

# 05



함수와 참조, 복사 생성자

## 학습 목표

1. 값에 의한 호출과 주소에 의해 호출을 복습한다.
2. 함수 호출 시 객체가 전달되는 과정을 이해한다.
3. 객체 치환과 객체 리턴을 이해한다.
4. 참조에 대한 개념을 이해하고, 참조 변수를 선언할 수 있다.
5. 참조에 의한 호출과 참조 리턴에 대해 이해하고 코드를 작성할 수 있다.

# 참조란?

3

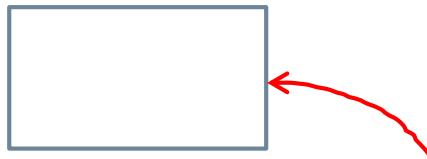
- 참조(reference)
  - ▣ 이미 존재하는 변수나 객체에 대한 별명
- 참조의 활용
  - ▣ 함수의 매개변수(참조에 의한 호출), 함수의 리턴값(참조 리턴)



# 참조 변수

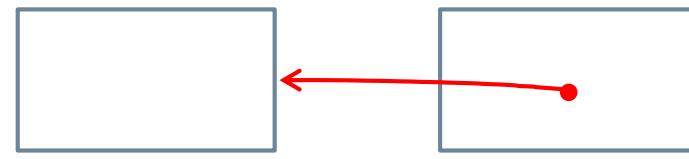
4

- 이미 존재하는 변수(원본변수)에 대한 다른 이름(별명)
- 참조 변수는 이름만 존재하고 메모리 공간이 할당되지 않음
- 원본변수의 공간을 공유 -> 같은 공간을 나타내는 다른 이름
- 변수 선언 시 반드시 원본 변수로 초기화 -> 누구의 별명인지 알려줘야 함, 초기화는 한번만 가능하고 변경이 불가능
- 포인터 변수와 기능이 비슷하나 사용법이 완전히 다름



변수(원본)

참조(별명)



변수(원본)

포인터

# 참조 변수 선언

5

- 자료형& 참조변수명 = 원본변수명;
- 자료형 : 원본변수의 자료형
- & : 참조자, 참조변수임을 선언, 주소연산자 아님

```
int n = 2;  
int& refn = n;           // 참조변수 refn선언, refn은 n에 대한 별명
```

```
Circle circle;  
Circle& refc = circle; //참조변수 refc선언, refc는 circle에 대한 별명
```

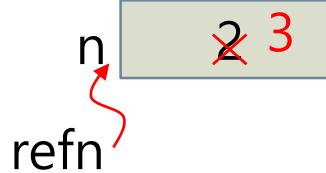
```
int& refn = n;           // &의 위치는 3가지 모두 가능  
int & refn = n;  
int &refn = n;
```

# 참조 변수 사용 방법

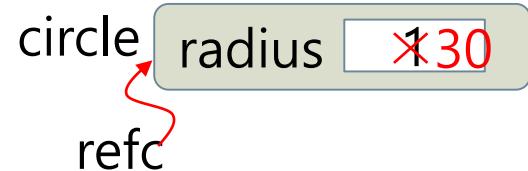
6

- 원본변수와 동일 -> 참조변수에 대한 연산은 원본변수에 대한 연산과 같음

```
int n = 2;  
int& refn = n;  
refn = 3;           // n=3;
```



```
Circle circle;  
Circle& refc = circle;  
refc.setRadius(30);      // circle.setRadius(30);  
refc->setRadius(30);   // error
```



# 참조 변수 선언시 주의 사항

7

```
int n = 2;
```

```
int& refn; // 컴파일 에러, 참조변수는 반드시 초기화해야 함
```

```
& int refn = n; // 컴파일 에러
```

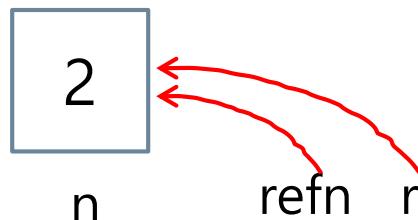
```
int refn & = n; // 컴파일 에러
```

```
char& refn[10]; // 컴파일 에러, 참조변수의 배열을 만들 수 없음
```

```
int n = 2;
```

```
int& refn = n;
```

```
int& r= refn; // 참조 변수를 참조가능 r, refn, n은 같은 공간 공유
```



# 예제 5-3 기본 타입 변수에 대한 참조

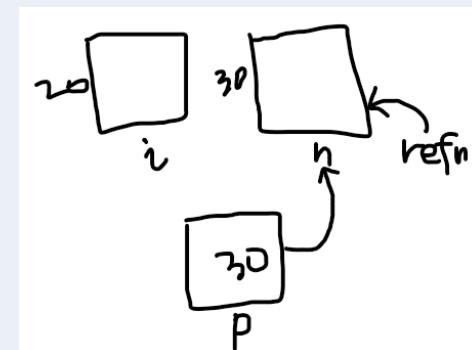
8

```
#include <iostream>
using namespace std;
int main() {
    cout << "i" << '\t' << "n" << '\t' << "refn" << endl;
    int i = 1;
    int n = 2;
    int& refn = n;          // 참조 변수 refn 선언. refn은 n에 대한 별명
    n = 4;
    refn++;                // refn=5, n=5
    cout << i << '\t' << n << '\t' << refn << endl;

    refn = i;               // refn=1, n=1
    refn++;                // refn=2, n=2
    cout << i << '\t' << n << '\t' << refn << endl;

    int* p = &refn;         // p는 n의 주소를 가짐
    *p = 20;                // refn=20, n=20
    cout << i << '\t' << n << '\t' << refn << endl;
}
```

i	n	refn
1	5	5
1	2	2
1	20	20

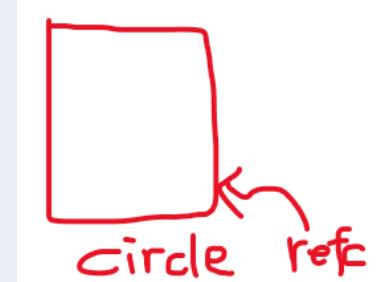


# 예제 5-4 객체에 대한 참조

9

```
#include <iostream>
using namespace std;
class Circle {
    int radius;
public:
    Circle() { radius = 1; }
    Circle(int radius) { this->radius = radius; }
    void setRadius(int radius) { this->radius = radius; }
    double getArea() { return 3.14 * radius * radius; }
};
int main() {
    Circle circle;
    Circle& refc = circle;
    refc.setRadius(10);           // circle.setRadius(10);
    cout << refc.getArea() << " " << circle.getArea();      // circle.getArea();
}
```

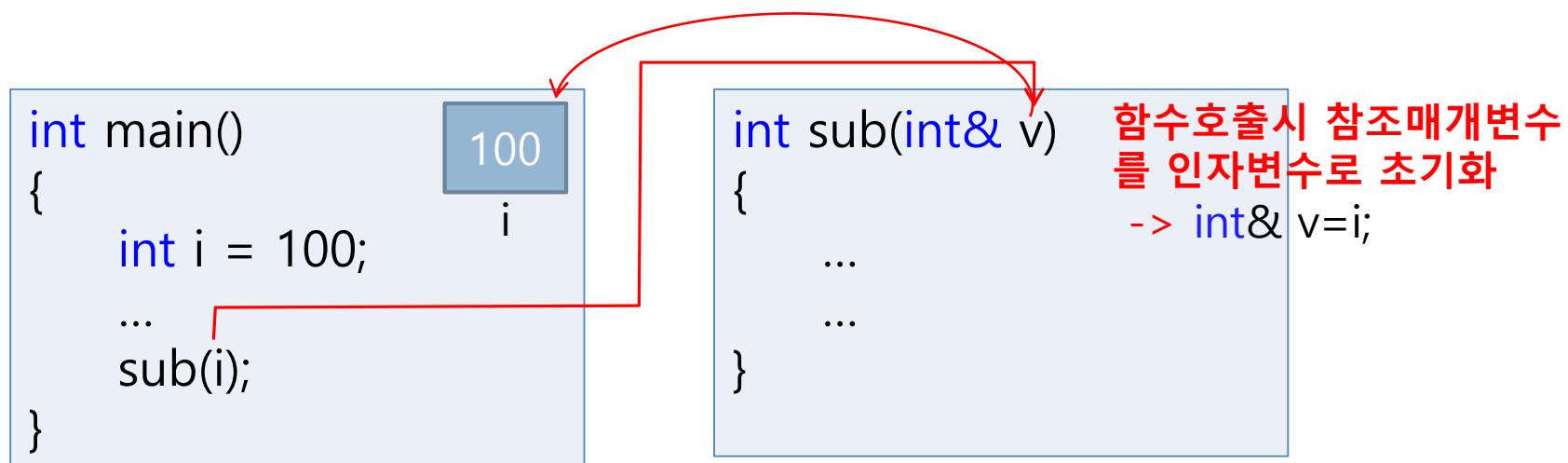
314 314



# 참조에 의한 호출(call by reference)

10

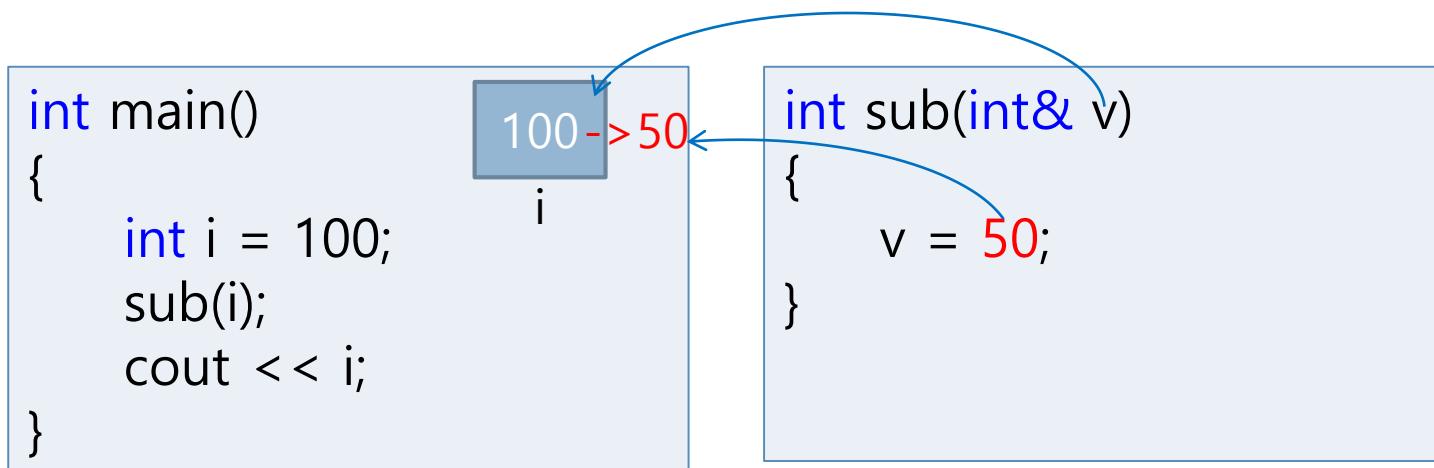
- 함수의 매개 변수를 참조로 선언 -> 참조 매개 변수(reference parameter)라고 부름
- 함수 호출시 참조 매개 변수의 이름만 생기고 메모리 공간은 할당되지 않음
- 인자로 참조 매개 변수를 초기화 -> 인자가 원본이 되고 참조 매개 변수가 인자의 별명이 됨
- 참조 매개 변수는 인자로 전달된 변수의 공간을 공유
- 매개변수 할당과 인자값을 매개변수에 복사하는 과정이 필요 없음**



# 참조에 의한 호출의 장점

11

- 호출하는 방식은 값에 의한 호출 방식과 동일하지만 효과는 주소에 의한 호출 방식과 동일
- 참조 매개 변수에 대한 연산은 인자 변수에 대한 연산이 됨
- 주소에 의한 호출처럼 주소연산자(&)나 간접 참조연산자(\*)와 같은 연산자를 사용할 필요 없음
- 주소에 의한 호출방식이 필요한 경우 참조에 의한 호출을 사용할 것

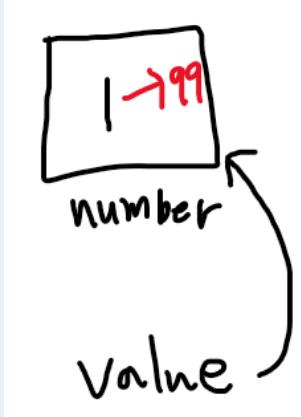


# 참조에 의한 호출 사례

12

```
#include<iostream>
using namespace std;
void modify(int& value);
int main(void)
{
    int number = 1;
    modify(number);
    cout<< "number ="<< number << endl;
    return 0;
}
void modify(int& value) // main함수의 number를 99로 변경하는 함수
{
    value = 99;           // value는 number의 별명
}
```

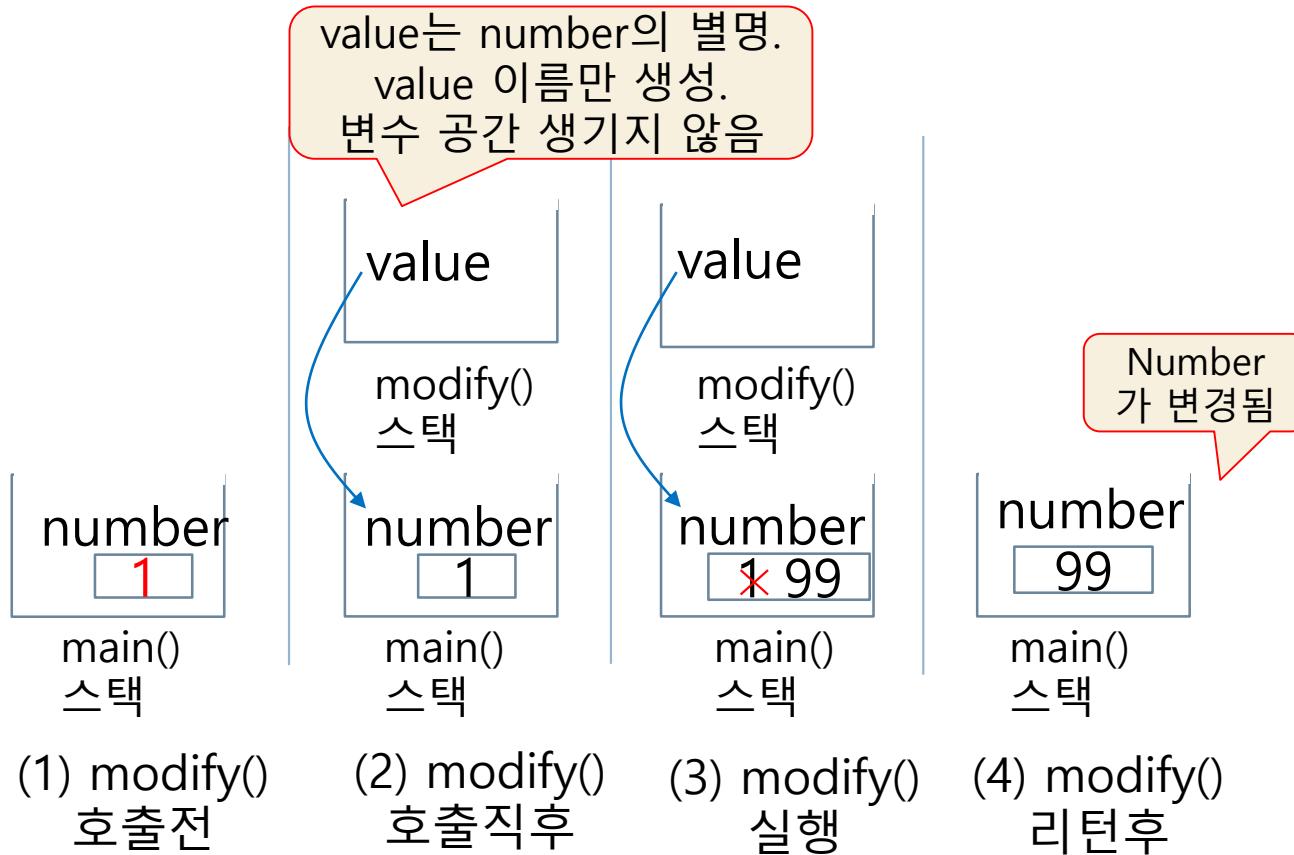
number =99



The diagram shows a rectangular box labeled 'number' containing the number '1'. An arrow labeled 'value' points from the left towards the box. Inside the box, the number '1' is crossed out and replaced by '99', indicating that the modification made through the reference parameter 'value' has changed the original variable 'number'.

# 참조에 의한 호출 사례

13



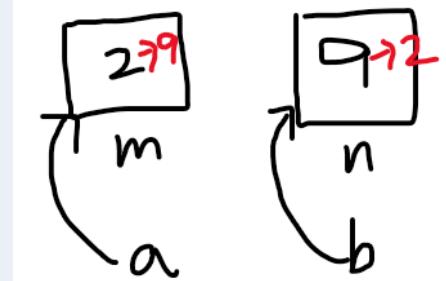
# 참조에 의한 호출 사례

14

```
#include <iostream>
using namespace std;
void swap(int& a, int& b);
int main() {
    int m = 2, n = 9;
    cout << m << ' ' << n;
    swap(m, n);
    cout << m << ' ' << n;
}
void swap(int& a, int& b) {
```

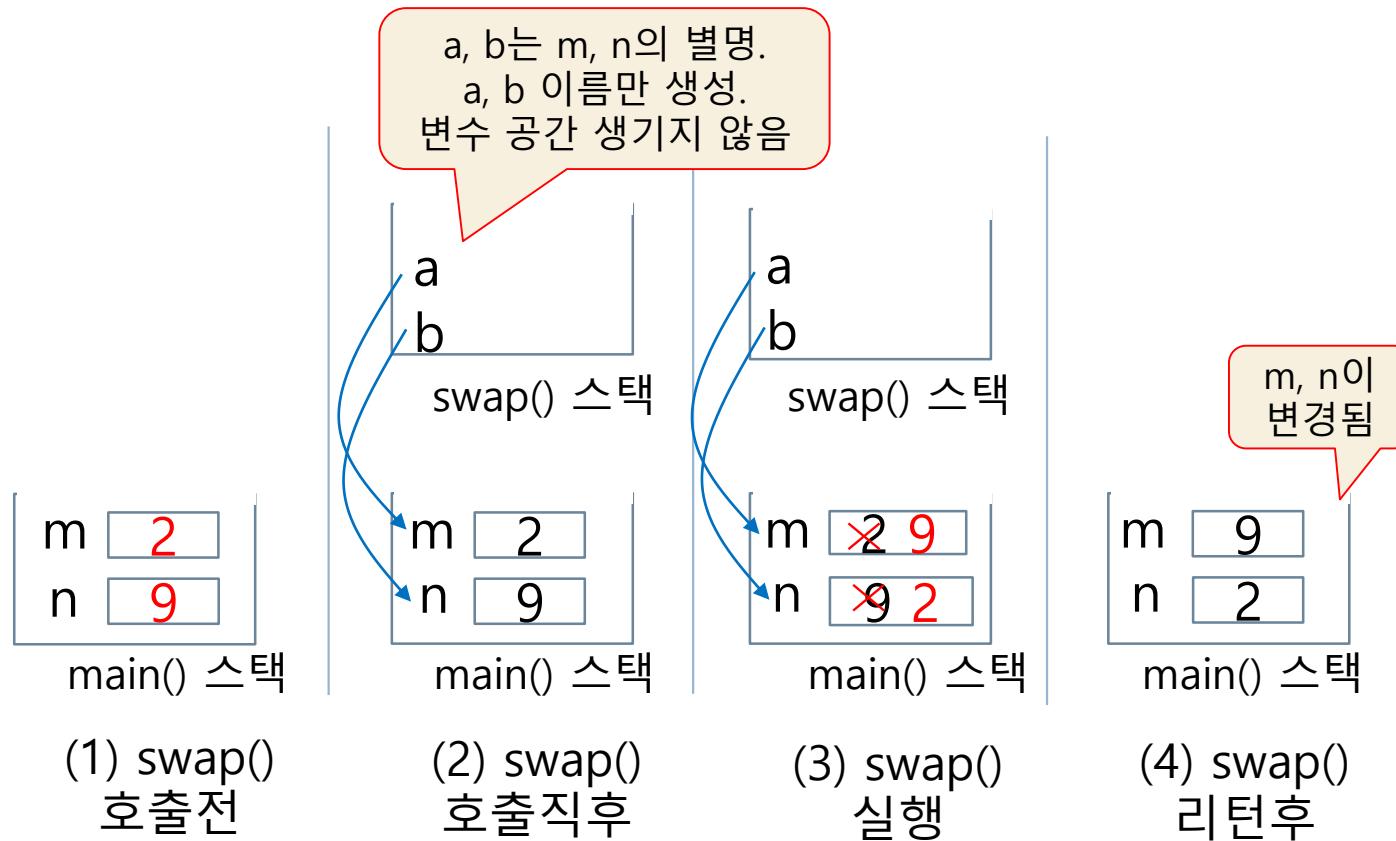
// 호출직후 a,b는 main함수의 m,n의 별명

2 9  
9 2



# 참조에 의한 호출 사례

15



# 예제5-5 참조 매개 변수로 평균 리턴하기

16

```
int average(int a[], int size) {  
    if (size <= 0) return 0;      // 사이즈 오류이면 0을 리턴  
    int sum = 0;  
    for (int i = 0; i < size; i++) sum += a[i];  
    return sum / size;          // 평균을 리턴  
}
```

```
int x[] = { 1,2,3,4 };  
int avg = average(x, 4);      // avg는 2 -> 평균값
```

```
int x[] = { 1,2,3,4 };  
int avg = average(x, -1);     // avg는 0 -> 평균이 0인지 오류인지?
```

- 계산에 오류가 있으면 0을 리턴, 아니면 평균 리턴
- 만일 리턴값이 0인 경우 -> 평균이 0인지 아니면 오류가 발생한 건지 구분 불가능

# 예제5-5 참조 매개 변수로 평균 리턴하기

17

- 참조 매개 변수를 통해 평균을 리턴하고 리턴문을 통해서는 함수의 성공 여부를 리턴하도록 average() 함수를 수정

```
#include <iostream>
using namespace std;
bool average(int a[], int size, int& avg) {
    if (size <= 0)
        return false;
    int sum = 0;
    for (int i = 0; i < size; i++)
        sum += a[i];
    avg = sum / size;           // 참조 매개 변수 avg에 평균 값 전달
    return true;
}
```

평균은 2  
매개 변수 오류

# 예제5-5 참조 매개 변수로 평균 리턴하기

18

```
int main() {  
    int x[] = { 0,1,2,3,4,5 };  
    int avg;  
  
    if (average(x, 6, avg)) cout << "평균은 " << avg << endl;  
    else cout << "매개 변수 오류" << endl;  
  
    if (average(x, -2, avg)) cout << "평균은 " << avg << endl;  
    else cout << "매개 변수 오류 " << endl;  
}
```

평균은 2  
매개 변수 오류

# 실습과제1

19

- 참조에 의한 호출과 주소에 의한 호출의 차이점을 설명하라.
- 참조에 의한 호출의 장점은 무엇인가?

# 실습과제2

20

- 아래 코드에서 add2함수는 매인함수의 변수 number를 2만큼 증가시키는 함수이다. 그러나 실행결과에서 보듯이 원하는 대로 작동하지 않는데 이유를 설명하고 원하는 결과가 나오도록 코드를 수정하시오.

```
#include<iostream>
using namespace std;
void add2(int value);
int main(void)
{
    int number;
    cout << "정수를 입력하세요 : ";
    cin >> number;
    add2(number);
    cout << "2만큼 증가한 값 :" << number << endl;
    return 0;
}
void add2(int value)
{
    value += 2;
}
```

정수를 입력 하시오: 15 <엔터>  
2만큼 증가한 값: 15

# 실습과제3

21

- 앞에서 나온 swap함수 예제를 아래 결과가 나오도록 코드를 수정 하시오. 두 정수를 교환하는 부분은 반드시 참조에 의한 호출을 이용한 함수 (swap)로 작성 하시오. 함수의 선언, 호출, 정의를 모두 사용하시고 C++ 언어를 사용 하시오.

정수x를 입력 하시오: 15 <엔터>

정수y를 입력 하시오: 22 <엔터>

swap함수 호출 전 x=15, y=22

swap함수 호출 후 x=22, y=15

# 실습과제4

22

- 키보드로 부터 실수를 입력 받아 그것의 정수부와 소수부를 구하여 출력 해주는 프로그램을 작성하시오.
- 정수부와 소수부를 구하여 리턴하는 부분을 1개의 함수 get\_parts로 작성할 것
- 화면에 입출력하는 부분은 함수 get\_parts에서 하지 말고 메인 함수에서 작성할 것
- 함수의 정의, 호출, 선언을 모두 작성하시오.

실수를 입력하시오 : 3.14159<엔터>  
정수부 : 3  
소수부 : 0.14159

# 과제제출방법

23

- 소스코드, 라인단위의 주석, 실행결과를 포함하는 pdf파일을 작성한 후 eclass 과제 게시판에 업로드, **반드시 하나의 pdf파일로 업로드할 것**
- 기한 : 과제 게시판에 마감시간 참조
- 실행결과를 캡쳐할 때 글자를 알아보기 쉽게 확대해서 캡쳐할 것.
- 소스코드의 첫 부분은 아래처럼 제목, 날짜, 작성자(학번, 이름)를 작성할 것

```
// *****
// 제 목 : 정수 4개의 평균을 구하는 프로그램
// 날 짜 : 2023년 9월 10일
// 작성자 : 15010101 홍길동
// *****

// 소스코드 작성
```