

# cpp\_07\_함수중복과 static 멤버

## ☐ 개념 확인 학습

1. 다음 질문에 O, X로 답하세요.

- A. 함수 중복은 매개변수의 타입이 다르거나 개수가 달라야 한다. ( )
- B. 함수 중복은 리턴 타입만 달라도 가능하다. ( )
- C. 소멸자는 매개변수를 가지지 않으므로 중복할 수 없다. ( )
- D. 디폴트 매개변수 선언은 위치와 상관없다. ( )
- E. 함수 중복 시 참조 매개변수로 인하여 모호성이 발생할 수 있다. ( )
- F. 디폴트 매개변수를 사용하여 함수 중복을 간소화할 수 있다. ( )
- G. 중복 함수와 디폴트 매개변수를 가진 함수를 함께 사용할 수 있다. ( )
- H. static 멤버 함수에서는 non-static 멤버를 사용할 수 없다. ( )
- I. static 멤버 함수에서 this는 선택적으로 사용할 수 있다. ( )
- J. static 멤버는 "클래스이름." 으로 접근할 수 있다. ( )
- K. static 멤버는 객체 포인터로 접근할 수 있다. ( )
- L. static 멤버 선언 시 inline으로 선언하면 전역 공간에 선언할 필요가 없다. ( )

2. 다음과 같은 디폴트 매개변수를 갖는 함수를 호출할 때 컴파일 오류가 발생하는 문장을 모두 고르세요.

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

- A. msg('a', 20);
- B. msg();
- C. msg('b', 30, "default");
- D. msg("C++");
- E. msg('b', "default");

3. 아래의 함수를 실행한 결과를 출력 할 수 있는 문장을 모두 제시하세요.

```
int sum(int a, int b, int c=30) {  
    return a + b + c;  
}
```

4. 아래의 함수를 실행한 결과를 출력 할 수 있는 문장을 모두 제시하세요.

```
double sum(double a=3.4, double b=6.5) {  
    return a + b;  
}
```

5. 다음과 같이 선언된 함수에서 오류가 발생하는 원인을 제시하고 수정하세요.

```
int calc(int a, int b=5, int c, int d=0)
```

6. 다음 함수 호출이 가능하도록 디폴트 매개변수를 갖는 함수를 제시하세요.

```
int main() {  
    cout << "sum = " << sum(2, 6, 5) << endl;  
    cout << "sum = " << sum(3.5, 5.6) << endl;  
    cout << "sum = " << sum(5, 12) << endl;  
}
```

7. 다음 두 개의 중복 함수를 디폴트 매개변수를 갖는 하나의 함수로 작성하세요.

```
void fillLine(int n) {  
    for(int i=0; i<n; i++) cout << '※';  
    cout << endl;  
}  
void fillLine(int n, char c) {  
    for(int i=0; i<n; i++) cout << c;  
    cout << endl;  
}
```

8. Test 클래스를 사용한 다음 문제의 해법을 제시하세요.

```
class Test {  
    public:  
        static int cnt;  
        static void print();  
};
```

A. static 멤버 함수를 호출하는 방법을 모두 제시하세요.

B. static 멤버 변수에 대한 외부 선언 문장을 제시하세요.

C. static 멤버 변수에 대한 외부 선언이 필요 없도록 선언하는 문장을 제시하세요.

9. 실행결과를 참고하여 main() 함수에서 Person 클래스 타입의 객체를 생성할 때 모든 객체를 하나의 생성자로 처리 할 수 있도록 프로그램을 완성하세요.

<pre>class Person {     int id;     double weight;     string name; public:     //     //Person 생성자 작성     //     void show() { cout &lt;&lt; id &lt;&lt; ' ' &lt;&lt; weight &lt;&lt; ' ' &lt;&lt; name &lt;&lt; endl; } };  int main() {     Person grace, ashley(2, "Ashley"), helen(3, "Helen", 32.5);     grace.show();     ashley.show();     helen.show(); }</pre>	<pre>PS C:\yanges\lecture\lecture_src\cpp&gt; g++ cpptest.cpp PS C:\yanges\lecture\lecture_src\cpp&gt; ./a 1 20.5 Grace 2 20.5 Ashley 3 32.5 Helen</pre>
---	--

---

## □ 응용 프로그래밍

---

10. 함수 big()을 호출하는 경우는 다음과 같습니다. 디폴트 매개변수를 가진 하나의 함수로 big()을 작성하세요.

```
int main() {
    int x = big(3, 5); //3과 5중 큰 값 5는 최대값 100보다 작으므로, 5 리턴
    int y = big(300, 60); //300과 60중 큰 값 300이 최대값 100보다 크므로, 100 리턴
    int z = big(30, 60, 50); // 30과 60 중 큰 값 60이 최대값 50보다 크므로, 50 리턴
    cout << x << ' ' << y << ' ' << z << endl;
}
```

11. 디폴트 매개변수를 갖는 하나의 add() 함수를 작성하고 프로그램을 완성 하세요.

```
int main() {
    int a[] = {1, 2, 3, 4, 5};
    int b[] = {6, 7, 8, 9, 10};

    int c = add(a, sizeof(a)/sizeof(int)); //배열 a의 정수를 size 인덱스까지 모두 더한 값 리턴
    int d = add(a, sizeof(a)/sizeof(int), b); //배열 a, b의 정수를 size 인덱스까지 모두 더한 값 리턴

    cout << c << endl; // 15 출력
    cout << d << endl; // 55 출력
}
```

12. ArrayUtil 클래스의 static 함수인 concat()은 두 개의 배열을 연결한 새로운 배열을 동적 생성하여 리턴합니다. ArrayUtil 클래스의 static 함수인 remove()는 앞 배열에서 뒤 배열의 있는 숫자를 모두 삭제한 새로운 배열을 동적 생성하여 리턴합니다. 아래에 제시된 ArrayUtil 클래스와 실행화면을 참고하여 프로그램을 완성 하세요.

```
class ArrayUtil {
public:
    //s1과 s2를 연결한 새로운 배열을 동적 생성하여 리턴
    static int* concat(int s1[], int size1, int s2[], int size2, int&retSize);

    //s1에서 s2에 있는 숫자를 모두 삭제한 새로운 배열을 동적 생성하여 리턴
    static int* remove(int s1[], int size1, int s2[], int size2, int&retSize);
};

//concat() 구현합니다.

//remove() 구현합니다.

int main() {
    int xcount = 0, ycount = 0;
    int *x, *y;

    //x배열의 원소의 개수(1~5)를 입력 받는다.
    //입력 받은 수만큼 x배열을 동적 생성한다.
    //정수를 입력 받아 x배열에 저장한다.

    //y배열의 원소의 개수(1~5)를 입력 받는다.
    //입력 받은 수만큼 y배열을 동적 생성한다.
    //정수를 입력 받아 y배열에 저장한다.

    int retsize = 0;

    //concat() 함수를 호출한다.

    cout << "Wnx배열과 y배열을 연결한 배열은 ";
    for (int i = 0; i < retsize; i++) cout << conarr[i] << ' ';
    cout << endl;

    //remove() 함수를 호출한다.

    cout << "Wnx배열에서 y배열의 원소를 삭제한 결과, x배열의 개수는 " << retsize << "개, ";
    for (int i = 0; i < retsize; i++) cout << remarr[i] << ' ';
    cout << endl;

    //동적 할당된 모든 변수를 해제 한다.
}
```

```
PS C:\yanges\lecture\lecture_src\cpp> ./a
x배열의 원소(1~5)는 몇 개 ? 4
x배열에 저장할 정수 입력 : 3 4 5 6

y배열의 원소(1~5)는 몇 개 ? 5
y배열에 저장할 정수 입력 : 5 6 7 8 9

x배열과 y배열을 연결한 배열은 3 4 5 6 5 6 7 8 9

x배열에서 y배열의 원소를 삭제한 결과, x배열의 개수는 2개, 3 4
PS C:\yanges\lecture\lecture_src\cpp>
PS C:\yanges\lecture\lecture_src\cpp> ./a
x배열의 원소(1~5)는 몇 개 ? 3
x배열에 저장할 정수 입력 : 1 2 3

y배열의 원소(1~5)는 몇 개 ? 3
y배열에 저장할 정수 입력 : 1 2 3

x배열과 y배열을 연결한 배열은 1 2 3 1 2 3

x배열에서 y배열의 원소를 삭제한 결과, x배열의 개수는 0개,
```