

# 심화프로그래밍

실습강의 7주차

# 실습강의 소개

## ● 실습 진행 방법

- 간단한 이론 복습 및 해당주차 실습과제 설명
- 실습 후 보고서와 소스코드를 압축하여 **수요일 자정(23:59)까지**  
꼭!! 이클래스 제출(**이메일 제출 불가, 반드시 이클래스를 통해 제출**)
- 실습과제 제출기한 엄수(제출기한 이후로는 0점 처리)

## ● Q & A

- 이클래스 및 실습조교 이메일을 통해 질의응답
- **이메일 제목 : [심화프로그래밍\_홍길동] \*본인 과목명과 성명 꼭 작성!!**
- 실습조교 메일 주소 : [0hae@dgu.ac.kr](mailto:0hae@dgu.ac.kr), [wundermilch@dgu.ac.kr](mailto:wundermilch@dgu.ac.kr)

# 실습 보고서 작성 방법 [1/2]

## ● 실습 보고서

- 문제 분석: 실습 문제에 대한 요구 사항 파악, 해결 방법 등 기술
- 프로그램 설계 및 알고리즘
  - 해결 방법에 따라 프로그램 설계 및 알고리즘 등 기술
  - e.g.) 문제 해결 과정 및 핵심 알고리즘 기술
- 소스코드 및 주석
  - 소스코드와 그에 해당하는 주석 첨부
  - 각각의 함수가 수행하는 작업, 매개변수, 반환 값 등을 명시
  - 소스코드 전체 첨부 (소스코드 화면 캡처X, 소스코드는 복사/붙여넣기로 첨부)
- 결과 및 결과 분석
  - 결과 화면을 캡처 하여 첨부, 해당 결과가 도출된 이유와 타당성 분석
- 소감
  - 실습 문제를 통해 습득할 수 있었던 지식, 느낀 점 등을 기술

# 실습 보고서 작성 방법 [2/2]

## ● 제출 방법

- 보고서, 소스코드를 1개의 파일로 압축하여 e-class “과제” 메뉴를 통해 제출
  - “이름학번실습주차.zip” 형태로 제출(e.g. :김동국19919876실습7.zip)
  - 파일명에 공백, 특수 문자 등 사용 금지

## ● 유의 사항

- 보고서의 표지에는 학과, 학번, 이름, 담당 교수님, 제출일자 반드시 작성
- 정해진 기한내 제출
  - 기한 넘기면 **0점** 처리
  - 이클래스가 과제 제출 마지막 날 오류로 동작하지 않을 수 있으므로, 최소 1~2일전에 제출
  - **과제 제출 당일 이클래스 오류로 인한 미제출은 불인정**
- 소스코드, 보고서를 자신이 작성하지 않은 경우 **실습 전체 점수 0점 처리**
- **Visual Studio 2019 또는 Sharstra 웹 IDE 기반 학습 프로그램 사용하여 실습 진행**

# 함수 중복(Function Overloading)[1/2]

## ● 함수 중복

- 동일한 이름의 함수가 공존하는 것을 의미함
  - 같은 이름으로 여러가지의 형태를 정의해놓고, 각각의 기능을 수행하도록 만드는 것
- 함수 이름을 구분하여 기억할 필요가 없고, 함수 호출을 잘못하는 실수를 줄일 수 있음

## ● 함수 중복이 가능한 범위

- 보통 함수들 사이
- 클래스의 멤버 함수들 사이
- 상속 관계에 있는 기본 클래스와 파생 클래스의 멤버 함수들 사이

## ● 함수 중복이 가능한 조건

- 중복된 함수들의 이름 동일해야함
- 중복된 함수들의 매개 변수 타입이 다르거나 개수가 달라야 함
- 리턴 타입은 함수 중복과 무관함
  - 매개 변수의 자료형과 개수가 모두 똑같은 동시에 반환형만 다른 경우 함수 중복 실패

```
1. int sum(int a, int b){ return a + b; }  
2. double sum(int a, int b){ return (double)(a + b); }
```

# 함수 중복(Function Overloading)[2/2]

## ● 생성자 함수 중복

- 객체를 생성할 때 매개 변수를 통해 다양한 형태로 초깃값을 전달하기 위해 생성자 함수를 중복 선언
- C++ 표준의 string 클래스는 다음과 같이 다양한 생성자를 제공하며 다양한 초깃값으로 string 객체를 생성할 수 있도록 한다.

```
1.  class String {
2.      ...
3.  public:
4.      string(); // 빈 문자열을 가진 스트링 객체 생성
5.      string(char* s); // null('\0')값으로 끝나는 C-스트링 문자 배열을 스트링 객체로 생성
6.      string(string& str); // str을 복사한 새로운 스트링 객체 생성
7.  };

8.  int main() {
9.      string str; // string() 호출
10.     string address("서울 중구"); // string(char* s) 호출
11.     string copyAddress(address); // string(string& str) 호출; address 문자열을
                                   복사한 별도의 copyAddress 생성
12. }
```

## ● 소멸자 함수 중복

- 소멸자는 매개 변수를 가지지 않으므로, 한 클래스 당 오직 하나만 존재하므로 함수 중복이 불가능하다.

# 디폴트 매개 변수(Default Parameter)

- 디폴트 매개 변수/기본 매개 변수

- 기본적으로 설정되어 있는 값
- 사용자가 이 매개 변수의 값을 제공하지 않으면 기본값(default value)이 사용된다. 반대로 매개 변수의 값을 제공하면 사용자 제공 값이 기본값 대신 사용된다.

- 디폴트 매개 변수 선언

- '매개 변수 = 디폴트 값' 형태로 선언 됨.

- 디폴트 매개 변수 조건과 규칙

- 디폴트 매개 변수는 모두 오른쪽 끝 쪽에 몰려 선언되어야 한다.

```
1. // int sum(int a = 0, int b, int c) { return (a + b + c); };  
2. // int sum(int a, int b = 0, int c) { return (a + b + c); }  
3. // 위 두 함수가 호출되면 컴파일 오류가 남
```

- 디폴트 매개 변수의 장점

- 디폴트 매개 변수로 인해서 중복 생성자들을 디폴트 매개 변수를 가진 하나의 생성자로 간소화할 수 있다.

# 디폴트 매개 변수

- 디폴트 매개 변수(Default Parameter)

- 함수 호출 시 인자를 넣지 않으면, “미리 정해진 인자 값”이 들어감
- 반대로, 인자로 값을 넣게 되면 “미리 정해진 인자 값이”이 아니라 **지금 넣은 인자**가 들어감

```
1. void printValues(int x, int y = 10) {  
2.     cout << "x: " << x << '\n';  
3.     cout << "y: " << y << '\n';  
4. }  
5.  
6. int main() {  
7.     printValues(1);    // y will use default parameter of 10  
8.     printValues(3, 4); // y will use user-supplied value 4  
9. }
```

```
x: 1  
y: 10  
x: 3  
y: 4
```

- Default Parameter를 사용하는 몇 가지 함수 원형

```
1. void openLogFile(string filename="default.log");  
2. int rollDie(int sides=6);  
3. void printStringInColor(string str, Color color=COLOR_BLACK);  
4. // Color is an enum
```



# 디폴트 매개 변수의 오류 발생 예시

- 모든 디폴트 매개 변수는 오른쪽부터 지정해야함

```
1. void printValue(int x = 10, int y); // error
```

- 전방 선언이나 함수 정의 둘 중 하나만 선언할 수 있음

- 헤더 파일을 사용하는 경우, 전방 선언 코드를 헤더로 옮겨서 사용하는 것이 가장 좋음

```
1. void printValues(int x, int y=10);  
2.  
3. void printValues(int x, int y=10) { // error  
4.     std::cout << "x: " << x << '\n';  
5.     std::cout << "y: " << y << '\n';  
6. }
```

- 다음 예제는 모두 사용 불가능함

- 디폴트 값을 앞에서 지정하거나, 중간에 비어있는 게 있다면 어느 인자를 생략할지 알 수 없음

```
1. void func5(int id = 1, bool isStudent, int age, string name) {} // 왼쪽  
2. void func6(int id, bool isStudent = true, int age, string name) {} // 가운데  
3. void func7(int id, bool isStudent, int age = 44, string name) {} // 가운데  
4. void func8(int id = 3, bool isStudent, int age, string name = "BlockDMask") {}  
5. void func9(int id, bool isStudent = true, int age, string name = "BlockDMask") {}  
6. void func10(int id, bool isStudent, int age = 11, string name) {}
```

- 동국대학교  
dongguk university

5. 함수 중복이 성공적인지 판별은 언제 이루어지는가?

- ① 전처리기(preprocessor)에 의해 컴파일 직전
- ② 컴파일러에 의해 컴파일 시
- ③ 실행 시간 중 함수가 호출되기 직전
- ④ 실행 시간 중 함수 중복을 판별하는 특정 시간에

6. 중복된 함수와 이를 호출하는 호출문의 연결(바인딩)은 언제 이루어지는가?

- ① 전처리기(preprocessor)에 의해 컴파일 직전
- ② 컴파일러에 의해 컴파일 시
- ③ 실행 시간 중 함수가 호출되기 직전
- ④ 실행 시간 중 함수 중복을 판별하는 특정 시간에

# 연습문제(6장)

7. 다음과 같은 디폴트 매개 변수를 가진 함수가 있을 때, 적절치 못한 함수 호출은?

```
int add(int a=10, double b=10.0);
```

- ① int x = add();
- ② int x = add(20);
- ③ add(20, 3.14);
- ④ int x = add(20, 3.14, 10);

8. 다음과 같은 디폴트 매개 변수를 가진 함수가 있을 때, 적절치 못한 함수 호출은?

```
int add(int a, double b=10.0);
```

- ① int x = add(10);
- ② int x = add(10, 3.14, 10);
- ③ add(10, 3.14);
- ④ int x = add(10, 3.14);

9. 다음 함수의 선언이 잘못된 것은?

- ① void f1(int a, int b, int c);
- ② int f2(int a, int b, int c=0);
- ③ void f3(int a, int b=0, int c);
- ④ int f4(int a=0, int b=0, int c=0);

10. 중복 함수의 호출시 어떤 함수를 호출하는지 판단할 수 없는 모호성의 가능성이 있는 것은?

- ① int add(int a); int add(int a, int &b);
- ② int add(int a, int b); int add(int a, int &b);
- ③ int add(int a, int b); int add(int a, int\* b);
- ④ int add(int a, int\* b); int add(int a, int &b);

# 실습과제(6장)

1. add() 함수를 호출하는 main() 함수는 다음과 같다.

```
int main () {  
    int a[] = {1, 2, 3, 4, 5};  
    int b[] = {6, 7, 8, 9, 10};  
    int c = add(a, 5); // 배열 a의 정수를 모두 더한 값 리턴  
    int d= add(a, 5, b); // 배열 a와 b의 정수를 모두 더한 값 리턴  
    cout << c<< endl; // 15 출력  
    cout << d << endl; // 55 출력  
}
```

- (1) add() 함수를 중복 작성하고 프로그램을 완성하라.
- (2) 디폴트 매개 변수를 가진 하나의 add() 함수를 작성하고 프로그램을 완성하라.

# 실습과제(6장)

2. Person 클래스의 객체를 생성하는 main() 함수는 다음과 같다.

```
class Person {  
    int id;  
    double weight;  
    string name;  
public:  
    void show() { cout << id << ' ' << weight << ' ' << name << endl; }  
};  
  
int main () {  
    Person grace, ashley(2, "Ashley"), helen(3, "Helen", 32.5);  
    grace.show();  
    ashley.show();  
    helen.show();  
}
```

```
1 20.5 Grace  
2 20.5 Ashley  
3 32.5 Helen
```

- (1) 생성자를 중복 작성하고 프로그램을 완성하라.
- (2) 디폴트 매개 변수를 가진 하나의 생성자를 작성하고 프로그램을 완성하라.

# 실습과제(6장)

3. 함수 big()을 호출하는 경우는 다음과 같다.

```
int main() {  
    int x = big(3, 5); // 3과 5중 큰 값 5는 최대값 100보다 작으므로, 5 리턴  
    int y = big(300, 60); // 300과 60 중 큰 값 300이 최대값 1100보다 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;  
}
```

5 100 50

- (1) big() 함수를 2개 중복하여 작성하고 프로그램을 완성하라.
- (2) 디폴트 매개 변수를 가진 하나의 함수로 big()을 작성하고 프로그램을 완성하라.

# 실습과제(6장)

4. 다음 클래스에 중복된 생성자를 디폴트 매개 변수를 가진 하나의 생성자로 작성하고 테스트 프로그램을 작성하라.

```
class MyVector{
    int *mem;
    int size;
public:
    MyVector ();
    MyVector (int n, int val);
    ~MyVector() { delete [] mem; }
};

MyVector: : MyVector() {
    mem = new int [100];
    size = 100;
    for(int i=0; i < size; i++) mem[i] = 0;
}

MyVector: : MyVector(int n, int val) {
    mem = new int [n];
    size = n;
    for(int i=0; i < size; i++) mem[i] = val;
}
```