

제3장 함수와 문자열

이번 장에서 만들어볼 프로그램

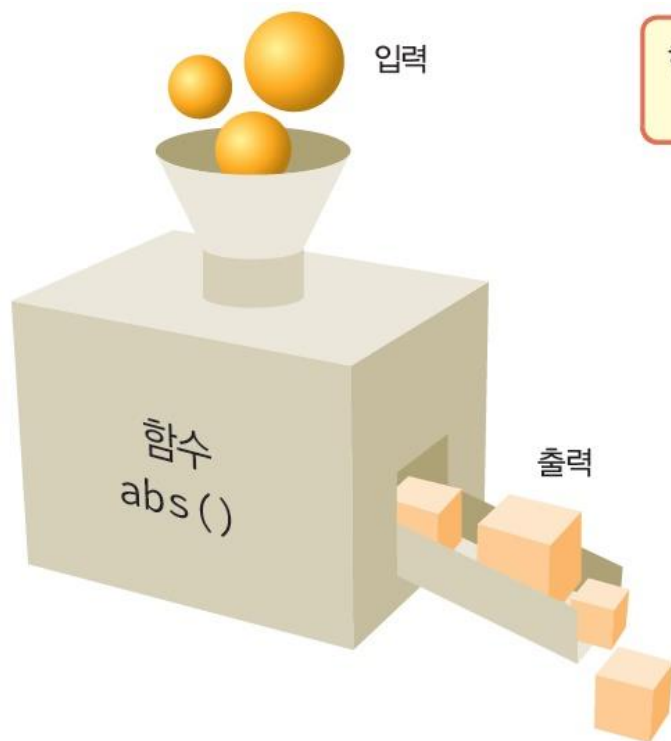
```
C:\Windows\system32\cmd.exe

---
글자를 입력하시오: t
---
글자를 입력하시오: c
C__
글자를 입력하시오: +
c++
성공하였습니다.!계속하려면 아무 키나 누르십시오 . . .
```

```
C:\Windows\system32\cmd.exe

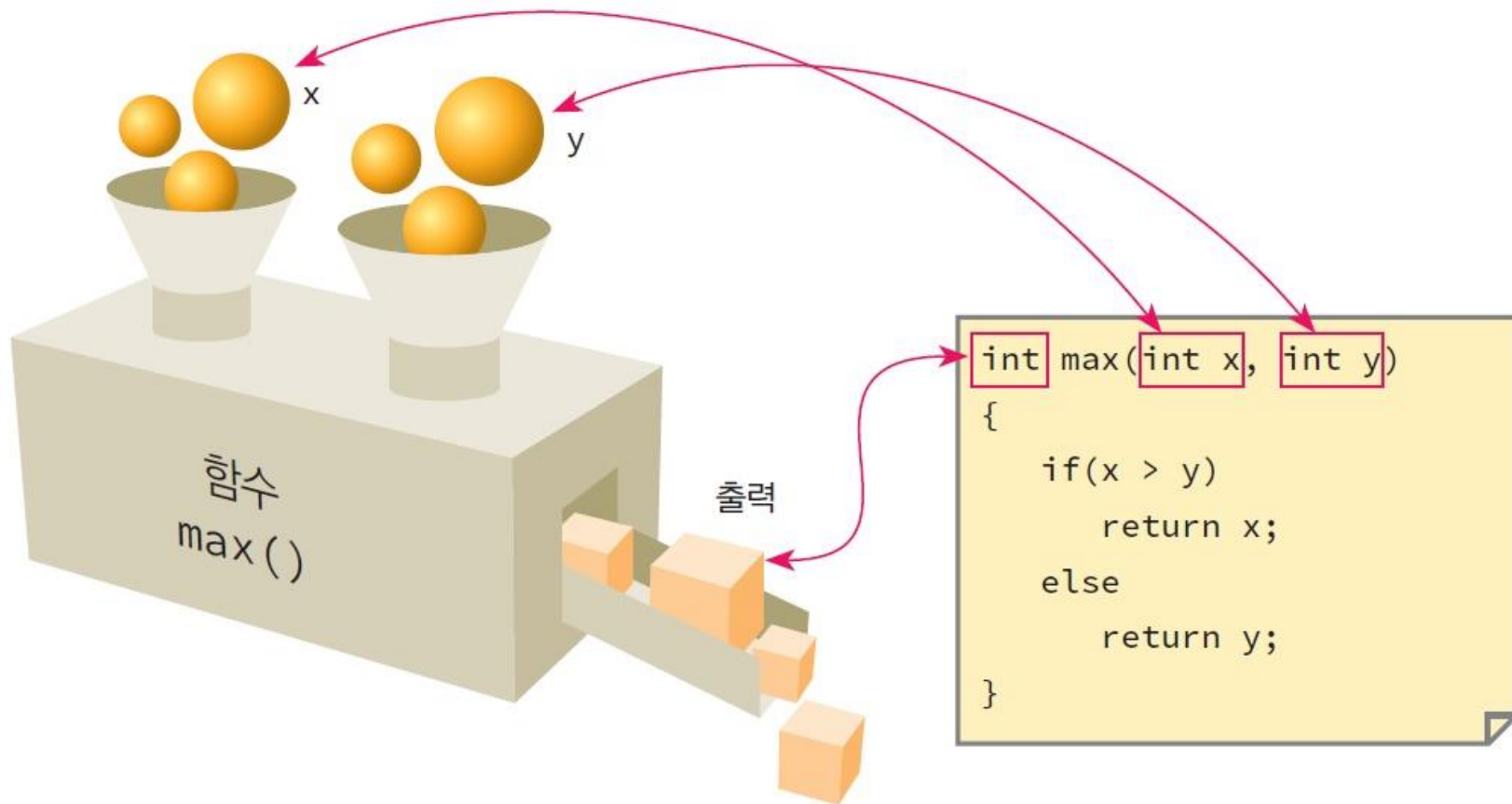
sum(10, 15)=25
sum(10, 15, 25)=50
sum(10, 15, 25, 30)=80
계속하려면 아무 키나 누르십시오 . . .
```

함수란?



함수는 이미 알고
있는데요?

C++에서 많은 기능이
함수에 추가되었습니다.



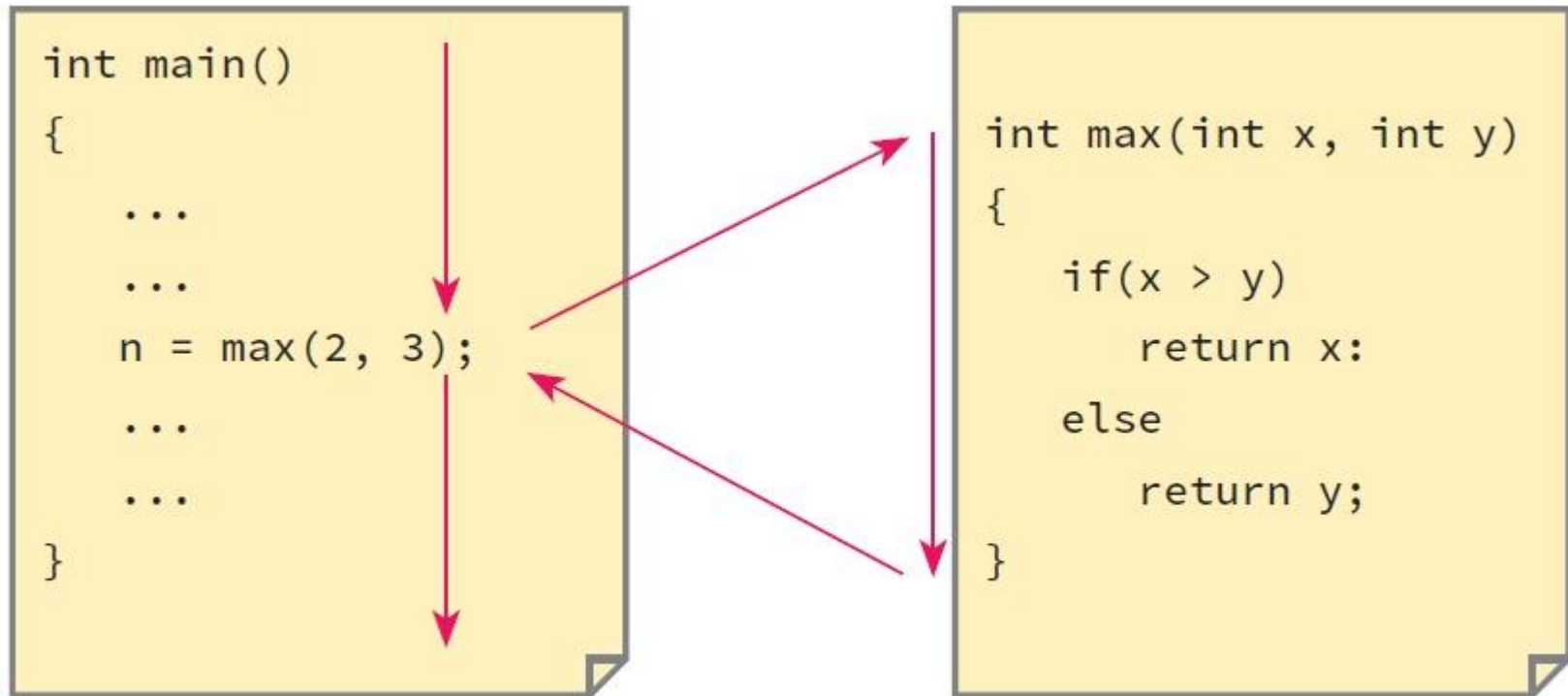


그림 3.1 함수 호출

예제

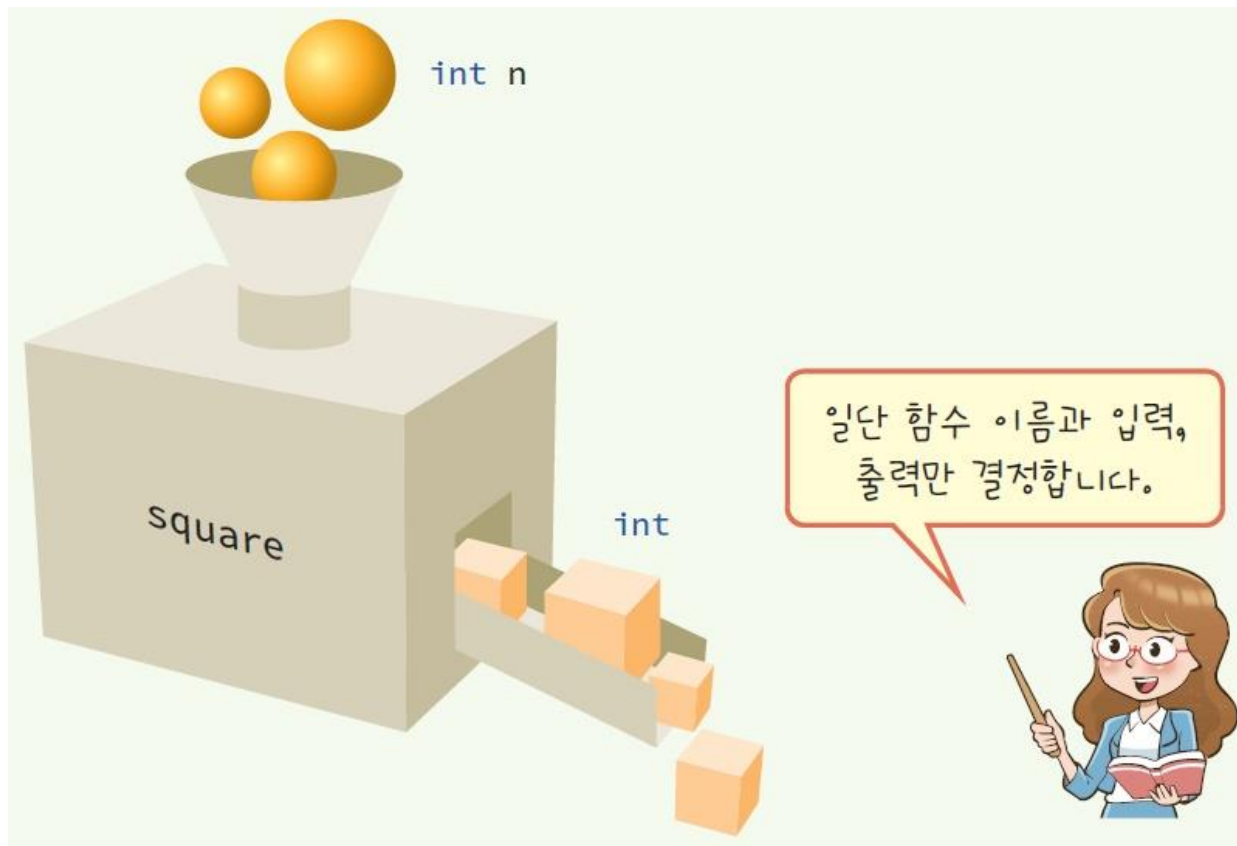
```
#include <iostream>
using namespace std;

// 함수 정의
int max(int x, int y)
{
    if (x>y)
        return x;
    else
        return y;
}

int main()
{
    int n;
    n = max(2, 3);    // 함수 호출
    cout << "연산 결과 = " << n << endl;
    return 0;
}
```

Lab: 함수 만들기

- 정수의 제곱값을 구하는 함수를 만들어보자.



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

int square(int n)
{
    return(n*n);
}

int main()
{
    int n;
    cout << "제공할 정수를 입력하시오: ";
    cin >> n;

    cout << square(n) << endl;
    return 0;
}
```


- 함수 원형(**function prototype**)은 함수의 이름, 매개변수, 반환형을 함수가 정의되기 전에 미리 한번 써주는 것이다.

```
#include <stdio.h>
int square(int n);

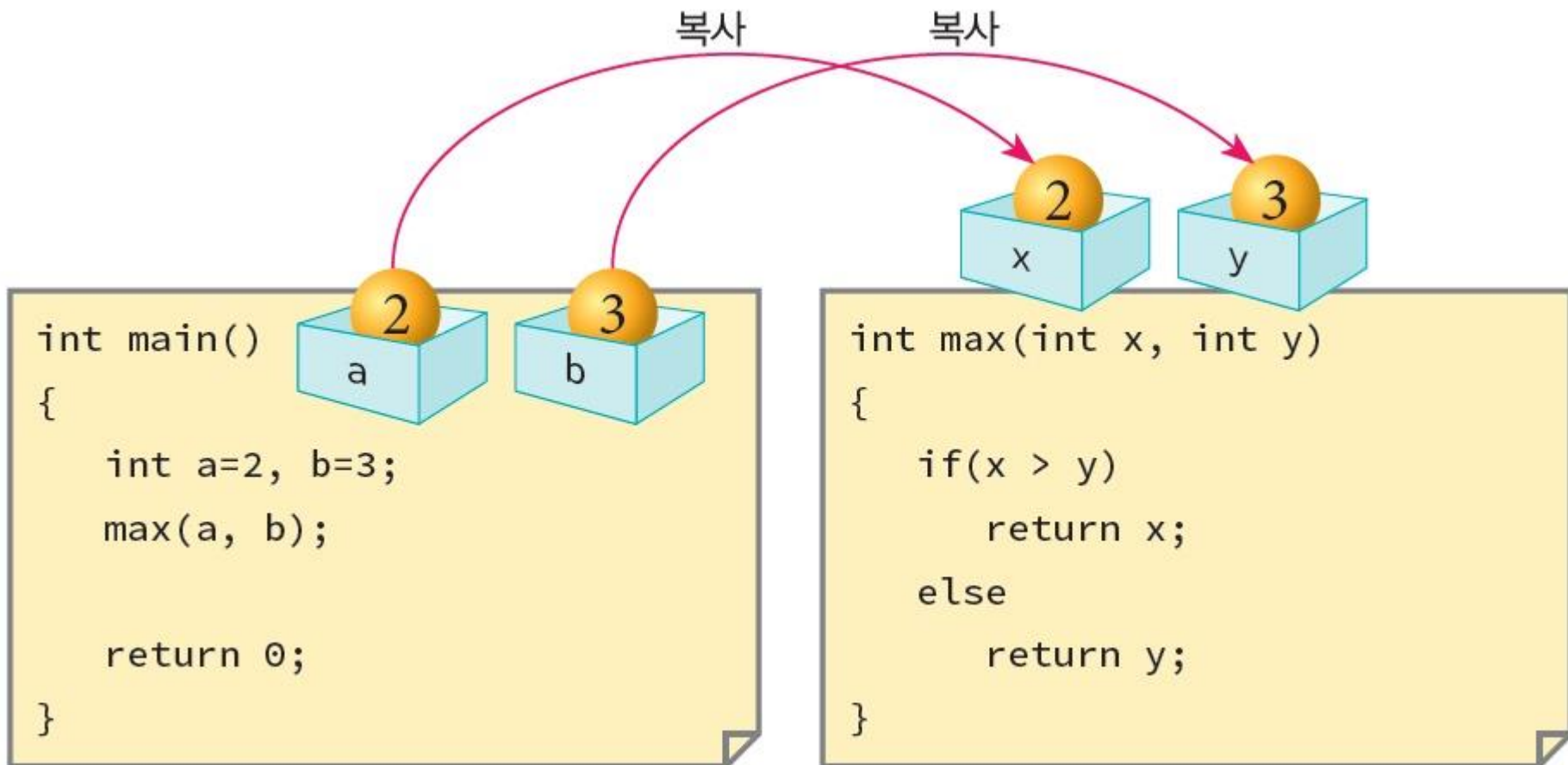
int main()
{
    int result;
    result = square(5);
    printf("%d \n", result);
}

int square(int n)
{
    return(n * n);
}
```

함수 호출시 인수 전달 방법

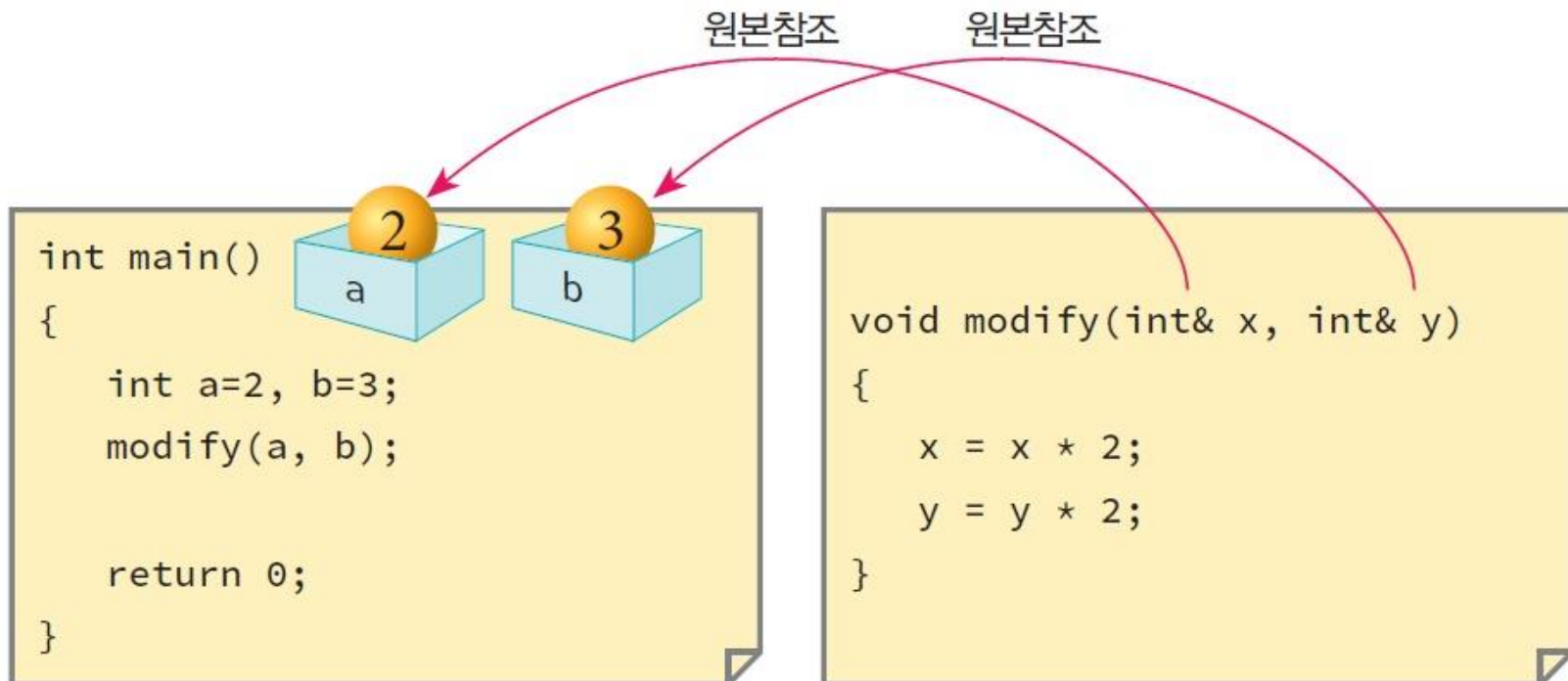
- 값으로 호출하기(call-by-value): 호출하는 곳에서 인수를 전달할 때 인수의 값이 매개 변수로 복사되는 방법이다.
- 참조로 호출하기(call-by-reference): 원본 인수가 함수에 전달되는 방법이다. "참조로 호출하기" 방식에서 함수 안에서 매개 변수는 변경하면 원본 인수가 변경된다.

가으로 전달하기



- 참조자(reference)는 변수의 별명

```
int var = 10;  
int &ref = var;
```



Lab: swap() 함수 만들기

- swap(a, b)와 같이 호출하면 변수 a와 변수 b의 값을 교환

```
int main()
{
    int a = 100, b = 200;

    printf("a=%d b=%d\n", a, b);
    swap(a, b);
    printf("a=%d b=%d\n", a, b);
    return 0;
}
```

Solution:

```
#include <stdio.h>

void swap(int& x, int& y)
{
    int tmp;
    tmp = x;
    x = y;
    y = tmp;
}

int main()
{
    int a = 100, b = 200;

    printf("a=%d b=%d\n", a, b);
    swap(a, b);
    printf("a=%d b=%d\n", a, b);
    return 0;
}
```

- 동일한 이름의 함수를 여러 개 정의하는 것을 **중복 함수 (overloaded functions)**라고 한다.

```
// 정수값을 제공하는 함수
int square(int i)
{
    return i*i;
}

// 실수값을 제공하는 함수
double square(double i)
{
    return i*i;
}
```

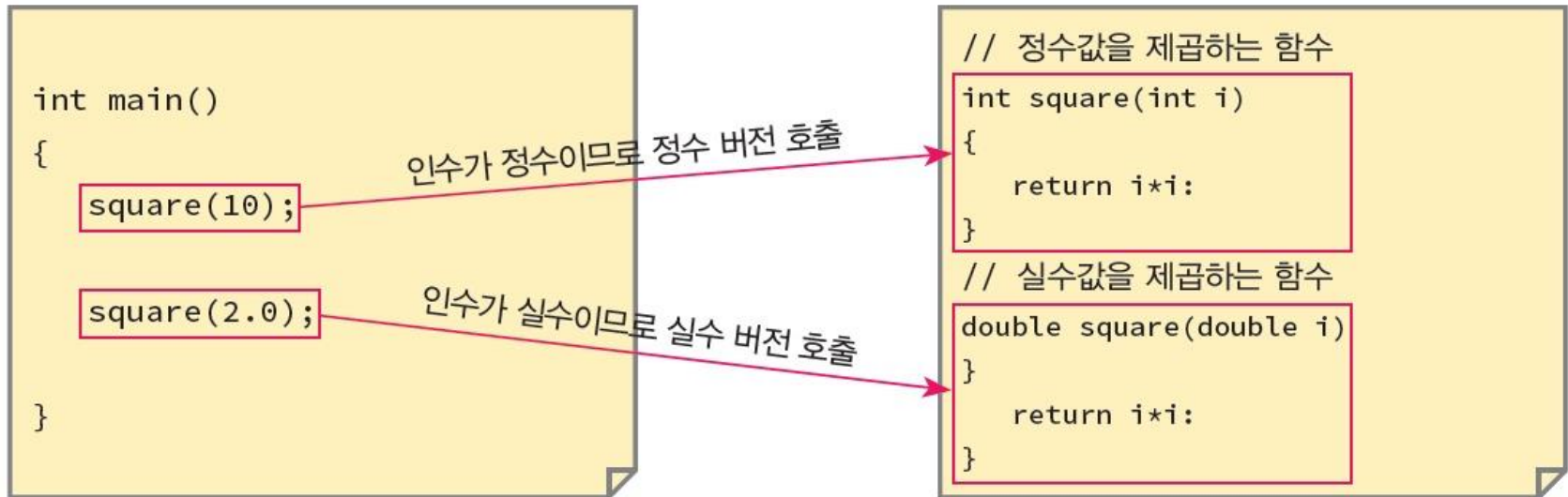


그림 3.3 중복 함수의 개념

예제

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

int square(int i)
{
    cout << "square(int) 호출" << endl;
    return i*i;
}

double square(double i)
{
    cout << "square(double) 호출" << endl;
    return i*i;
}

int main()
{
    cout << square(10) << endl;
    cout << square(2.0) << endl;
    return 0;
}
```

Lab: 중복 함수

- 정수, 실수, 문자를 모두 출력할 수 있는 `print()` 함수를 중복 함수로 정의하고 사용해 보자.



```
C:\Windows\system32\cmd.exe
정수 출력: 100
실수 출력: 3.14
문자 출력: C
계속하려면 아무 키나 누르십시오 . . .
```

예제

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

void print(int i) {
    cout << "정수 출력: " << i << endl;
}

void print(double f) {
    cout << "실수 출력: " << f << endl;
}

void print(char c) {
    cout << "문자 출력: " << c << endl;
}

int main()
{
    print(100);        // 정수를 출력하기 위하여 호출한다.
    print(3.14);       // 정수를 출력하기 위하여 호출한다.
    print('C');        // 문자를 출력하기 위하여 호출한다.
    return 0;
}
```

- 인수를 전달하지 않아도 디폴트값을 대신 넣어주는 기능을 디폴트 인수 (default argument)라고 한다.

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

// 문자 c를 n번 반복하여 화면에 출력한다.
void display(char c = '*', int n = 10)
{
    for (int i = 0; i < n; i++)
        cout << c;
    cout << endl;
}
```

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

void display(char c = '*', int n = 10)
{
    for (int i = 0; i < n; i++)
        cout << c;
    cout << endl;
}

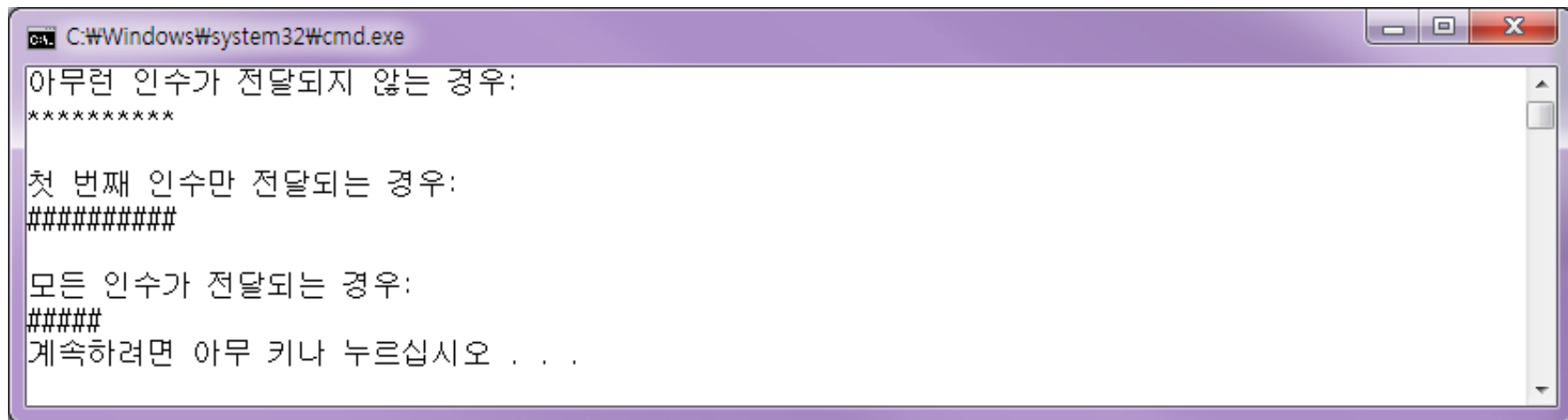
int main()
{
    cout << "아무런 인수가 전달되지 않는 경우:\n";
    display();

    cout << "\n첫 번째 인수만 전달되는 경우:\n";
    display('#');

    cout << "\n모든 인수가 전달되는 경우:\n";
    display('#', 5);

    return 0;
}
```

실행결과



```
C:\Windows\system32\cmd.exe
아무런 인수가 전달되지 않는 경우:
*****

첫 번째 인수만 전달되는 경우:
#####

모든 인수가 전달되는 경우:
#####
계속하려면 아무 키나 누르십시오 . . .
```

Lab: 디폴트 매개 변수 실패하기

```
int main()
{
    cout << "sum(10, 15)=" << sum(10, 15) << endl;
    cout << "sum(10, 15, 25)=" << sum(10, 15, 25) << endl;
    cout << "sum(10, 15, 25, 30)=" << sum(10, 15, 25, 30) << endl;
    return 0;
}
```



```
C:\Windows\system32\cmd.exe
sum(10, 15)=25
sum(10, 15, 25)=50
sum(10, 15, 25, 30)=80
계속하려면 아무 키나 누르십시오 . . .
```

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

int sum(int x, int y, int z = 0, int w = 0)
{
    return x + y + z + w;
}

int main()
{
    cout << "sum(10, 15)=" << sum(10, 15) << endl;
    cout << "sum(10, 15, 25)=" << sum(10, 15, 25) << endl;
    cout << "sum(10, 15, 25, 30)=" << sum(10, 15, 25, 30) << endl;

    return 0;
}
```


- 함수 이름 앞에 **inline**이 붙으면 컴파일러는 함수를 생성하지 않고 함수의 코드를 호출한 곳에 직접 집어넣는다.

```
// 실수값을 제공하는 함수
inline double square(double i)
{
    return i*i;
}
```

□ string 클래스

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o		W	o	r	l	d	!

s[i]
s.empty()
s.insert(pos, s2)
s.remove(pos, len)
s.find(s2)
s.find(pos, s2)
s.reverse()

내부 구현을 몰라도
find()를 사용할 수
있죠!



string 객체

□ 클래스와 객체



string 클래스 사용하기

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

void main()
{
    string s; // string 객체 s를 생성한다.
    string s = "Hello World!"; // string 객체를 생성하고 초기화한다.
    string s{ "Hello World!" }; // string 객체를 생성하고 초기화한다.
}
```

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

void main()
{

string subject = "Money";
string other = " has no value if it is not used";
string sentence = subject + other;
}
```

예제

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

int main()
{
    string s1 = "Slow", s2 = "steady";
    string s3 = "the race.";
    string s4;

    s4 = s1 + " and " + s2 + " wins " + s3;
    cout << s4 << endl;
    return 0;
}
```



C:\Windows\system32\cmd.exe

Slow and steady wins the race.
계속하려면 아무 키나 누르십시오 . . .

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

void main()
{
    string s1 = "Hello", s2 = "World";
    if( s1 == s2 )
        cout << "동일한 문자열입니다." << endl;
    else
        cout << "동일한 문자열이 아닙니다." << endl;
    if( s1 > s2 )
        cout << "s1이 앞에 있습니다. " << endl;
    else
        cout << "s2가 앞에 있습니다. " << endl;
}
```

예제:

- 사용자로부터 이름과 주소를 받아서 친근하게 인사하는 프로그램을 작성



```
C:\Windows\system32\cmd.exe
이름을 입력하시오 : 홍길동
주소를 입력하시오 : 서울특별시 종로구
서울특별시 종로구의 홍길동씨 안녕하세요?
계속하려면 아무 키나 누르십시오 . . .
```



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

int main()
{
    string s1, addr;

    cout << "이름을 입력하시오 : ";
    cin >> s1;
    cin.ignore();      // 엔터키를 없애기 위하여 필요하다.

    cout << "주소를 입력하시오 : ";
    getline(cin, addr);
    cout << addr << "의 " << s1 << "씨 안녕하세요? " << endl;
    return 0;
}
```

string 클래스 멤버 함수 사용

멤버 함수	설명
<code>s[i]</code>	i번째 원소
<code>s.empty()</code>	s가 비어있으면 true 반환
<code>s.insert(pos, s2)</code>	s의 pos 위치에 s2를 삽입
<code>s.remove(pos, len)</code>	s의 pos 위치에 len만큼을 삭제
<code>s.find(s2)</code>	s에서 문자열 s2가 발견되는 첫번째 인덱스를 반환
<code>s.find(pos, s2)</code>	s의 pos 위치부터 문자열 s2가 발견되는 첫번째 인덱스를 반환

- 문자열 "When in Rome, do as the Romans."중에서 "Rome"이 몇 번째 위치에 있는지를 계산하는 프로그램




```
C:\Windows\system32\cmd.exe
8
계속하려면 아무 키나 누르십시오 . . .
```

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

int main()
{
    string s="When in Rome, do as the Romans.";

    int index = s.find("Rome");
    cout << index << endl;
    return 0;
}
```

- 사용자가 입력한 주민등록번호에서 '-' 문자를 삭제하는 프로그램을 작성하여 보자.



```
C:\Windows\system32\cmd.exe
주민등록번호를 입력하시오: 123456-123456
-가 제거된 주민등록번호: 123456123456
계속하려면 아무 키나 누르십시오 . . .
```

string 객체에서 문자 추출하기

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

int main()
{
    string s;
    cout << "주민등록번호를 입력하시오: ";
    cin >> s;

    cout << "-가 제거된 주민등록번호: ";
    for (auto& c : s) {
        if (c == '-') continue;
        cout << c;
    }
    cout << endl;

    return 0;
}
```

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

int main()
{
    string list[] = { "철수", "영희", "길동" };

    for (auto& x : list)
        cout << (x + "야 안녕!") << endl;
    return 0;
}
```



C:\Windows\system32\cmd.exe

```
철수야 안녕!
영희야 안녕!
길동야 안녕!
계속하려면 아무 키나 누르십시오 . . .
```

Lab: 해밍 거리 구하기

- 유전자를 나타내는 2개의 문자열을 받아서 동일한 위치에 틀린 글자가 몇 개나 있는지를 계산하는 프로그램을 작성해보자. 이것을 해밍 거리(Hamming distance)라고 한다



```
C:\Windows\system32\cmd.exe
DNA1: GAGCCTACTAACGGGAT
DNA2: CATCGTAATGACGGCCT
해밍 거리는 7
계속하려면 아무 키나 누르십시오 . . .
```



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

int main() {
    string s1, s2;
    int count = 0;

    cout << "DNA1: ";
    cin >> s1;
    cout << "DNA2: ";
    cin >> s2;

    if (s1.length() != s2.length())
        cout << "오류: 길이가 다름" << endl;
    else {
        for (int i = 0; i < s1.length(); i++) {
            if (s1[i] != s2[i])
                count += 1;
        }
        cout << "해밍 거리는 " << count << endl;
    }
    return 0;
}
```