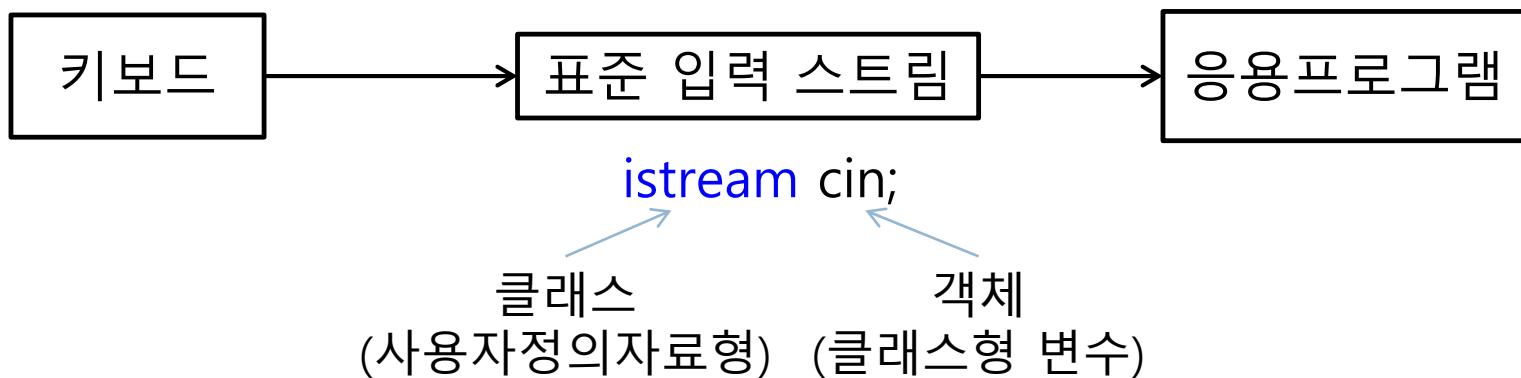
```
#include <iostream>           // cin, >> 선언포함
using namespace std;          // 객체명앞에 std:: 생략
...
int width;                   // 변수선언
cout << "너비를 입력하세요>>"; // 도움말 출력
cin >> width;               // 값저장-> scanf("%d", &width);
```

cin과 >> 연산자를 이용한 키 입력

5

□ cin 객체

- ▣ 표준 입력(키보드) 스트림을 나타내는 객체(변수)
- ▣ <iostream> 헤더 파일에 선언되어 있음
- ▣ std 이름 공간에 선언되어 있음 -> std::cin으로 사용



cin과 >> 연산자를 이용한 키 입력

6

□ >> 연산자

- ▣ 스트림 추출 연산자(stream extraction operator)
- ▣ C/C++에서 시프트 연산자임 -> **a >> 5;**
- ▣ C++에서는 스트림 추출 연산자로 재정의 -> **cin >> x;**
- ▣ 왼쪽 cin 객체에서 값을 읽어 오른쪽 변수에 저장하는 연산자
- ▣ <iostream> 헤더 파일에 선언되어 있음
- ▣ **bool, char, short, int, double 등** 모든 타입의 데이터 입력 가능
- ▣ 연속된 >> 연산자를 사용하여 여러 개의 데이터를 입력 가능

```
int width, height;  
cout << "너비와 높이를 입력하세요>>";  
cin >> width >> height;  
cout << width << ' ' << height << '\n';
```

너비와 높이를 입력하세요>>23 36<엔터>
23 36
// scanf("%d%d", &width, &height);

cin과 >>를 이용한 입력과정

7

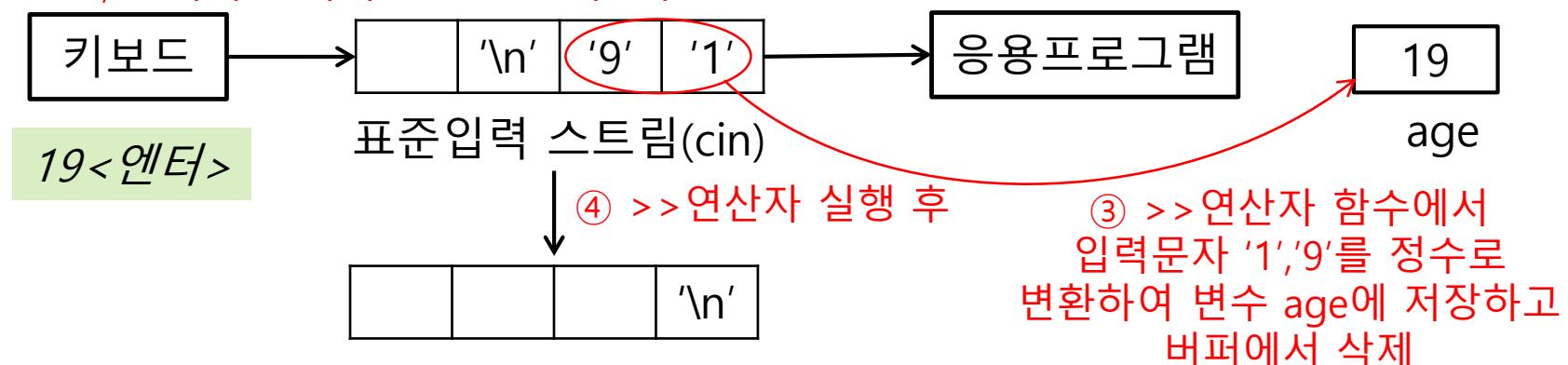
- 버퍼이용, 스트림 버퍼의 데이터를 변수에 저장해줌, 입력 완료전까지 수정가능

```
int age;  
cin >> age;
```

19 <엔터>

- ② 운영체제의 함수가 입력된 문자를 버퍼로 전송, 엔터키 입력되면 >>연산자함수 호출

- ① >>연산자함수에서 버퍼가 비어 있으면 운영체제의 함수 호출하고 대기



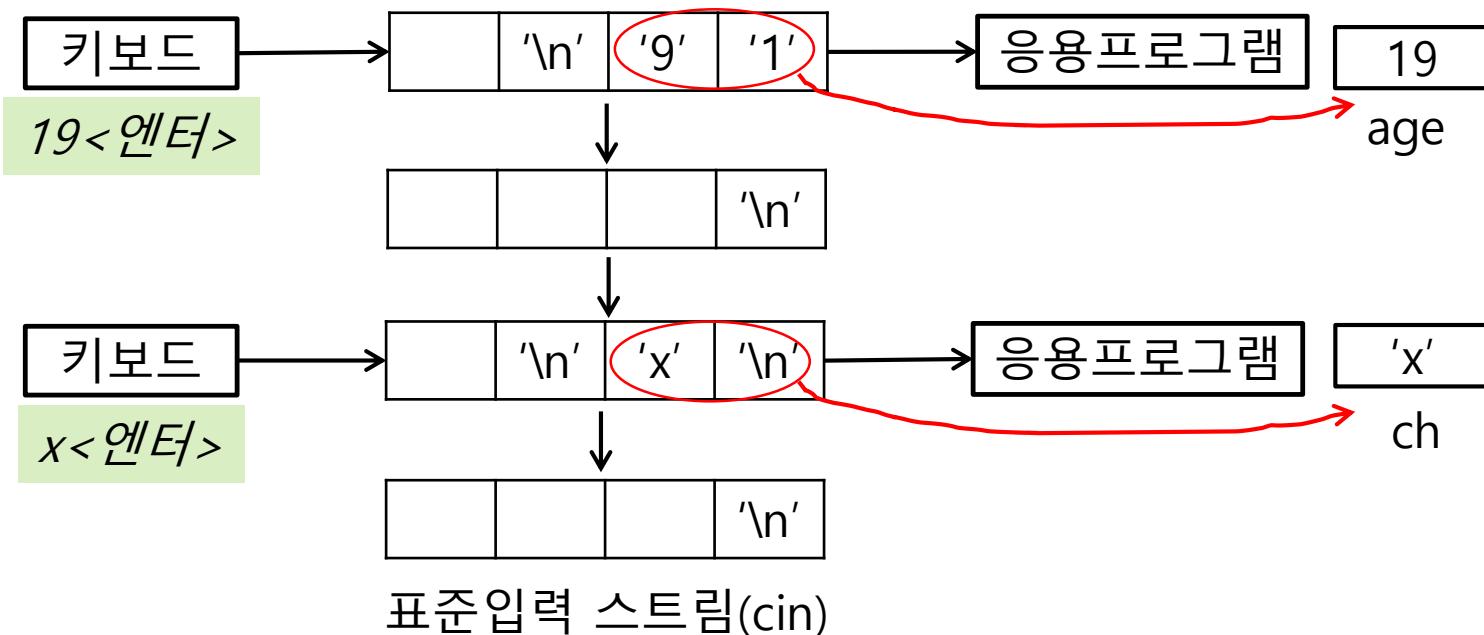
cin과 >>를 이용한 입력과정

8

- >> 연산자는 버퍼에 남아있는 엔터키 무시함-> scanf(" %c",...);

```
int age;  
cin >> age;  
cout << age << endl;  
char ch;  
cin >> ch;  
cout << ch << endl;
```

19<엔터>
19
x<엔터>
x



실행문 중간에 변수 선언

9

- C++언어에서 변수 선언은 아무 곳이나 가능(C언어는 함수 첫 부분에서 모아서 선언)
 - ▣ 장점 : 변수를 사용하기 직전 선언함으로써 변수이름의 타이핑 오류 줄임
 - ▣ 단점 : 변수선언이 흩어져 있으므로 선언된 변수를 찾기 어려움

```
cout << "너비를 입력하세요>>";
int width;                                // 문장들 중간에 변수 선언가능
cin >> width;
cout << " 높이를 입력하세요>> ";
int height;                                // 문장들 중간에 변수 선언가능
cin >> height;
int area = width * height;      // 문장들 중간에 변수 선언가능
cout << "면적은 " << area << "\n";
```

실행문 중간에 변수 선언

10

- for문 안에서 변수 선언 가능
 - ▣ for문의 블록{...}안에서만 사용 가능하고 블록을 벗어나면 사라짐

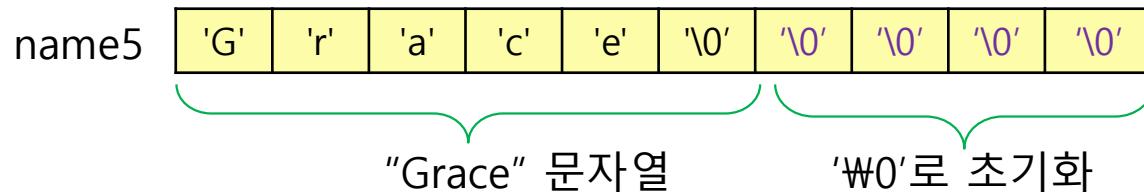
```
for(int n=0; n<10; n++)          // 변수n은 for문안에서만 사용가능
{
    cout << n;
    ...
}
```

C++의 문자열 저장방식

11

- C-스트링 방식 : 문자(char) 배열에 저장

```
char name1[6] = { 'G', 'r', 'a', 'c', 'e', '\0' };    // 문자열 "Grace"
char name2[5] = { 'G', 'r', 'a', 'c', 'e' };        // 단순 문자 배열
char name3[6] = "Grace";
char name4[ ] = "Grace";
char name5[10] = "Grace";
```



□ string 클래스 이용

- C++ 표준 라이브러리에서 제공하는 클래스
 - <string> 헤더 파일에 선언됨
 - 다양한 멤버 함수 제공, 문자열 비교, 복사, 연결, 길이 등

C-스트링 방식으로 문자열 다루기

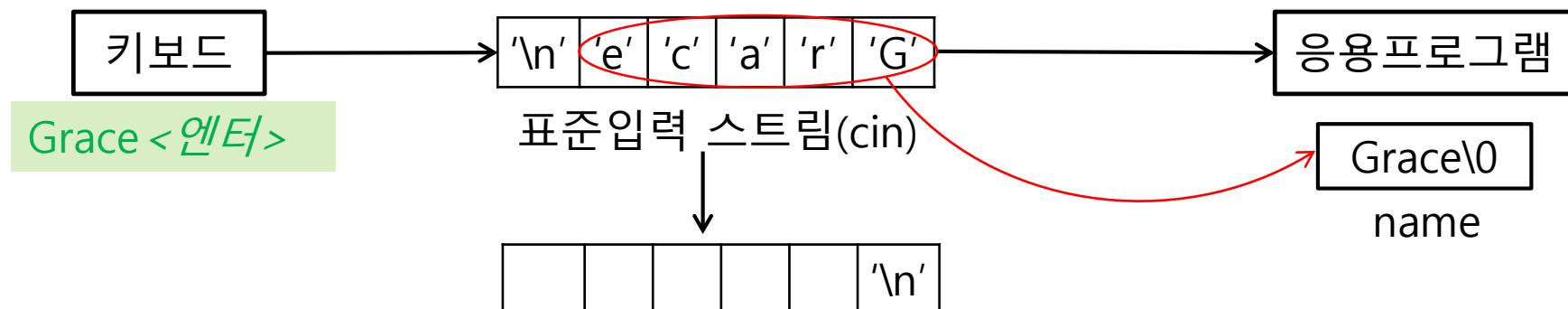
12

- `cin`과 `>>`를 이용한 문자열 입출력

```
char name[6];
cout << "문자열 입력:";
cin >> name;
cout << name;
```

```
// scanf("%s", name);
// printf("%s", name);
```

문자열입력 : Grace<엔터>
Grace



C-스트링 방식으로 문자열 다루기

13

- `cin`과 `>>`를 이용한 문자열 입출력

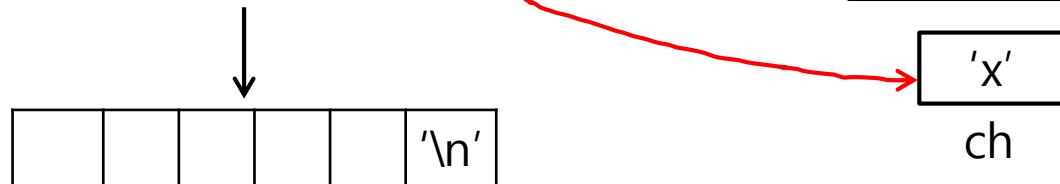
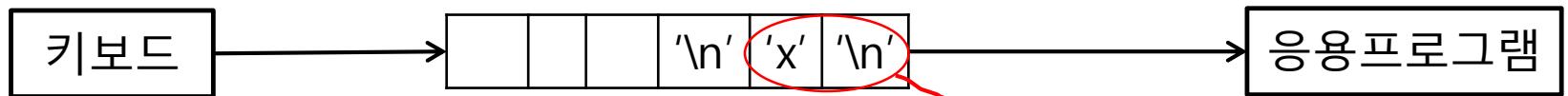
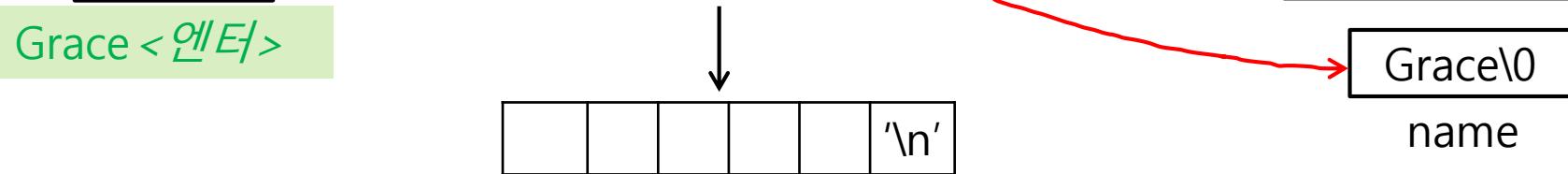
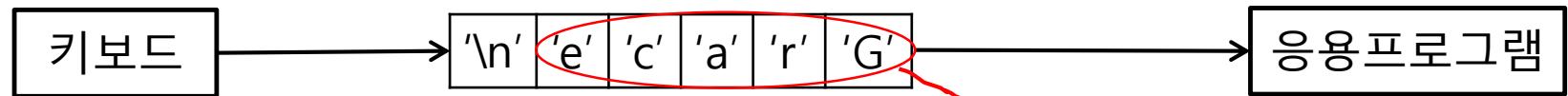
```
char name[6];
cout << "문자열 입력:";
cin >> name;
cout << name;
char ch;
cout << "문자 입력:";
cin >> ch;
cout << ch;
```

문자열입력 : Grace<엔터>
Grace
문자 입력 : x<엔터>
x

C-스트링 방식으로 문자열 다루기

14

- >> 연산자는 버퍼에 남아있는 엔터키 무시함-> scanf(" %c",...);



C-스트링 방식으로 문자열 다루기

15

- C 언어에서 사용한 문자열 처리함수 사용 가능
 - ▣ strcmp(), strlen(), strcpy(), strcat() 등
- <cstring>이나 <string.h> 헤더 파일 include

```
#include <cstring> 또는 #include <string.h>
```

...

```
int n = strlen("hello");
```

- <cstring> 헤더 파일을 사용하는 것이 바람직함
 - ▣ <cstring>이 C++ 표준 방식으로 작성됨

예제 2-4 키보드에서 문자열 입력 받고 출력

16

```
#include <iostream>
using namespace std;
int main() {
    char name[11];
    cout << "이름을 입력하세요>>";
    cin >> name;
    cout << "이름은 " << name << "입니다\n";
    return 0;
}
```

이름을 입력하세요>>마이클<엔터>
이름은 마이클입니다

이름을 입력하세요>>마 이 클<엔터>
이름은 마입니다

예제 2-5 C-스트링을 이용하여 문자열 입력

17

```
#include <iostream>
#include <cstring>
using namespace std;
int main() {
    char password[11];
    cout << "종료하려면 암호를 입력하세요." << endl;
    while (true) {
        cout << "암호>>";
        cin >> password;           // 공백없이 입력해야 함
        if (strcmp(password, "C++") == 0) {
            cout << "프로그램을 정상 종료합니다." << endl;
            break;
        }
        else
            cout << "암호가 틀립니다~~" << endl;
    }
    return 0;
}
```

종료하려면 암호를 입력하세요.
암호>>Java <엔터>
암호가 틀립니다~~
암호>>C <엔터>
암호가 틀립니다~~
암호>>C++ <엔터>
프로그램을 정상 종료합니다.

공백을 포함한 문자열 입력

18

- `cin.getline(char buf[], int size, char delimiterChar);`
 - ▣ buf : 문자열을 저장할 배열의 시작주소
 - ▣ size : 저장할 문자열의 최대 크기, 최대 (size-1)개의 문자 입력 가능
 - ▣ delimiterChar : getline 함수를 종료하게 하는 문자
 - ▣ 버퍼에서 (size-1)개의 문자를 만날 때까지 또는 구분문자(delimiterChar)를 만날 때까지 문자열을 읽어서 배열에 저장하고 마지막에 널문자를 저장하고 함수를 리턴(종료)
 - ▣ 구분문자도 버퍼에서 읽어서 '\0'으로 변경하여 저장하므로 버퍼에 구분문자를 남기지 않음
 - ▣ **입력시 엔터키는 운영체제함수를 종료하게 하고 구분문자는 getline 함수를 종료하게 함**

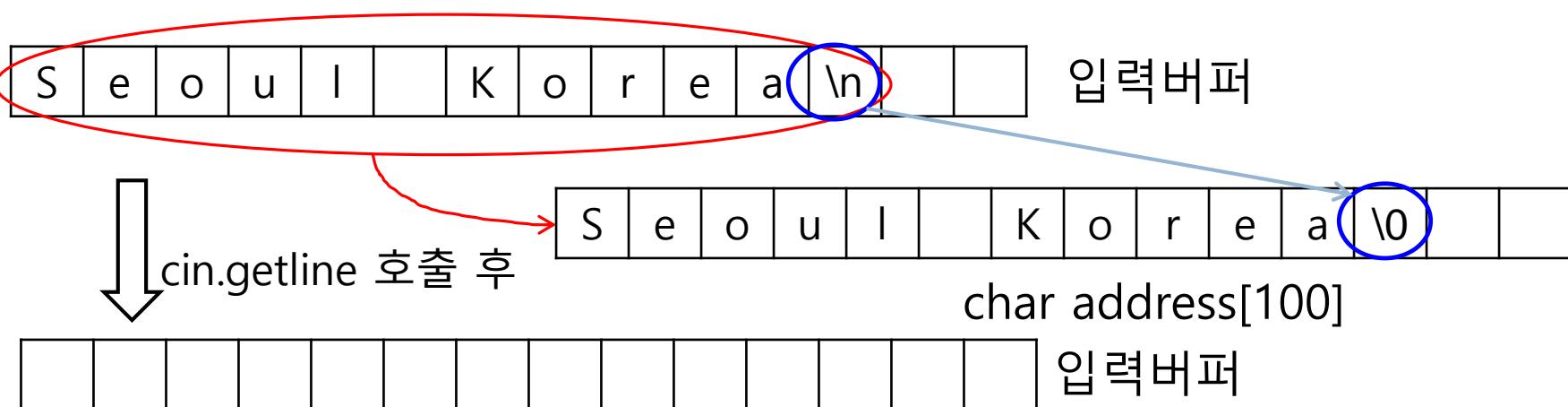
공백을 포함한 문자열 입력

19

- ‘\n’도 읽고 널문자로 변환하여 배열에 저장, 버퍼에는 ‘\n’은 남지 않음 -> fgets 함수(단, ‘\n’은 저장하지 않음)

```
char address[100];
cin.getline(address, sizeof(address), '\n');
```

Seoul Korea <엔터>



공백을 포함한 문자열 입력

20

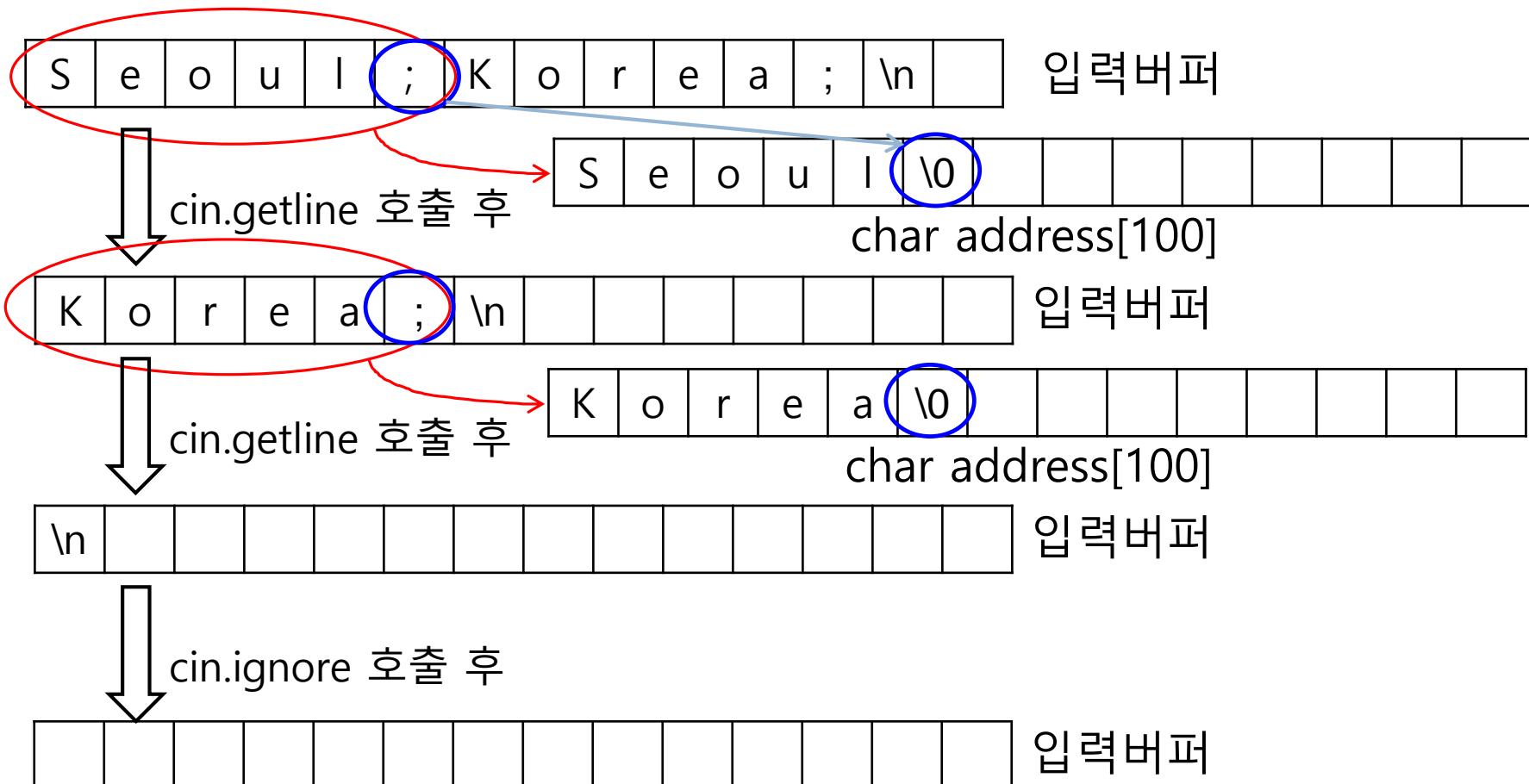
- cin.getline()함수는 구분문자를 만나면 리턴함, 구분문자는 버퍼에서 제거함

```
char address[100];
cin.getline(address, sizeof(address), ';');
cout << address << endl;
cin.getline(address, sizeof(address), ';');
cout << address << endl;
cin.ignore();
```

Seoul;Korea;<엔터>
Seoul
Korea

공백을 포함한 문자열 입력

21



예제 2-6 cin.getline()을 이용한 문자열 입력

22

```
#include <iostream> 주소를 입력하세요:컴퓨터시 프로그램구 C++동<엔터>
using namespace std; 주소는 컴퓨터시 프로그램구 C++동입니다
```

```
int main() {
    cout << "주소를 입력하세요:";
    char address[100];
    cin.getline(address, sizeof(address), '\n'); // 키보드로부터 주소 읽기
    cout << "주소는 " << address << "입니다\n"; // 주소 출력
    return 0;
}
```

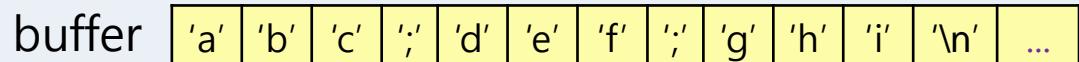
예제 : 구분문자를 이용한 문자열 입력

23

```
#include <iostream>
using namespace std;

int main() {
    cout << "데이터를 입력하세요>>";
    char data[100];
    cin.getline(data, sizeof(data), ';');
    cout << data << endl;
    cin.getline(data, sizeof(data), ';');
    cout << data << endl;
    cin.getline(data, sizeof(data), ';');
    cout << data << endl;
    return 0;
}
```

데이터를 입력하세요>>abc;def;ghi; <엔터>
abc
def
ghi



cin.getline 첫번째 호출후 ->

data

'a'	'b'	'c'	'\0'	...
-----	-----	-----	------	-----

cin.getline 두번째 호출후 ->

data

'd'	'e'	'f'	'\0'	...
-----	-----	-----	------	-----

cin.getline 세번째 호출후 ->

data

'g'	'h'	'i'	'\0'	...
-----	-----	-----	------	-----

string 클래스를 이용하여 문자열 다루기

24

- string 클래스(4.6절에서 자세히 다룸)
 - ▣ C++에서 강력 추천
 - ▣ C++ 표준 클래스 -> 사용자 정의 자료형, 구조체와 비슷
 - ▣ 문자열의 크기에 따른 제약 없음
 - string 클래스가 스스로 문자열 크기게 맞게 내부 버퍼 조절
 - ▣ <string> 헤더 파일에 선언
 - #include <string> 필요
 - ▣ C-스트링보다 다루기 쉬움

표준 C++ 헤더 파일은 확장자가 없다

25

- 표준 C++언어의 헤더 파일은 확장자 없고, std 이름공간에 정의됨
 - #include <iostream>
 - using namespace std;
- 헤더 파일의 확장자 비교

언어	헤더 파일 확장자	사례	설명
C	.h	<string.h>	C/C++ 프로그램에서 사용 가능
C++	확장자 없음	<cstring>	using namespace std;와 함께 사용해야 함

#include <iostream>에 숨은 진실

26

- 생략, 각자 읽어 볼 것

실습과제1

27

- 함수 $Y = x^2 + 2x - 5$ 에서 x 값을 키보드로 입력 받아 y 의 값을 구하여 화면에 실행결과처럼 출력되도록 C++ 프로그램을 작성하시오.
- 여러 번 실행하여 실행할 때마다 입력 값을 변경시키면서 결과를 확인해볼 것

정수 x 를 입력 하시오: 5 <엔터>

$x=5$ 일때 y 의 값은 30입니다.

실습과제2

28

- 다음처럼 학번, 이름, 주소를 입력 받아 변수에 저장하고 출력하는 C++ 프로그램을 작성하시오.
- 힌트: 문제 발생시 키보드 버퍼에 남아있는 엔터키('\n')를 지우는 함수 cin.ignore() 사용할 것

학번을 입력하시오: 192040<엔터>

이름을 입력하시오: 홍길동 <엔터>

주소를 입력하시오: 군산시 대학로 558 <엔터>

1. 학번: 192040

2. 이름: 홍길동

3. 주소: 군산시 대학로 558

실습과제3

29

- 교재 89페이지 4번문제를 푸시오.

실습과제4

30

- 교재 89페이지 5번문제를 푸시오.

실습과제5

31

- 교재 89페이지 7번문제를 푸시오.

과제제출방법

32

□ 과제 제출방법

- ▣ 소스코드, 주석, 실행결과를 포함하는 pdf파일을 작성한 후 eclass 과제 게시판에 업로드, 반드시 하나의 pdf파일로 작성하여 업로드할 것
- ▣ 기한 : 과제 게시판에 마감시간 참조
- ▣ 소스코드의 모든 줄에 주석을 추가한다. (// 사용)
- ▣ 결과화면을 캡쳐할때 글자가 잘 보이도록 명령창 부분만 캡쳐할 것.
- ▣ 소스코드의 첫 부분은 아래그림처럼 제목, 날짜, 작성자(학번, 이름)를 작성 할 것

```
// *****
// 제 목 : 정수 4개의 평균을 구하는 프로그램
// 날 짜 : 2080년 9월10일
// 작성자 : 15010101 김박사
// *****
```