



06

함수 중복과 static 멤버

학습 목표

1. 함수 중복의 개념을 이해하고, 중복 함수를 작성할 수 있다.
2. 디폴트 매개 변수를 이해하고 작성할 수 있다.
3. 함수 중복 시 발생하는 모호성의 경우를 판별할 수 있다.
4. static 속성으로 선언된 멤버의 특성을 이해하고, static 속성을 활용할 수 있다.

디폴트 매개 변수(default parameter)

3

- 디폴트값(초기값)이 지정되어 있는 매개변수
- **자료형 매개변수 = 디폴트값 형태로 선언**
- 함수 호출전에 디폴트값을 선언해야 함(함수 선언할 때)
- 함수 호출시 인자가 전달되지 않는 경우, 디폴트값(초기값)이 전달됨 -> 인자 없이도 호출가능
- 디폴트 매개 변수 선언 및 호출

```
void star(int a = 5); // a : 디폴트 매개변수, a의 디폴트값은 5
```

```
star(); // 매개 변수 a에 디폴트값 5가 전달됨 -> star(5);
star(10); // 매개 변수 a에 10이 전달됨
```

디폴트 매개 변수 사례

4

- 디폴트 매개변수는 인자 없이 함수 호출 가능
- 일반 매개변수는 호출 시 반드시 인수를 지정해야 함

```
void msg(int id, string text = "Hello");
```

일반 매개변수

디폴트 매개변수

```
msg(10);                      // msg(10, "Hello"); 호출과 동일
msg(20, "Good Morning");      // id->20, text->"Good Morning"
msg();                         // 컴파일 오류
msg("Hello");                 // 컴파일 오류
```

디폴트 매개 변수에 관한 제약 조건

5

- 디폴트 매개 변수는 일반 매개 변수 앞에 선언될 수 없음 ->
디폴트 매개 변수는 뒤쪽에 모아서 선언해야 함

```
void calc(int a, int b = 5, int c, int d = 0);      // 컴파일 오류  
void sum(int a = 0, int b, int c);                  // 컴파일 오류  
void calc(int a, int b = 5, int c = 0, int d = 0);  // 컴파일 성공
```

매개변수에 값을 정하는 규칙

6

- 함수 호출문에 나열된 인자를 순서대로 매개 변수에 전달하고 나머지는 디폴트값으로 전달

```
void square(int width = 1, int height = 1);
```

square(); → square(__, __); → square(1, 1);
square(5); → square(5, __); → square(5, 1);
square(3, 8); → square(3, 8); → square(3, 8);

컴파일러에 의해 변환되는 과정

매개변수에 값을 정하는 규칙

7

```
void g(int a, int b = 0, int c = 0, int d = 0);
```

g(10); → g(10, , ,); → g(10, 0, 0, 0);

g(10, 5); → g(10, 5, ,); → g(10, 5, 0, 0);

g(10, 5, 20); → g(10, 5, 20,); → g(10, 5, 20, 0);

g(10, 5, 20, 30); → g(10, 5, 20, 30); → g(10, 5, 20, 30);

컴파일러에 의해 변환되는 과정

예제6-3 디폴트 매개변수를 가진 함수 선언 및 호출

8

```
#include <iostream>
#include <string>
using namespace std;
void star(int a = 5);
void msg(int id, string text = ""); // 함수 호출전에 선언해야 함
int main() {
    star();           // star(5)
    star(10);

    msg(10);         // msg(10,"")
    msg(10, "Hello");
    return 0;
}
void star(int a) {
    for (int i = 0; i < a; i++)
        cout << '*';
    cout << endl;
}
void msg(int id, string text) {
    cout << id << ' ' << text << endl;
}
```


10
10 Hello

```
void star(int a = 5) {
    for (int i = 0; i < a; i++)
        cout << '*';
    cout << endl;
}

void msg(int id, string text = "") {
    cout << id << ' ' << text
        << endl;
}

// 함수 정의시 디폴트값을 선언
// 할때는 함수 정의가 호출전에 나
// 와야함
```

예제6-4 디폴트 매개 변수를 가진 함수 만들기 연습

9

- 함수 f()를 호출하는 경우가 다음과 같을 때 f()를 디폴트 매개 변수를 가진 함수로 작성하라.

```
f();           // 한 줄에 빈칸을 10개 출력한다.  
f('%');       // 한 줄에 '%'를 10개 출력한다.  
f('@', 5);    // 다섯 줄에 '@'를 10개 출력한다.
```

```
%%%%%%%%%%%%%%  
@oooooooooooo  
oooooooooooo  
oooooooooooo  
oooooooooooo  
oooooooooooo
```

예제6-4 디폴트 매개 변수를 가진 함수 만들기 연습

10

```
#include <iostream>
using namespace std;
void f(char c = ' ', int line = 1);    // 함수 선언
int main() {
    f();                      // 한줄에 빈칸을 10개 출력한다.
    f('%');                   // 한 줄에 '%'를 10개 출력한다.
    f('@', 5);                // 5 줄에 '@' 문자를 10개 출력한다.
    return 0;
}
void f(char c, int line) {    // 함수 구현
    for (int i = 0; i < line; i++) {
        for (int j = 0; j < 10; j++)
            cout << c;
        cout << endl;
    }
}
```

빈 칸이 10
개 출력됨

%%%%%%%%%%%%%%
@@@@@@@aaaaaa
@@@aaaaaaa@aa
@@@aaaaaaa@aa
@@@aaaaaaa@aa
@@@aaaaaaa@aa

함수 중복 간소화

11

- 디폴트 매개 변수를 활용하면 중복된 함수들을 간소화

```
class Circle {  
    ....  
    public:  
        Circle() { radius = 1; }  
        Circle(int r) { radius = r; }  
    ....  
};
```

```
Circle x;          // Circle()  
Circle y(10);     // Circle(10)
```

```
class Circle {  
    ....  
    public:  
        Circle(int r = 1) { radius = r; }  
    ....  
};
```

```
Circle x;          // Circle()->Circle(1)  
Circle y(10);     // Circle(10)
```

2 개의 생성자 함수를
디폴트 매개 변수를 가진
하나의 함수로 간소화

함수 중복 간소화

12

- 중복 함수들과 디폴트 매개 변수를 가진 함수를 함께 사용 불가

```
class Circle {  
    ....  
    public:  
        Circle() { radius = 1; }  
        Circle(int r) { radius = r; }  
        Circle(int r = 1) { radius = r; }  
        ....  
};
```

중복된 함수와
동시 사용 불가

예제6-5 디폴트 매개변수에 의한 중복 함수 간소화

13

- 다음 두개의 중복 함수를 디폴트 매개변수를 가진 하나의 함수로 작성하라.

```
void fillLine() {                                // 25 개의 '*' 문자를 한 라인에 출력
    for (int i = 0; i < 25; i++) cout << '*';
    cout << endl;
}
void fillLine(int n, char c) {                  // n개의 c 문자를 한 라인에 출력
    for (int i = 0; i < n; i++) cout << c;
    cout << endl;
}
```

예제6-5 디폴트 매개변수에 의한 중복 함수 간소화

14

```
#include <iostream>
using namespace std;
void fillLine(int n = 25, char c = '*') { //n개의 c문자를 한라인에 출력
    for (int i = 0; i < n; i++) cout << c;
    cout << endl;
}

int main() {
    fillLine();           // 25개의 '*'를 한 라인에 출력
    fillLine(10, '%');   // 10개의 '%'를 한 라인에 출력
    return 0;
}
```

%%%%%

예제6-6 생성자 함수의 중복 간소화

15

- 다음 클래스에 중복된 생성자를 디폴트 매개 변수를 가진 하나의 생성자로 작성하라.

```
class MyVector {  
    int* p;  
    int size;  
    public:  
        MyVector() {  
            p = new int[100];  
            size = 100;  
        }  
        MyVector(int n) {  
            p = new int[n];  
            size = n;  
        }  
        ~MyVector() { delete[ ] p; }  
};
```

예제6-6 생성자 함수의 중복 간소화

16

```
#include <iostream>
using namespace std;
class MyVector {
    int* p;
    int size;
public:
    MyVector(int n = 100) {
        p = new int[n];
        size = n;
    }
    ~MyVector() { delete[ ] p; }
};
int main() {
    MyVector* v1;
    MyVector* v2;
    v1 = new MyVector;
    v2 = new MyVector(1024);

    delete v1;
    delete v2;
    return 0;
}
```

위임 생성자로 작성할 수도 있음(3.4절참고)

```
MyVector() : MyVector(100) { }
MyVector(int n) {
    p = new int[n];
    size = n;
}
```

// 디폴트로 정수 100개의 배열 동적 할당
// 정수 1024개의 배열 동적 할당

함수 중복의 모호성

17

- 함수 중복이 모호하여 컴파일러가 어떤 함수를 호출하는지 판단하지 못하는 경우
 - ▣ 형 변환으로 인한 모호성
 - ▣ 참조 매개 변수로 인한 모호성
 - ▣ 디폴트 매개 변수로 인한 모호성

형 변환으로 인한 함수 중복의 모호성

18

- 실인자와 매개변수가 자료형이 다를 경우 실인자가 매개변수의 자료형으로 자동형변환 됨
- 자동형변환은 범위가 작은 타입을 큰 타입으로 변환(`char -> int -> float -> double`)

```
double square(double a) {
    return a * a;
}
int main() {
    cout << square(3); // 3(int)이 3.0(double)으로 자동형변환
    return 0;
}
```

(a) 정상 컴파일

형 변환으로 인한 함수 중복의 모호성

19

□ 매개 변수의 형 변환으로 인한 중복 함수 호출의 모호성

```
float square(float a) {  
    return a * a;  
}  
double square(double a) {  
    return a * a;  
}  
int main() {  
    cout << square(3.0f);           // 3.0f은 float형 -> 1번째함수 호출  
    cout << square(3.0);           // 3.0은 double형 -> 2번째함수 호출  
    cout << square(3);            // 컴파일 오류 -> 3(int)은 float와  
                                // double로 모두 형변환 가능  
}
```

(b) 모호한 호출, 컴파일 오류

예제6-7 형변환으로 인해 함수중복이 모호한 경우

20

```
#include <iostream>
using namespace std;
float square(float a) {
    return a * a;
}
double square(double a) {
    return a * a;
}
int main() {
    cout << square(3.0);           // square(double a); 호출
    cout << square(3);            // 컴파일 오류
    return 0;
}
```

예제6-8 참조 매개변수로 인한 함수중복의 모호성

21

```
#include <iostream>
using namespace std;
int add(int a, int b) {
    return a + b;
}
int add(int a, int& b) {
    b = b + a;
    return b;
}
int main() {
    int s = 10, t = 20;
    cout << add(s, t); // 컴파일 오류
    return 0;
}
```

두 함수는 근본적으로 중복 시킬 수 없다.

call by value인지
call by reference인지 모호

예제6-9 디폴트 매개변수로 인한 함수중복의 모호성

22

```
#include <iostream>
#include <string>
using namespace std;
void msg(int id) {
    cout << id << endl;
}
void msg(int id, string s = "") {
    cout << id << ":" << s << endl;
}
int main() {
    msg(5, "Good Morning"); // 정상 컴파일. 두 번째 msg() 호출
    msg(6); // 함수 호출 모호 -> 컴파일 오류
    return 0;
}
```

실습과제1

23

- 디폴트 매개변수는 함수의 선언에서 정의해주어야 한다.
이유를 설명하라.
- 만약 함수의 정의에서 해준다면 어떻게 될지 생각해보라

실습과제2

24

- 아래 실행 결과가 나오도록 클래스 Rectangle를 정의하라. 생성자는 1개만 선언하라.
- 힌트: 디폴트 매개변수를 이용, 멤버변수는 폭과 높이, 멤버함수 show 는 정보출력

```
// 클래스 선언 및 구현
int main() {
    Rectangle rect0;
    rect0.show();
    Rectangle rect1(10);
    rect1.show();
    Rectangle rect2(10, 20);
    rect2.show();
    return 0;
}
```

사각형 폭은 1 높이는 1
사각형 폭은 10 높이는 1
사각형 폭은 10 높이는 20

실습과제3

25

- 아래 실행결과가 나오도록 클래스 Point3D를 정의하라. 생성자는 1개만 선언하라.
- 힌트: 멤버변수는 x, y, z축 좌표, 생성자는 디폴트 매개변수를 사용, 멤버함수 show는 정보출력

```
// 클래스 선언 및 구현
int main() {
    Point3D p0;
    p0.show();
    Point3D p1(1);
    p1.show();
    Point3D p2(1, 2);
    p2.show();
    Point3D p3(1, 2, 3);
    p3.show();
    return 0;
}
```

3차원 점의 좌표는 (0,0,0)
3차원 점의 좌표는 (1,0,0)
3차원 점의 좌표는 (1,2,0)
3차원 점의 좌표는 (1,2,3)

과제제출방법

26

- 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
```