$$\mathbb{C}++$$

박성기

Day01 - Contents

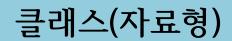
- 1. 객체지향특징 C++
- 2. 객체를 이용한 입출력 iostream cin, cout
- 3. 예제 코드

Day01 - 1. 객체지향 특징

객체지향

- 클래스: 데이터(속성)와 기능(함수)을 변수와 메소드 정의
- 객체: 클래스의 정의를 기반으로 실제 메모리에 할당된 물리적 실체
- 추상화: 객체들의 공통 속성과 기능(함수)을 클래스로 만드는 방법
- 캡슐화: 속성과 기능(함수)을 목적에 맞게 묶어서 독립된 단위로 구성
- 상속성: 상위 객체의 속성과 기능을 물려받는 것
- 다형성: 기능(함수)의 재정의

프로그램의 구조화로 코드의 재사용, 호환성, 디버깅, 유지보수가 좋다

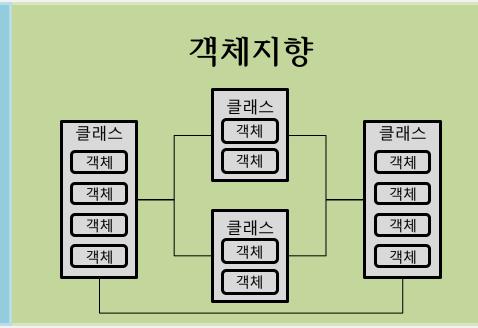


객체 1 (변수)

데이터 연산 정수, +, -, /, 실수, *, %, = 문자,…

객체 2 (변수)

데이터	연산
정수,	+, -, /,
실수,	*, %, =
문자,…	•••



Day01 - 1. C++

C++의 특징

- C++은 다른 C와 같은 방법으로 컴파일된다.
 소스(c++) 전처리 컴파일 어셈블리 링킹 -> 기계어
- *.cpp 확장자
- 코드 중간에 변수 선언 가능
- C와 호환성이 좋다 (헤더파일 불러올 수 있다)
- 객체지향 (캡슐화, 상속, 다형성)
- 오버로딩(함수 중복 사용)
- 디폴트 매개 변수 가능
- 참조와 참조 변수
- 참조에 의한 호출
- new/delete 동적할당
- 연산자 재정의가 가능
- class (클래스 = 표현할 정보의 종류 & 동사적 기능)

Day01 - 2. 객체를 이용한 입출력

#include <iostream>

- 객체를 이용한 표준 입출력
- 입력 cin >> (변수)
- 출력 cout << "Hello";
- 출력 clog << "World";
- 출력 cerr << "C++";
- clog, cerr 의 스트림을 변경하면 출력을 다른 파일로 할 수 있다.
- cout, clog, cerr의 default 스트림은 콘솔창
- 'endl' 다음 줄로 이동(개행) ('\n'과 같은 기능)
- using namespace std;

#include <iomanip>

- left : 왼쪽 정렬, 한번 설정 후 계속 적용된다.
- right : 오른쪽 정렬, 한번 설정 후 계속 적용된다. (디폴트는 오른쪽 정렬)
- setw(변수 or 상수) : 표시 필드 폭, setw() 바로 뒤 한 개의 데이터에 적용
- setprecision(변수 or 상수) : 부동 소수점 전체 자릿수, 이후 모두 적용
- setbase(8 or 10 or 16): 8 or 10 or 16 진법 출력
- setfill(문자상수): 공백에 (문자) 출력

Day01 - 2. 객체를 이용한 입출력

#include <iostream>

- cin >> (변수) 입력 버퍼의 공백(space bar, tap, enter) 전까지 값들을 변수로 대입 \n (Enter key)를 버퍼에 남김 변수로 입력을 실패 할 경우 변수 값 0 초기화 , failbit = 1 [while(getchar()!='\n'); 기능] void cinclear() { // 입력 버퍼 비우는 함수 if (cin.fail()) { // 입력 실패 체크 함수 cin.clear(); // 입력 초기화 while (cin.get()!='\n'); // 버퍼 비우기 (비워진 경우 Enter 입력) } while (cin.get()!='\n'); // 버퍼 비우기 (비워진 경우 Enter 입력)

 cin.getline(배열 이름, 배열 칸수, 종결문자[생략 가능]) 띄어쓰기(공백) 포함 \n (Enter key)까지 변수로 전달. \n 키를 \0로 변경해 저장 (문자열 끝을 알리는 기호) (종결문자 전까지 입력 받음)

 $Hello(\n : Enter) \rightarrow Hello(0)$

Day01 - 2. 객체를 이용한 입출력

표준입출력(cin, cout)

- 입출력을 위한 클래스와 객체,변수 등이 선언된 헤더파일 <iostream>
- 객체를 이용한 표준 입출력 (cin, cout)
- 'endl' 다음 줄로 이동(개행) '\n'과 동일하다
- 데이터의 <mark>타입을 지정하지 않아도</mark> 자체적으로 입출력 (%d, %f, %c)
- 데이터가 여러 개인 경우 <<, >> 연산자로 구분 cin >> 문자열 입력 시 <mark>공백이 포함된 문자열을 받을 수 없다</mark> (공백을 입력의 끝으로 처리)
- 자료형과 다르게 출력 하고 싶다면 형변환 필요
 char 문자;
 cout << 문자 << (int)문자 << endl; // 정수 출력
 int 정수;
 cout << 정수 << (char)정수 << endl; // 문자 출력
 char* 주소;
 cout << 주소 << (void*)주소 << endl; // 주소 출력
- 실수는 유효 자릿수(6~7자리) 출력, 자리수가 커지면 지수표기로 출력
- ""(큰 따옴표)는 문자열을 의미, ' '(작은 따옴표) 문자 상수 (ASCII Code)
- char string[20] = "Hello World!"; 각각의 문자를 배열의 1칸씩 저장, 맨 뒤에 널 문자('\0')를 자동 추가 cout << string << endl; 저장된 문자열 널 문자('\0)까지 출력
- (const) char* pstring = "Hello World!";
 문자열이 저장된 시작 주소를 pstring에 저장
 cout << pstring << endl; 시작 주소부터 널 문자('\0)까지 출력

Day01 - 2. ASCII코드

10진수	16진수	문자
0	0x00	NUL
1	0x01	SOH
2	0x02	STX
3	0x03	ETX
4	0x04	ЕОТ
5	0x05	ENQ
6	0x06	ACK
7	0x07	BEL
8	0x08	BS
9	0x09	TAB
10	0x0A	LF
11	0x0B	VT
12	0x0C	FF
13	0x0D	CR
14	0x0E	SO
15	0x0F	SI
16	0x10	DLE
17	0x11	DC1
18	0x12	DC2
19	0x13	DC3
20	0x14	DC4
21	0x15	NAK
22	0x16	SYN

10713	10717	п -1
10진수	16진수	문자
23	0x17	ETB
24	0x18	CAN
25	0x19	EM
26	0x1A	SUB
27	0x1B	ESC
28	0x1C	FS
29	0x1D	GS
30	0x1E	RS
31	0x1F	US
32	0x20	공백
33	0x21	!
34	0x22	"
35	0x23	#
36	0x24	\$
37	0x25	%
38	0x26	&
39	0x27	'
40	0x28	(
41	0x29)
42	0x2A	*
43	0x2B	+
44	0x2C	,
45	0x2D	-

10진수	16진수	문자
46	0x2E	
47	0x2F	/
48	0x30	0
49	0x31	1
50	0x32	2
51	0x33	3
52	0x34	4
53	0x35	5
54	0x36	6
55	0x37	7
56	0x38	8
57	0x39	9
58	0x3A	:
59	0x3B	;
60	0x3C	<
61	0x3D	=
62	0x3E	>
63	0x3F	?
64	0x40	@
65	0x41	Α
66	0x42	В
67	0x43	С
68	0x44	D

근자	10진수	16진수	문자
	69	0x45	E
/	70	0x46	F
0	71	0x47	G
1	72	0x48	Н
2	73	0x49	I
3	74	0x4A	J
4	75	0x4B	K
5	76	0x4C	L
6	77	0x4D	M
7	78	0x4E	N
8	79	0x4F	O
9	80	0x50	P
;	81	0x51	Q
;	82	0x52	R
<	83	0x53	S
=	84	0x54	T
>	85	0x55	U
?	86	0x56	V
@	87	0x57	W
Α	88	0x58	X
В	89	0x59	Y
С	90	0x5A	Z
D	91	0x5B	ſ

10진수	16진수	문자
92	0x5C	\
93	0x5D]
94	0x5E	^
95	0x5F	_
96	0x60	`
97	0x61	a
98	0x62	b
99	0x63	С
100	0x64	d
101	0x65	е
102	0x66	f
103	0x67	g
104	0x68	h
105	0x69	i
106	0x6A	j
107	0x6B	k
108	0x6C	1
109	0x6D	m
110	0x6E	n
111	0x6F	O
112	0x70	p
113	0x71	q
114	0x72	r

10진수	16진수	문자
115	0x73	S
116	0x74	t
117	0x75	u
118	0x76	v
119	0x77	w
120	0x78	x
121	0x79	y
122	0x7A	z
123	0x7B	{
124	0x7C	-
125	0x7D	}
126	0x7E	~
127	0x7F	DEL

```
#include <stdio.h>
                                                           #include <iostream>
int main() {
                                                           #include <iomanip>
  int a, b;
                                                           //left, right, setw()
                                                           //setfill(), setbase(), setprecision()
  printf("두 수 입력:");
  scanf_s("%d %d", &a, &b);
                                                           using namespace std;
  printf("입력결과: %d, %d\n", a, b);
                                                           int main() {
  return 0;
                                                            //cout << setfill('0');</pre>
                                                            cout << setbase(10);
                                                            cout << setprecision(4);</pre>
#include <iostream>
                                                            // setprecision 부동 소수점 실수 값의 자릿수
int main() {
                                                            for (int i = 1; i \le 100; i++) {
  std::cout << "두 수 입력:";
                                                             if (i % 2 == 0) {
  int a, b;
                                                               cout << right;
  std::cin >> a >> b;
                                                               cout << setw(6) << i / 3.0;
  std::cout << "입력결과: " << a << ", " << b << std::endl;
                                                               // setw 출력 자리수
  return 0:
                                                               cout << endl;
                                                              else {
                                                               cout << left;
                                                               cout << setw(6) << i / 3.0 << " | ";
                                                            cout << endl;
                                                            return 0;
```

★과제★

C++ 작성하기

- 문자 입력 하면 ASCII 코드 출력하기

입력문자 : A ASCII코드: 65

- 소문자 입력 -> 대문자 변환 대문자 입력 -> 소문자 변환

문자입력 : A A ---> a

문자입력: a a ---> A

```
#include <iostream>
using namespace std;
int main() {
 char ch = 'A'; //문자 상수 'A' = 65
 cout << "입력문자 : " << ch << endl;
 cout << "ASCII코드: " << (int)ch << endl;
#include <iostream>
using namespace std;
int main() {
 char ch;
 cout << "문자입력:";
 cin >> ch;
 if (ch >= 'A' && ch <= 'Z') cout << ch << " ---> " << (char)(ch + 32) << endl;
 else if (ch >= 'a' && ch <= 'z') cout << ch << " ---> " << (char)(ch - 32) << endl;
```

★과제★ C++ 작성하기

- 문자열 입력 받고 검색한 알파벳 수량 출력하기 문자열 저장 할 배열을 선언 cin.getline(배열 이름, 배열 칸 수); // 문자열 입력 함수
- 입력 받은 문자열 모두 소문자 -> 대문자 (or 대문자 -> 소문자) 변환
- 검색 문자 입력:a
- 결과 출력 검색 문자 [A(a)]의 개수는 [4]개 입니다.

```
//문자열에서 문자를 검색해서 개수 출력하시오.
//문자열 상수는 뒤에 '\0'이(NULL) 붙는다. "ASCII CODE" + \0 (숨겨진 값)
#include <iostream>
using namespace std;
int main() {
 char str[255];
 cout << "문자열 입력:";
 cin.getline(str, 255);
 for (int i = 0; str[i] != '\0'; i++) {
  if ('a' <= str[i] && str[i] <= 'z') str[i] = str[i] - 32;
 char ch;
 cout << "검색 문자 입력:";
 cin >> ch;
 if (ch >= 'a' && ch <= 'z') ch -= 32;
 int count = 0;
 for (int i = 0; str[i]!= '\0'; i++) {
  if (str[i] == ch) ++count;
 cout << "\n검색 문자[" << ch << "(" << (char)(ch + 32) << ")]의 개수는[" << count << "]개 입니다.\n";
 return 0;
```

★과제★

C++ 작성하기

[단위변환 프로그램]

- 1. 인치 <-> 센티미터
- 2. 화씨 <-> 섭씨 메뉴를 선택 하세요. : 1



- 1. 인치 -> 센티미터
- 2. 센티미터 -> 인치 메뉴를 선택 하세요. : 1

inch 를 입력 하세요. : 100 100 inch = 254 cm

- 1. 인치 <-> 센티미터
- 2. 화씨 <-> 섭씨

메뉴를 선택 하세요. : 2



- 1. 화씨 -> 섭씨

메뉴를 선택 하세요. : 1

°F 를 입력하세요. : 100 100 °F = 37.7778 °C

```
#include <iostream>
using namespace std;
int main() {
                                                             else if (main_menu == 2) {
 int main_menu, sub_menu;
                                                              cout << endl;
                                                              cout << "1. °F <-> °C" << endl;
 float cel, fah;
                                                               cout << "2. °C <-> °F" << endl;
 float centi, inch;
                                                              cout << "메뉴를 선택하세요.:";
 while (1) {
  cout << "[단위 변환 프로그램]" << endl;
                                                              cin >> sub_menu;
  cout << "1. inch <-> cm" << endl;
                                                              if (sub_menu == 1) {
  cout << "2. °F <-> °C" << endl:
                                                                cout << endl;
  cout << "메뉴를 선택하세요.:";
                                                                cout << " ℉ 를 입력하세요.:";
  cin >> main menu;
                                                                cin >> fah;
  if (main_menu == 1) {
                                                                cel = (5.0 / 9.0) * (fah - 32.0);
                                                                cout << fah << " °F = " << cel << " °C\n";
   cout << endl;
   cout << "1. inch -> cm" << endl;
    cout << "2. cm -> inch" << endl;
                                                               else if (sub_menu == 2) {
    cout << "메뉴를 선택하세요.:";
                                                                cout << endl;
                                                                cout << " ℃ 를 입력하세요.:";
   cin >> sub_menu;
   if (sub_menu == 1) {
                                                                cin >> cel;
     cout << endl;
                                                                fah = (9.0 / 5.0) * cel + 32.0;
     cout << " inch 를 입력하세요.:";
                                                                cout << cel << " °C = " << fah << " °F \n";
     cin >> inch;
     centi = inch *2.54;
     cout << inch << "inch = " << centi << "cm" << endl;
                                                            return 0;
    else if (sub_menu == 2) {
     cout << endl;
     cout << " cm 를 입력하세요.:";
     cin >> centi;
     inch = centi / 2.54;
     cout << centi << " cm = " << inch << "inch" << endl;
```

Day02 - Contents

- 1. Function Overloading
- 2. Default Parameter
- 3. inline
- 4. namespace
- 5. 예제 코드

Day02 - 1. Function Overloading

Function Overloading

C언어

함수의 이름만을 가지고 함수를 호출하기 때문에 같은 이름의 함수 사용이 불가능

C++

함수의 이름과 매개변수(인수)의 정보를 함께 가지고 호출하기 때문에 함수이름은 같아도 매개변수(인수)의 자료 형이나 개수가 다르면 함수 오버로딩이라고 한다. (리턴 타입은 오버로딩과 무관하다)

다형성(polymorphism) 같은 이름을 가진 함수를 다양한 형태로 구현

같은 기능의 함수를 동일한 이름으로 구현하면 사용자가 이해하기 쉽고 일관된 함수를 만들 수 있다.

```
char max_char (char a, char b);
int max_int (int a, int b);
double max_double (double a, double b);
```

int max (char a, char b);
int max (int a, int b);
double max (double a, double b);

```
C++ 코드 작성
Function Overloading [함수 오버로딩]
함수이름, 매개변수(인수)의 자료 형, 개수로 함수를 구분
( 위 기준에서 차이가 있다면 다른 함수로 인식 )
char MAX (char a, char b); // 두 문자 중 큰 값 반환 함수
int MAX (int a, int b); // 두 정수 중 큰 값 반환 함수
double MAX (double a, double b); //두 실수 중 큰 값 반환 함수
int SUM (char a, char b); // 두 문자의 합 반환 함수
int SUM (int a, int b); // 두 정수의 합 반환 함수
double SUM (double a, double b); // 두 실수의 합 반환 함수
void call (int a) { cout << "인수 1개 호출\n"; }
void call (int a, int b) { cout << "인수 2개 호출\n"; }
동일한 이름으로 함수 작성, 사용 확인하기
```

```
#include <iostream>
using namespace std;
int SUM(char a, char b) { return a + b; }
double SUM(double a, double b) { return a + b; }
int SUM(int a, int b) { return a + b; }
int MAX(int a, int b) { return (a > b) ? a : b; }
char MAX(char a, char b) { return (a > b) ? a : b; }
double MAX(double a. double b) { return (a > b) ? a : b; }
void call(int a) { cout << "인수 1개 호출\n"; }
void call(int a, int b) { cout << "인수 2개 호출\n"; }
int main() {
 char char_a = 'A', char_b = 'B';
 int int a = 50. int b = 30;
 double double a = 12.3456, double b = 45.6789;
 cout << "문자 비교:" << MAX(char_a, char_b) << endl;
 cout << "정수 비교:" << MAX(int_a, int_b) << endl;
 cout << "실수 비교:" << MAX(double_a, double_b) << endl;
 cout << "문자 합계:" << SUM(char_a, char_b) << endl;
 cout << "정수 합계:" << SUM(int_a, int_b) << endl;
 cout << "실수 합계:" << SUM(double_a, double_b) << endl;
 call(1);
 call(2, 3);
 return 0;
```

Day02 - 1. Function Overloading

Function Overloading

Overloading 함수 호출 순서

- (1) Exactly Matching 전달하는 매개변수와 정확하게 일치하는 타입이 있는지 찾는다.
- (2) Promotion
 데이터의 손실이 없는 방향으로의 변환이 가능한 함수가 있는지 찾는다.
 int형 -> double형으로 전달 시 데이터의 손실 없어 double형 찾아 호출
- (3) Standard Conversion 표준 타입들 간에는 <mark>암시적인 형 변환</mark>이 발생한다. double형 -> int형으로 전달 시 데이터의 손실이 있지만 int형 찾아 호출
- (4) User Define Conversion Operator 사용자가 정의한 변환 연산자가 있다면 이것을 사용해서 변환을 한 후, 함수를 찾는다.
- (5) Variable Argument Function
 '…' 가변 파라미터를('…') 매개변수로 갖는 함수는 파라미터의 타입이나 개수에 상관없이 호출된다.

square함수 오버로딩

- 1. 정수의 제곱을 구해 리턴 하는 함수 int square(int a);
- 2. 실수의 제곱을 구해 리턴 하는 함수 double square(double a);

Date함수 오버로딩

- 1. 년, 월, 일을 인수로 날짜를 출력하는 함수 (날짜 구분 기호는 '/') void date(int year, int month, int day); //date(2022, 01, 01);
- 2. 년, 월, 일과 구분 기호를 인수로 날짜를 출력하는 함수 void date(int year, int month, int day, char separate); // date(2022, 01, 01, '#');
- 3. 문자열을 인수로 날짜를 출력하는 함수 void date(const char* date); // date("2022/01/01");

getTotal함수 오버로딩

- 1. int getTotal(): 1 ~ 100 합계를 구해 리턴
- 2. int getTotal(int x): x ~ 100 합계를 구해 리턴
- 3. int getTotal(int x, int y): x ~ y 합계를 구해 리턴

```
//square함수 오버로딩
#include <iostream>
using namespace std;
int square(int a) {
 return a * a;
double square(double a) {
 return a * a;
int main() {
 int su;
 cout << "정수 입력:";
 cin >> su;
 cout << "정수 제곱값: " << square(su) << endl;
 double f;
 cout << "실수 입력:";
 cin >> f;
 cout << "실수 제곱값 : " << square(f) << endl;
 return 0;
```

```
// Date함수 오버로딩
#include <iostream>
using namespace std;
void date(int year, int month, int day) {
 cout << year << "년/" << month << "월/" << day << "일" << endl;
void date(int year, int month, int day, char separate) {
 cout << year << "년" << separate << month << "월" << separate << day << "일" << endl;
void date(const char* date) {
 cout << date << endl;
int main() {
 date(2022, 01, 01);
 date(2022, 01, 01, '#');
 date("2022/01/01");
 return 0;
```

```
//getTotal함수 오버로딩
#include <iostream>
using namespace std;
int getTotal() {
 int sum = 0;
 for (int i = 1; i \le 100; i++) sum += i;
 return sum;
int getTotal(int x) {
 int sum = 0;
 for (int i = x; i \le 100; i++) sum += i;
 return sum;
int getTotal(int x, int y) {
 int sum = 0;
 for (int i = x; i <= y; i++) sum += i;
 return sum;
int main() {
 cout << getTotal() << endl;</pre>
 cout << getTotal(50) << endl;</pre>
 cout << getTotal(50, 500) << endl;
 return 0;
```

Day02 - 2. Default Parameter

Default Parameter

매개변수에 기본 값이 있는 함수 함수 호출 시 전달되는 값이 있으면 전달된 값으로 대체 전달되는 값이 없으면 자신의 기본 값을 사용

다형성(polymorphism) 한 개의 함수가 다양한 경우에 호출 할 수 있다. 매개변수가 없어도 기본 값으로 호출, 수행 가능하다.

특징

- 매개변수의 디폴트 값은 오른쪽에서부터 차례대로 할당
- 함수의 선언과 정의를 나눠 쓸 경우 디폴트 값은 선언 부에만 쓴다.

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

```
int sum(int a, int b, int c=0, int d=0, int e=0) {
  return a + b + c + d + e;
}
```

Day02 - 2. Default Parameter

Default Parameter

Function Overloading에서 Default Parameter 사용을 주의해야 한다. 함수의 이름과 매개변수가 구분되지 않으면 overloading 불가

```
#include <iostream>
using namespace std;
void call(int a) {
 cout << "인수 1개 호출\n";
void call(int a, int b=0) {
 cout << "인수 2개 호출\n";
int main() {
 call(1);
 call(2, 3);
 return 0;
```

Default Parameter

1. 등비수열의 합을 구하는 함수 작성하기

double geometric_sqc_sum(int a , int r, int n);

a: 첫째항 (기본값 1) r: 공비 (기본값 2) n: 항수 (기본값 5)

$$S_n = \frac{a \cdot (r^n - 1)}{r - 1}$$

main() 호출
geometric_sqc_sum();
geometric_sqc_sum(a);
geometric_sqc_sum(a, r);
geometric_sqc_sum(a, r, n);

```
//등비수열의 합을 구하는 함수 오버로딩 + 디폴트 파라미터 사용
1. geometric_sqc_sum(): a_1 = 1, r = 2, n = 5
2. geometric_sqc_sum(x): a_1 = x, r = 2, n = 5
3. geometric_sqc_sum(x, y): a_1 = x, r = y, n = 5
4. geometric_sqc_sum(x, y, z): a_1 = x, r = y, n = z
//등비수열의 합을 구하는 함수
#include <iostream>
#include <cmath>
using namespace std;
double geometric_sqc_sum(int a = 1, int r = 2, int n=5) {
 if (r == 1) return n * a;
 else return (a * ((pow(r,n)-1))) / (r-1);
int main() {
 cout << geometric_sqc_sum() << endl;
 cout << geometric_sqc_sum(5) << endl;</pre>
 cout << geometric_sqc_sum(5, 2) << endl;</pre>
 cout << geometric_sqc_sum(5, 3, 6) << endl;
 return 0;
```

Day02 - 3. inline

inline

- 함수의 호출부가 호출함수의 정의부로 대체
- 함수의 정의부가 간단한 경우 선언하는 키워드다.
- 함수의 호출 없이 정의부의 코드가 호출 부에 복사 (호출 횟수 × N번 복사)
- 일반 함수보다 더 빠르게 실행하지만 메모리를 더 사용한다.
- inline함수에서 일반함수(inline으로 선언되지 않은 함수)를 호출하거나 함수의 코드 부분이 너무 큰 경우에는 사용자가 inline으로 지정해도 컴파일러가 무시하기도 한다.
- 내부연결(internal linkage)을 갖기 때문에 반드시 헤더파일에 작성
- 컴파일러에 의해 처리

함수 호출 과정

- (1) 매개변수를 스택(Stack)에 저장 (매개변수 전달)
- (2) 함수 호출명령 바로 다음 명령의 주소를 스택에 저장 (호출한 곳으로 다시 돌아오기 위해)
- (3) 함수가 정의된 주소로 점프, 함수 코드를 수행
- (4) 리턴 값을 저장
- (5) (2)에 저장해 둔 주소로 돌아온다.

Day02 - 3. inline

```
macro vs inline
매크로 함수는 컴파일 되기 전에 전처리기에 의해 처리된다.
매크로 함수 정의 시 반드시 괄호를 써서 잘못된 우선순위연산을 방지
연산 식 중에는 괄호를 쓴다고 해도 원치 않는 결과를 초래된다.
#include <iostream>
using namespace std;
#define SUM(a,b) ((a)+(b))
#define SQ(x)((x)*(x))
#define MUL(a,b)((a)*(b))
inline int square(int x) {//컴파일러에 의해 처리되는 inline함수를 이용
return x * x;
int main() {
 cout << "(3+4)*5=" << SUM(3, 4) * 5 << endl;
 cout << "(1+2)*(3+4)=" << MUL(1+2, 3+4) << endl;
 int n = 3:
 cout << "macro: " << SQ(++n) << endl;
 //매크로 함수 SQ(++n) = ( (++n) * (++n) ) , 매크로 함수의 사용은 권장하지 않는다.
 n = 3;
 cout << "inline : " << square(++n) << endl;// square(++x) > x=++n; x*x;
 return 0;
}// 매크로 함수의 문제점을 해결하면서 실행 성능을 향상.
```

Day02 - 4. namespace

namespace(이름공간)

대형 프로젝트를 진행할 때 or 이전에 프로젝트를 가져올 때 중복되는 이름 충돌을 막기 위한 도구 (using namespace std; 추가한 이유)

이름 공간 안에 선언 된 멤버는 바로 호출이 불가능 반드시 어느 이름 공간에 속한 멤버인지를 밝힌 후 사용 (using namespace 이름공간명;)

iostream 헤더파일은 모든 클래스와 객체, 함수들이 std라는 이름 공간 안에 정의

이름 공간 안에 정의된 심볼은 아래의 3가지 방법으로 호출할 수 있다.

- (1) namespace명 :: 식별자를 사용 [std::cout;]
- (2) using 지시어를 사용하여 이름 공간 안의 특정 심볼(symbol) 사용 [using std::cout;]
- (3) using 지시어를 사용하여 이름 공간 안의 모든 심볼 사용 [using namespace std;]
- :: 연산자(영역 지정 연산자)는 멤버에 접근할 때 사용하는 연산자다.

Day02 - 4. namespace

namespace

(1) 하나의 파일에 여러 개의 namespace를 쓸 수 있다.

namespace A
namespace B
File

(2) 여러 개 파일에 한 개의 namespace를 쓸 수 있다.

namespace C
File File File

표준 헤더파일 C++표준안으로 C의 기본 헤더파일들도 다시 만들었다. iostream.h -> iostream | stdlib.h -> cstdlib string.h -> cstring | math.h -> cmath 모든 심볼이 std라는 이름 공간 안에 선언되어 있는지 아닌지의 차이

```
#include <iostream>
using namespace std;
namespace A {
  void BUBLE() {
     cout << "A::BUBLE\n";</pre>
  void pop() {
     cout << "A::POP!\n";
namespace B {
  void BUBLE() {
     cout << "B::BUBLE\n";</pre>
  void pop() {
     cout << "B::POP! \n";
using namespace B;
int main() {
  A::BUBLE();
  BUBLE();
  A::pop();
  pop();
  return 0;
```

Day03 - Contents

- 1. Bool 형
- 2. Reference
- 3. 예제 코드

Day03 - 1. Bool 형

Bool 형 true와 false의 값을 갖는 1byte 데이터 형 (참과 거짓을 판별) 내부적으로 true =1, false = 0 처리 1과 0으로 참과 거짓을 판별하는 작업을 좀 더 직관적으로 처리 #include <iostream> using namespace std; bool positiveNumber(int number) { //bool: true, false의 값을 갖는 1바이트 자료형(true는 1로, false는 0으로 처리된다.) if (number > 0) return true; //1; else return false; //0; int main() { int num; bool start = -10.125; // 0이 아니면 true bool stop = 0; // false start = 0;cout << sizeof(bool) << endl; cout << "정수 입력:"; cin >> num; if (num == 0) cout << "0 입니다." << endl; else if (positiveNumber(num)) cout << "양의 정수 입니다." << endl; else cout << "음의 정수 입니다." << endl; return 0;

판정 함수 bool

- 1. 대문자 판정 bool isUpper(char ch);
- 2. 소문자 판정 bool isLower(char ch);
- 3. 양수 판정 bool isPositive(int n);
- 4. 음수 판정 bool isNegative(int n);

Day03 - 1. Bool 형

```
#include <iostream>
using namespace std;
//bool:true, false의 값을 갖는 1바이트 자료형(true는 1로, false는 0으로 처리된다.)
bool isUpper(char ch) {
 return('A' <= ch && ch <= 'Z') ? true : false;
bool isLower(char ch) {
 return('a' <= ch && ch <= 'z')? true: false;
bool isPositive(int n) {
 return (0 < n)? true: false;
bool isNegative(int n) {
 return (0 > n)? true: false;
int main() {
 int num;
 char ch;
 while (1) {
  cout << "문자 입력: ";
  cin >> ch;
  cout << "숫자 입력: ";
  cin >> num:
  cout << "문자 판별: " << (isUpper(ch)? "대문자 입니다.": "소문자 입니다.") << endl;
  cout << "문자 판별: " << (isLower(ch)? "소문자 입니다.": "대문자 입니다.") << endl;
  cout << "숫자 판별: " << (isPositive(num)? "양수 입니다.": "음수 입니다.") << endl;
  cout << "숫자 판별: " << (isNegative(num)? "음수 입니다.": "양수 입니다.") << endl << endl;
 return 0;
```

Day03 - 2. Pointer

포인터 선언과 초기화 포인터의 정의 메모리의 주소를 저장하는 변수 메모리 주소를 가리키는 변수로 포인터 변수라고 한다.

포인터 선언 - * (asterisk)연산자 사용 형식 자료형 * 포인터 변수 이름; //int * score; 자료형의 주소를 저장 할 수 있는 포인터 변수 선언

포인터 초기화 - & (ampersand) 연산자 사용 형식 포인터 변수 이름 = &변수; // int * score = &a; &(변수)를 이용하면 변수의 주소를 알 수 있다.

메모리 주소	변수	값		메모리 주소	변수	값
•••	•••		*pc	•••	•••	•••
0x7ffffb0ed245	рс	0x7ffffb0ee612		0x7ffffb0ee612	С	A
•••	•••	•••		•••	•••	•••

Day03 - 2. Pointer

포인터 선언과 초기화

예시

포인터 변수가 가리키는 변수의 자료 타입이 같아야 한다.

int a=10; 시작 주소 + 자료형 크기로 메모리 읽기

int *pa=&a; int * (시작 주소 + sizeof(int))

1 2 3 4

메모리 주소	변수	값		메모리 주소	변수	갔
	•••	•••	*pa		•••	
0x7ffffb0ee000	pa	0x7ffffb0ee628		0x7ffffb0ee628	а	10
	•••				•••	•••

// 'A' = 65 char c='A'; char *pc=&c; 포인터 변수가 가리키는 변수의 자료 타입이 같아야 한다. 시작 주소 + 자료형 크기로 메모리 읽기

시식 수소 + 사료영 크기도 메모리 읽/ char * (시작 주소 + sizeof(char))

메모리 주소	변수	값		메모리 주소	변수	값
•••	•••		*pc	•••	•••	
0x7ffffb0ed245	рс	0x7ffffb0ee612		0x7ffffb0ee612	С	A
	•••	•••			•••	• • •

Day03 - 2. Reference

Reference

C언어는 변수에 한 개의 이름만을 부여 C++에서는 변수에 여러 개의 이름을 부여 즉, 서로 다른 이름으로 같은 변수에 접근 가능 //별칭 변수에 또 다른 이름을 붙이는걸 Reference reference는 C에서 사용한 포인터의 역할, 많은 부분을 대신 한다. C++에서 reference와 pointer의 차이는 메모리 접근의 가능성

특징

- (1) reference 선언 시 & 연산자를 쓴다.
- (2) 선언과 동시에 초기화되어야 한다.
- (3) 리터럴 상수(1, 2, 3, ···)는 reference 대입할 수 없다.

```
int a = 3; // a는 3으로 초기화
int &r=a; // reference r은 변수 a의 메모리를 공유
++a; // a == 4
++r; // r == a == 5
++a는 1 증가하여 4가 된다.
++r에서 r은 a와 같은 메모리를 공유하기 때문에 1 증가하여 5가 된다.
즉, 변수와 reference는 완전히 같다고 생각해도 무방하다.
```

Day03 - 2. Reference

연산자 * , 연산자 &

연산자	용도	형식	예시
	곱하기	변수 or 상수	a*b; x*3;
*	포인터 선언 주소를 저장하는 변수	데이터 형 * 변수명	<pre>int * ip; double * dp;</pre>
	포인터 참조(간접접근) 포인터가 가리키는 메모리 접근	* 포인터명	*ip; *dp;
	비트 논리 연산자	변수 or 상수	a & b; x & 3;
&	주소 변수의 주소	<mark>&</mark> 변수명	&a &b
	reference 메모리 공유	데이터형 &변수명	int &a double &d

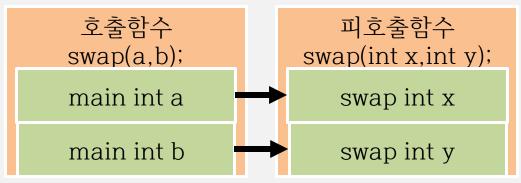
변수 선언할 때 * 연산자를 쓰면 포인터, & 연산자를 쓰면 reference

```
#include <iostream>
using namespace std;
int main() {
 int a = 100, b=200;
 int* p = &a;
 int\& r = a;
 cout << "a의 주소 =" << &a << endl;
 cout << "r의 주소 =" << &r << endl;
 cout << "p의 값 =" << p << endl;
 cout << "p의 주소 =" << &p << endl;
 cout << "a = " << a << "r = " << r << "(레퍼런스) " << "*p = " << *p << "(포인터 p 가 가리키는 값) " << endl;
 ++a;
 cout << "a = " << a << "r = " << r << "(레퍼런스) " << "*p = " << *p << "(포인터 p 가 가리키는 값) " << endl;
 ++r;
 cout << "a = " << a << "r = " << r << " (레퍼런스) " << "*p = " << *p << " (포인터 p 가 가리키는 값) " << endl;
 r = b; // reference는 다른 변수로 변경 불가
 cout << "a = " << a << "r = " << r << " (레퍼런스) " << "*p = " << *p << " (포인터 p 가 가리키는 값) " << endl;
 return 0:
```

Day03 - 2. Reference

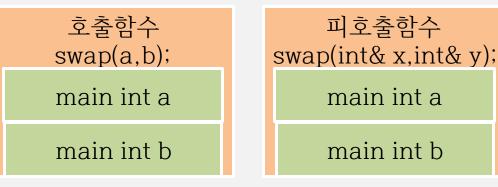
Call by value, Call by reference

- Call by value 호출함수의 변수 값을 복사해서 전달하는 방식이다. 호출 함수의 변수에는 아무런 영향을 미치지 않는다. Call by value 데이터를 변경 할 경우 초기화가 필요. 값 복사



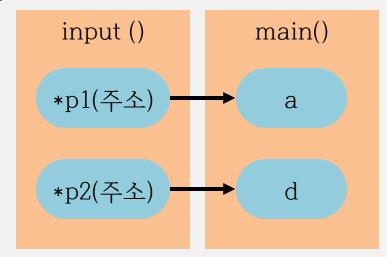
- Call by reference 호출함수 변수의 주소를 공유하는 방식 호출함수 변수에 간접 접근할 수 있다.

Call by reference 같은 변수, 별칭 (복사 X)

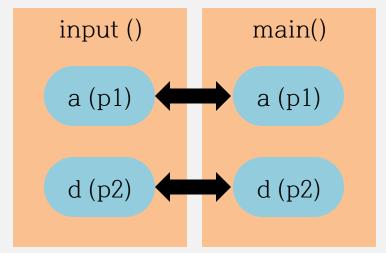


Day03 - 2. Reference

```
#include <iostream>
using namespace std;
void input(int* p1, double* p2) {
// int* p1 = &a1, double* p2 = &d1
    cout << "정수입력:";
    cin >> *p1;
    cout << "실수입력:";
    cin >> *p2;
}
int main() {
    int a1;
    double d1;
    input(&a1, &d1);
    cout << a1 << ", " << d1 << endl;
    return 0;
}
```



```
#include <iostream>
using namespace std;
void input(int& p1, double& p2) {
// int \& p1 = a2, double \& p2 = d2
  cout << "정수입력:";
  cin >> p1;
  cout << "실수입력:";
  cin >> p2;
int main() {
  int a2;
  double d2;
  input(a2, d2);
  cout << a2 << ", " << d2 << endl;
  return 0;
```



Reference 예제

1. 문자, 정수, 실수 값을 swap하는 함수, reference 구현

```
void swap(char& x, char& y);
void swap(int& x, int& y);
void swap(double& x, double& y);
```

2. 문자열 swap하는 함수, reference 구현 void swap(char* p1, char* p2);

char * (문자형 주소) 를 교환 하지 않고 값을 교환

```
#include <iostream>
using namespace std;
void swap(char& x, char& y);
void swap(int& x, int& y);
void swap(double& x, double& y);
void swap(char* p1, char* p2);
int main() {
 charc1, c2;
 cout << "문자 2개 입력 (공백 구분):";
 cin >> c1 >> c2;
 cout << "교환 전:" << c1 << "." << c2 << endl;
 swap(c1, c2);
 cout << "교환 후:" << c1 << ", " << c2 << endl << endl;
 int i1. i2;
 cout << "정수 2개 입력 (공백 구분): ";
 cin >> i1 >> i2;
 cout << "교화 전:" << i1 << "." << i2 << endl;
 swap(i1. i2);
 cout << "교환 후:" << i1 << ", " << i2 << endl << endl;
 double d1. d2;
 cout << "실수 2개 입력 (공백 구분): ";
 cin >> d1 >> d2;
 cout << "교환 전:" << d1 <<", " << d2 << endl;
 swap(d1, d2);
 cout << "교환 후:" << d1 << ", " << d2 << endl << endl;
 char str1[255], str2[255];
 cout << "문자열 2개 입력 (공백 구분): ";
 cin >> str1 >> str2;
 cout << "교환 전:" << str1 << ", " << str2 << endl;
 swap(str1, str2);
 cout << "교환 후: " << str1 << ", " << str2 << endl << endl;
 return 0;
```

```
void swap(char& x, char& y) {
 chartemp = x;
 x = y;
 y = temp;
void swap(int& x, int& y) {
 int temp = x;
 x = y;
 y = temp;
void swap(double& x, double& y) {
 double temp = x;
 x = y;
 y = temp;
void swap(char* p1, char* p2) {
 chartemp[255];
 for (int i = 0; i \le strlen(p1); i++) temp[i] = p1[i];
 for (int i = 0; i <= strlen(p2); i++) p1[i] = p2[i];
 for (int i = 0; i \le strlen(temp); i++) p2[i] = temp[i];
```

```
#include <iostream>
using namespace std;
struct student {//C++에서는 구조체도 완벽한 자료형 typedef 필요없음
 int no:
 char name[20];
 char phone[20];
};
void output(const student* p) { //st의 주소를 저장하는 포인터 p
 //p->no = 9999; //const 포인터가 가리키는 값을 수정 불가능
// 포인터의 간접접근(역참조) 사용
 cout << "포인터 (주소) 번호: " << p->no << ", 이름: " << p->name << ", 전화번호: " << p->phone << endl;
void output(const student& ref) { //st의 메모리를 공유하는 레퍼런스 ref
//ref.no = 5555; //const 레퍼런스로 구조체 변경 불가능
// 레퍼런스 메모리 공유 사용 ( 구조체 멤버 접근 )
 cout << "레퍼런스(별칭) 번호: " << ref.no << ", 이름: " << ref.name << ", 전화번호: " << ref.phone << endl;
int main() {
 student st = { 1, "christopher", "010-1234-5678" };
 output(&st);
 output(st);
 return 0;
```

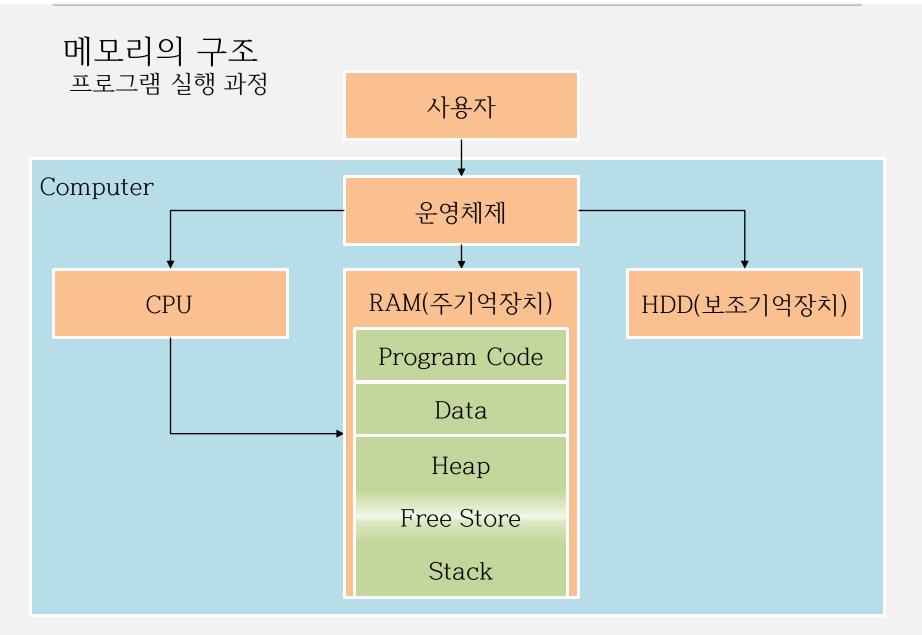
Day 04 - Contents

- 1. 기억클래스- 지역, 전역, 정적 변수
- 2. new/delete
- 3. 예제 코드

기억클래스

- (1) 지역 변수
 - 함수나 블록({}) 내에서 선언된 변수로 선언된 함수(블록) 내에서만 사용
 - 함수가 종료되면 메모리에서 소멸
 - 매개변수도 지역 변수
 - 메모리 영역에 원래 존재하고 있는 값(쓰레기 값)으로 초기화 필요.
- (2) 전역 변수
 - 함수 외부에서 선언된 변수
 - 프로그램이 실행될 때 메모리 할당, 프로그램이 종료되면 메모리 소멸
 - 초기화하지 않으면 ()으로 자동 초기화
 - 모든 함수에서 자유롭게 사용할 수 있어 프로그램 작성이 쉽다
 - 프로그램 모듈화에 방해가 되므로 가급적 사용을 피해주는 것이 좋다.
- (3) 정적 변수
 - static을 붙여 선언한 변수
 - 함수나 블록 내에서 선언된 변수는 선언된 함수나 블록 내에서만 사용 (지역 변수 특징)
 - 프로그램이 실행될 때 메모리 할당, 프로그램이 종료되면 메모리 소멸 (전역 변수 특징)
 - 전역 변수 보다는 정적 변수의 사용을 권장한다.

```
// 지역변수 a, 정적변수 stt_a, 전역변수 global
#include <iostream>
#include <iomanip>
using namespace std;
int global = 0;
void fun() {
  int a = 0;
  static int stt_a;
  cout \ll setw(3) \ll a \ll setw(8) \ll setw(8) \ll global \ll endl;
  a++;
  stt_a += 2;
  global += 3;
int main() {
  cout << "지역 | static | 전역" << endl;
  for (int i = 0; i < 10; i++) fun();
  return 0;
```



메모리의 구조 메모리 영역

RAM(주기억장치)
Program Code
Data
Heap
Free Store
Stack

Code 영역: 순차적인 명령문(제어문, 함수, 상수, …)의 저장 공간 CPU는 코드 영역에 명령어를 하나씩 가져가서 처리

Data 영역 전역변수, 정적 변수,… 저장 공간 프로그램 실행 전에 선언, 프로그램이 끝날 때 까지 살아있는 변수들

Heap 영역: 사용자의 동적 할당 (프로그램 실행 중 메모리 할당) 보통 낮은 주소 번지에서 높은 주소 번지로 할당

Stack 영역: 지역 변수, 매개 변수, ({} 안에 변수들, …) 함수가 종료되면 해당 함수에 할당된 변수들을 메모리에서 해제 보통 높은 주소 번지에서 낮은 주소 번지로 할당

Free Store: Heap or Stack 메모리 모두 할당 가능

Stack 영역의 메모리가 늘어나면 Heap 영역의 메모리가 줄어들고 Heap 영역의 메모리가 늘어나면 Stack 영역의 메모리가 줄어든다.

메모리의 구조 메모리 영역

RAM(주기억장치)

Program Code

Data

Heap

Free Store

Stack

	메모리 주소	변수	값
Code	0x0000000000	•••	
	0x0000000001	•••	•••
	•••	•••	•••
	0xFFFB0EE620	g	103
	0xFFFB0EE621	f	102
	0xFFFB0EE622	е	101
	0xFFFB0EE623	d	100
	0xFFFB0EE624	С	99
	0xFFFB0EE625	b	98
	0xFFFB0EE626	а	97
	•••	•••	•••
	0xFFFFFFFE	•••	•••
Stack	0xFFFFFFFF	•••	•••

Memory(메모리)

프로그램이 실행되는 동안 저장되는 데이터는 메모리 공간(RAM)에 저장 메모리는 크게 Stack, Heap, Data 영역으로 구분 Stack 영역이 많이 할당되면 Heap 영역은 줄어들고, Heap 영역이 많이 할당되면 Stack 영역은 줄어드는 가변적인 공간 Stack 영역과 Data 영역은 시스템에 의해 할당되고 해제되는 영역, Heap 영역은 프로그래머에 의해 할당 및 해제가 이뤄지는 영역.

메모리의 정적 할당과 동적 할당

- (1) 정적 할당
 - Compile(컴파일)시 할당될 메모리 크기가 결정
 - Data 영역에 할당되는 메모리는 모두 정적 할당
 - 전역 변수, 정적 변수 모두 컴파일 시 할당될 정적 할당
- (2) 동적 할당 [형식: 자료형 * 변수명 = new 자료형;, delete 변수명;]
 - Runtime(실행)중에 메모리가 할당
 - 동적 할당 메모리 영역: Heap 사용자 선택, stack 변수 사용

```
동적 할당
형식
자료형 * 변수명 = new 자료형;
delete 변수명;
int* ip = new int; //동적 할당 생성
 delete ip; // 동적 할당 해제
 형식
자료형 * 변수명 = new 자료형[n];
 delete[] 변수명;
int* ip = new int[5]; //동적 할당 생성
 delete[] ip; // 동적 할당 해제
new 연산자
동적 할당된 메모리의 시작주소를 리턴
동적 할당된 데이터 형과 같은 타입의 포인터가 동적 메모리의 주소를 저장
동적으로 할당 된 메모리는 해제하지 않으면 메모리 누수가 발생
동적으로 할당된 메모리는 반드시 해제가 필요
```

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main(){
   cout << "대문자 개수 입력:";
   int alphaCnt;
   cin >> alphaCnt;
   char* sentence = new char[alphaCnt];
   srand((unsigned int)time(NULL));
   for (int i = 0; i < alphaCnt; i++) {
      sentence[i] = (rand() % 26) + 65;
      cout << sentence[i] << ' ';
   }
   delete[] sentence;
   return 0;
}
```

동적 할당 코드 작성

```
#include <iomanip>//setw() 자리수 출력
#include <cstdlib>//srand() rand() 랜덤 값 출력
#include <cmath>//sqrt() 제곱근 함수
```

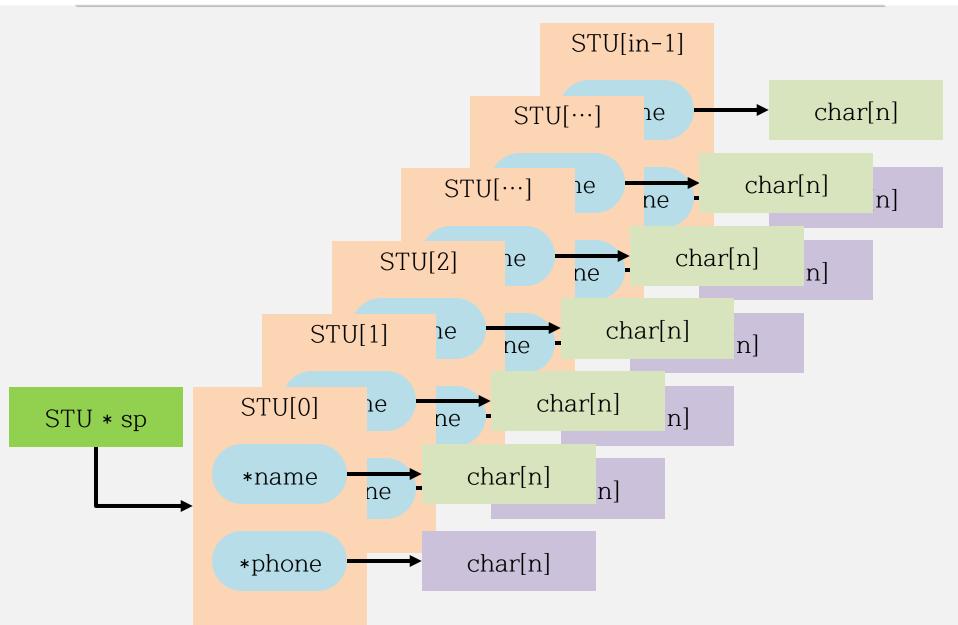
- 1. 학생 수 입력 받기
- 2. 학생 수 만큼 성적을 저장 할 메모리 공간 동적 할당
- 3. 각 학생 수 성적 입력 (rand() : 0~100)
- 4. 반 학생 성적 평균 값 계산
- 5. 반 학생 성적 분산 값 계산
- 6. 반 학생 성적 표준편차 값 계산
- 7. 각 학생 성적 모두 출력, 평균, 분산, 표준편차 값 출력

```
#include <iostream>
#include <iomanip>
#include <cstdlib>
#include <cmath>
using namespace std;
int main() {
  cout << "학생 수 입력하세요.:";
  int std_num;
  float sum = 0, ave = 0, var =0;
  cin >> std num;
  int* score = new int[std_num]; // int score[std_num];
  float * variance = new float[std_num]; // float variance[std_num];
  srand((unsigned int)time(NULL));
  for (int i = 0; i < std_num; i++) {
     score[i] = rand() % 101;
     sum += score[i];
  ave = sum / std_num;
  for (int i = 0; i < std_num; i++) {
     variance[i] = (score[i] - ave) * (score[i] - ave);
     var += variance[i];
     cout << "점 수: " << setw(6)<< score[i] << " 편차 제곱: " << setw(6) << variance[i] << endl;
  var /= std_num;
  cout << "총 점: " << sum << " 평 균: " << ave << " 분 산: " << var << " 표준편차: " << sgrt(var) << endl;
  delete[]score;
  delete[]variance;
  return 0:
```

동적 할당 코드 작성

```
#include <iostream>
#include <cstring>
struct STU {
   char* name, * phone;
};
```

- 1. 학생 수 입력 받기
- 2. 학생 수 만큼 STU 메모리 공간 동적 할당
- 3. 각 학생 수만큼 이름과 전화번호 입력 받기
- 4. 이름, 전화번호 문자열 길이만큼 동적 할당
- 5. 학생 수만큼 이름, 전화번호 출력

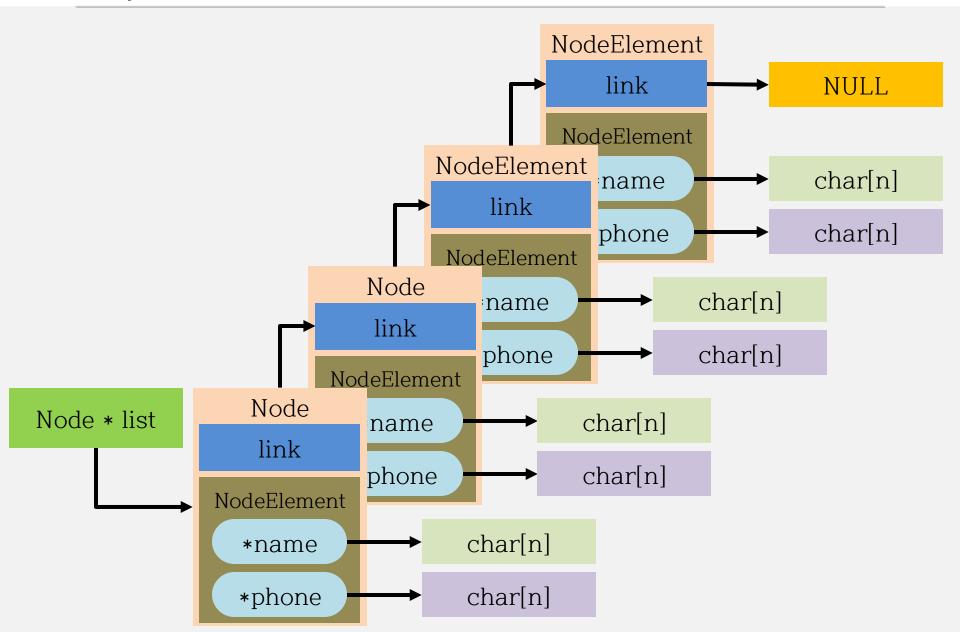


```
#include <iostream>
#include <cstring>
using namespace std;
struct STU {
  char* name. * phone;
};
int main() {
  int in;
  cout << "학생수 입력:";
  cin >> in;
  STU*sp = new STU[in];
  char buf[30];
  for (int i = 0; i < in; i++) {
     cout << "\n이 름:";
     cin >> buf;
     sp[i].name = new char[strlen(buf) + 1];
     strcpy_s(sp[i].name,strlen(buf)+1, buf);
     cout << "이름 할당 완료 ";
     cout << sp[i].name << endl;
     cout << "전화번호:";
     cin >> buf;
     sp[i].phone = new char[strlen(buf) + 1];
```

```
strcpy_s(sp[i].phone, strlen(buf) + 1, buf);
  cout << "번호 할당 완료 ";
  cout << sp[i].phone << endl;
for (int i = 0; i < in; i++) {
  delete[] sp[i].name;
  cout << i << "이름 할당 해제 " << endl;
  delete[] sp[i].phone;
  cout << i << "번호 할당 해제 " << endl;
delete[]sp;
cout << "구조체 할당 해제 " << endl;
return 0;
```

```
[명단 관리 프로그램]
1. 등록
                     REGIST
2. 삭제
                     REMOVE
3. 리스트 출력
                    DISPLAY
4. 전체 삭제
                    REMOVEALL
0. 프로그램 종료
                    TERMINATE
메뉴 선택:
struct STU {
 char* name, * phone;
};
                                                                   void cinput(char& value) {
typedef STU NodeElement;
                                                                    while (!(cin >> value)) {
struct Node {
                                                                     cinclear();
                                                                     cout << "\t다시 입력 하세요.: ";
 NodeElement value;
 Node* link;
};
                                                                   void cinput(int& value) {
                                                                    while (!(cin >> value)) {
void PrintMenu(int& menu);
                                                                     cinclear();
bool InsertNode(Node*& head, NodeElement data);
                                                                     cout << "\t다시 입력 하세요.: ";
bool DeleteNode(Node*& head, NodeElement data);
bool DeleteAll(Node*& head);
                                                                   void cinclear() {
bool PrintList(Node*& head);
                                                                    if (cin.fail()) {
void cinput(char& value);
                                                                     cin.clear();
                                                                     while (cin.get() != '\n');
void cinput(int& value);
void cinclear();
                                                                    else while (cin.get() != '\n');
```

Day04 - 2. 예제 코드



```
#include <iostream>
                                                     cin >> buf:
                                                                                                    cout << "목록을 추가해 주세요. \n\n";
                                                                                                                                                     cout << "목록이 비어있습니다.\n\n";
#include <cstdlib>
                                                     cinclear();
                                                                                                    return false;
                                                                                                                                                     return false;
                                                    Deletion.name = new char[strlen(buf) + 1];
using namespace std;
enum menuType
                                                     strcpy_s(Deletion.name, strlen(buf) + 1, buf);
                                                                                                  Node* DeleteNode = head, * preDelete = NULL;
                                                                                                                                                    for (Node* pNode = head; pNode != NULL; pNode
                                                                                                   while (DeleteNode != NULL) {
                                                                                                                                                   = pNode->link) {
 TERMINATE, REGISTER, REMOVE, DISPLAY,
                                                    cout << "\t\t 전화번호:";
                                                                                                    if ((strcmp(DeleteNode->value.name,
                                                                                                                                                     cout << endl;
REMOVEALL.
                                                    cin >> buf;
                                                                                                  data.name) == 0) && (strcmp(DeleteNode-
                                                                                                                                                     cout << "\t\t 이름: " << pNode->value.name
                                                    cinclear();
                                                                                                  >value.phone, data.phone) == 0)) {
                                                                                                                                                   << endl;
struct STU {
                                                    Deletion.phone = new char[strlen(buf) + 1];
                                                                                                     if (preDelete == NULL) head = DeleteNode-
                                                                                                                                                     cout << "\t\t 전화번호: " << pNode-
char* name, * phone;
                                                    strcpy_s(Deletion.phone, strlen(buf) + 1, buf); >link;
                                                                                                                                                   >value.phone << endl;
                                                                                                     else preDelete->link = DeleteNode->link;
typedef STU NodeElement;
                                                     DeleteNode(list1, Deletion);
                                                                                                     delete[] DeleteNode->value.name;
                                                                                                                                                    return true;
struct Node {
                                                    PrintList(list1);
                                                                                                     delete[] DeleteNode->value.phone;
NodeElement value;
                                                    break;
                                                                                                     delete DeleteNode;
                                                                                                                                                   void cinclear() {
 Node* link;
                                                   case DISPLAY: PrintList(list1); break;
                                                                                                     cout << "삭제 성공! \n\n";
                                                                                                                                                    if (cin.fail()) {
                                                   case REMOVEALL:
                                                                                                                                                     cin.clear();
                                                                                                     return true;
                                                                                                                                                     while (cin.get() != '\n');
                                                    DeleteAll(list1);
void PrintMenu(int& menu);
bool InsertNode(Node*& head, NodeElement data);
                                                    PrintList(list1);
                                                                                                    preDelete = DeleteNode;
bool DeleteNode(Node*& head, NodeElement data);
                                                                                                    DeleteNode = DeleteNode->link;
                                                                                                                                                    else while (cin.get() != '\n');
bool DeleteAll(Node*& head);
                                                   case TERMINATE:
bool PrintList(Node*& head);
                                                    cout << "\n\n\t\t[ 프로그램 종료 합니다. \\n\n"; cout << "목록에 삭제 할 자료가 없습니다. \\n\n";
                                                                                                                                                   void cinput(char& value) {
void cinput(char& value);
                                                                                                   return false;
                                                                                                                                                    while (!(cin >> value)) {
void cinput(int& value);
                                                                                                                                                     cinclear();
void cinclear();
                                                   cout << endl << endl;
                                                                                                  bool DeleteAll(Node*& head) {
                                                                                                                                                     cout << "\t다시 입력 하세요.: ";
                                                                                                   cout << endl << endl;
int main() {
                                                   system("pause");
int menu = 1;
                                                                                                   cout << "\t\t [ 목록을 모두 지우시겠습니까??
                                                                                                                                                    while (cin.get() != '\n');
 Node* list1 = NULL;
                                                  return 0;
                                                                                                  (Y/N) l\n\n";
 NodeElement Register, Deletion;
                                                                                                   cout << "\t\t 입력하세요.:";
                                                                                                                                                   void cinput(int& value) {
                                                                                                                                                    while (!(cin >> value)) {
 char buf[3];
                                                 void PrintMenu(int& menu)
                                                                                                   char answer = 'n';
 while (1) {
                                                                                                   cinput(answer);
                                                                                                                                                     cinclear();
  system("cls");
                                                  cout << "\n\n\t\t[ 명단 관리 프로그램 ]\n\n";
                                                                                                   if (answer == 'Y' || answer == 'y') {
                                                                                                                                                     cout << "\t다시 입력 하세요.: ";
  PrintMenu(menu);
                                                  cout << "\t\t1. 등록\n";
                                                                                                    for (Node* pNode = head, *DeleteNode = NULL;
  switch (menu) {
                                                  cout << "\t\t2. 삭제\n";
                                                                                                  pNode != NULL;) {
                                                                                                                                                    while (cin.get() != '\n');
  case REGISTER:
                                                  cout << "\t\t3. 리스트 출력\n";
                                                                                                     DeleteNode = pNode;
   cout << "\t\t 이름:";
                                                  cout << "\t\t4. 전체 삭제\n";
                                                                                                     pNode = pNode->link;
   cin >> buf;
                                                  cout << "\t\t0. 프로그램 종료\n\n";
                                                                                                     delete[] DeleteNode->value.name;
   cinclear();
                                                  cout << "\t\t메뉴 선택:";
                                                                                                     delete[] DeleteNode->value.phone;
   Register.name = new char[strlen(buf) + 1];
                                                  cinput(menu);
                                                                                                     delete DeleteNode;
   strcpy_s(Register.name, strlen(buf) + 1, buf);
                                                 bool InsertNode(Node*& head, NodeElement data)
                                                                                                    head = NULL;
   cout << "\t\t 전화번호:";
                                                                                                    cout << "\t\t [ 목록을 모두 지웠습니다. ]\n\n";
   cin >> buf;
                                                  Node* NewNode = new Node;
                                                                                                    return true;
   cinclear();
                                                  if (NewNode == NULL) return false;
   Register.phone = new char[strlen(buf) + 1];
                                                  NewNode->value = data:
   strcpy_s(Register.phone, strlen(buf) + 1, buf);
                                                 NewNode->link = head;
                                                                                                    cout << "\t\t [ 목록을 지우지 않고 메뉴로 돌아갑니
                                                  head = NewNode;
                                                                                                  다. ]\n\n";
   InsertNode(list1, Register);
                                                  return true;
                                                                                                    return false;
   break;
                                                 bool DeleteNode(Node*& head, NodeElement data)}
  case REMOVE:
                                                                                                  bool PrintList(Node * & head) {
   cout << "\t\t 이름:";
                                                                                                   if (head == NULL) {
                                                  if (head == NULL) {
```

Day05 - Contents

- 1. 구조체와 클래스
- 2. 클래스와 객체
- 3. 예제 코드

```
구조체 (struct)
정의
두 개 이상의 변수(자료형)를 그룹화 ( typedef 필요없음 )
구현하려는 대상의 속성을 묶어 프로그램 구현 및 관리가 쉽다.
형식
              예시
              struct human
struct 구조체명
 자료형 멤버 1;
               char name[10];
 자료형 멤버 2;
               int age;
구조체 선언
구조체명 변수명;
구조체 멤버의 참조
구조체 변수명 . 멤버명;
구조체 초기화
구조체명 구조체 변수명 = {값1, 값2, 값3, …};
```

구조체 (struct)

정의

두 개 이상의 변수(자료형)를 그룹화 (typedef 필요없음) 구현하려는 대상의 속성을 묶어 프로그램 구현 및 관리가 쉽다.

속성(데이터)만 묶어 자료 처리 시, 함수에 매번 매개변수로 넣어야 한다.

- 함수를 구조체에 같이 포함 시켜서 속성 + 자료 처리 함수로 해결

구조체 변수 1

멤버 변수 1.1

멤버 변수 1.2

멤버 함수 1.1

멤버 함수 1.2

구조체 변수 2

멤버 변수 2.1

멤버 변수 2.2

멤버 함수 2.1

멤버 함수 2.2

구조체 변수 1

멤버 변수 1.1

멤버 변수 1.2

구조체 변수 2

멤버 변수 2.1

멤버 변수 2.2

멤버 함수 1

멤버 함수 2

공유

```
//구조체 멤버 변수 + 멤버 함수 ver.
//구조체 멤버 변수 ver.
#include <iostream>
                                                         #include <iostream>
using namespace std;
                                                         using namespace std;
struct ACT {
                                                         struct ACT{
                  //계좌번호
                                                                            //계좌번호
  int ACTId;
                                                           int ACTId;
  char pw[5];
                  //비밀번호
                                                           char pw[5];
                                                                            //비밀번호
  char name[20];
                   //이름
                                                                             //이름
                                                           char name[20];
                //잔액
                                                           int bal;
  int bal;
};
                                                           void showData() {
                                                             cout << "\n계좌번호:" << ACTId << endl;
void deposit(ACT& a); //입금
                                                             cout << name << "회원님의 잔액은 " << bal << "입니다.\n";
void withdraw(ACT& a); //출금
void showData(ACT& a); //잔액조회
                                                           void deposit() {
int main() {
  ACT act_1 = { 1234, "5678", "christopher", 10000 };
                                                             int m;
                                                              cout << "\n입금금액 입력:";
  showData(act_1); //잔액조회 함수 호출
  deposit(act_1); //입금함수 호출
                                                             cin >> m;
  withdraw( act_1 );
                   //출금함수 호출
                                                              bal += m;
  showData(act_1); //잔액조회 함수 호출
                                                             cout << "\t\t" << m << "원 입금완료.\n";
  return 0;
                                                           void withdraw() {
void showData( ACT& a ) {
                                                              int m:
  cout << "\n계좌번호:" << a.ACTId << endl;
                                                              cout << "\n출금금액 입력:";
  cout << a.name << "회원님의 잔액은 " << a.bal << "입니다.\n";
                                                             cin >> m;
                                                              hal -= m:
                                                             cout << "\t\t" << m << "원 출금완료.\n";
void deposit( ACT& a ) {
  int m;
  cout << "\n입금금액 입력:";
  cin >> m;
                                                         int main(){
                                                           ACT act_1 = { 1234, "5678", "christopher", 10000 };
  a.bal += m;
  cout << "\t\t" << m << "원 입금완료.\n";
                                                           act_1.showData(); //잔액조회 멤버 함수 호출
                                                           act_1.deposit();
                                                                             //입금 멤버 함수 호출
                                                                             //출금 멤버 함수 호출
void withdraw(ACT& a) {
                                                           act_1.withdraw();
                                                                            //잔액조회 멤버 함수 호춬
  int m:
                                                           act_1.showData();
  cout << "\n출금금액 입력:";
                                                           return 0;
  cin >> m;
  a.bal -= m:
  cout << "\t\t" << m << "원 출금완료.\n";
```

```
//구조체 멤버 변수 + 멤버 함수 ver.
#include <iostream>
using namespace std;
struct ACT{
                  //계좌번호
  int ACTId;
  char pw[5];
                  //비밀번호
                   //이름
  char name[20];
                //잔액
  int bal;
  void showData()
    cout << "\n계좌번호: " << ACTId << endl;
    cout << name << "회원님의 잔액은 " << bal << "입니다.\n";
  void deposit() {
    int m;
    cout << "\n입금금액 입력:";
    cin >> m;
    bal += m;
    cout << "\t\t" << m << "원 입금완료.\n";
  void withdraw() {
    int m:
    cout << "\n출금금액 입력:";
    cin >> m;
    hal -= m:
    cout << "\t\t" << m << "원 출금완료.\n";
int main(){
  ACT act_1 = { 1234, "5678", "christopher", 10000 };
  act_1.showData(); //잔액조회 멤버 함수 호출
  act_1.deposit();
                  //입금 멤버 함수 호출
                  //출금 멤버 함수 호출
  act_1.withdraw();
                  //잔액조회 멤버 함수 호출
  act_1.showData();
  return 0;
```

```
//구조체 멤버 변수 + 멤버 함수 선언ver.
#include <iostream>
using namespace std;
struct ACT {
                  //계좌번호
  int ACTId;
  char pw[5];
                  //비밀번호
  char name[20];
                   //이름
                //잔액
  int bal;
  void showData(); //계좌정보 출력
  void deposit(); //입금 함수 & 출력
  void withdraw();//출금 함수 & 출력
int main() {
  ACT act_1 = { 1234, "5678", "christopher", 10000 };
  act_1.showData(); //잔액조회 멤버 함수 호출
                    //입금 멤버 함수 호출
  act_1.deposit();
  act_1.withdraw();
                    //출금 멤버 함수 호출
                     //잔액조회 멤버 함수 호출
  act_1.showData();
  return 0;
void ACT::showData() {
  cout << "\n계좌번호: " << ACTId << endl;
  cout << name << "회원님의 잔액은 " << bal << "입니다.\n";
void ACT::deposit(){
  int m;
  cout << "\n입금금액 입력:";
  cin >> m;
  bal += m;
  cout << "\t\t" << m << "원 입금완료.\n";
void ACT::withdraw(){
  int m:
  cout << "\n출금금액 입력:";
  cin >> m;
  hal -= m:
  cout << "\t\t" << m << "원 출금완료.\n";
```

```
//구조체 멤버 변수 + 멤버 함수 선언ver.
#include <iostream>
using namespace std;
struct ACT {
                  //계좌번호
  int ACTId;
  char pw[5];
                 //비밀번호
  char name[20];
                   //이름
                //잔액
  int bal;
  void showData(); //계좌정보 출력
  void deposit(); //입금 함수 & 출력
  void withdraw();//출금 함수 & 출력
int main() {
  ACT act_1 = { 1234, "5678", "christopher", 10000 };
                  //잔액조회 멤버 함수 호출
  act_1.showData();
                   //입금 멤버 함수 호출
  act_1.deposit();
  act_1.withdraw();
                   //출금 멤버 함수 호출
                   //잔액조회 멤버 함수 호출
  act_1.showData();
  return 0;
void ACT::showData() {
  cout << "\n계좌번호:" << ACTId << endl;
  cout << name << "회원님의 잔액은 " << bal << "입니다.\n";
void ACT::deposit(){
  int m;
  cout << "\n입금금액 입력:";
  cin >> m;
  bal += m;
  cout << "\t\t" << m << "원 입금완료.\n";
void ACT::withdraw(){
  int m:
  cout << "\n출금금액 입력:";
  cin >> m;
  bal -= m:
  cout << "\t\t" << m << "원 출금완료.\n";
```

```
//class 멤버 변수 + 멤버 함수 선언ver.
#include <iostream>
#include <cstring>
using namespace std;
class ACT {
private:
 int ACTId;
                  //계좌번호
 char pw[5];
                  //비밀번호
 char name[20];
                    //이름
 int bal;
public:
 void Init_act(int_actid, const_char*_PW, const_char*_NAME, int_BAL);
 void showData(); //계좌정보 출력
 void deposit(); //입금 함수 & 출력
 void withdraw();//출금 함수 & 출력
int main() {
 ACT act_1;
 act_1.Init_act(1234, "5678", "christopher", 10000);
 act_1.showData();
                    //잔액조회 멤버 함수 호출
                   //입금 멤버 함수 호출
 act_1.deposit();
 act_1.withdraw();
                    //출금 멤버 함수 호출
 act_1.showData();
                     //잔액조회 멤버 함수 호출
 return 0:
void ACT::Init_act(int _actid, const char* _PW, const char* _NAME, int _BAL) {
 ACTId = _actid;
 strcpy_s(pw, strlen(_PW) + 1, _PW);
 strcpy_s(name, strlen(_NAME) + 1, _NAME);
 bal = _BAL:
void ACT::showData()
 cout << "\n계좌번호:" << ACTId << endl;
 cout << name << "회원님의 잔액은 " << bal << "입니다.\n";
void ACT::deposit() {
 int m;
 cout << "\n입금금액 입력:";
 cin >> m;
 bal += m;
 cout << "\t\t" << m << "원 입금완료.\n";
void ACT::withdraw() {
 int m;
 cout << "\n출금금액 입력:";
 cin >> m;
 bal -= m;
 cout << "\t\t" << m << "원 출금완료.\n";
```

Day05 - 2. 클래스와 객체

```
클래스
정의
 자료와 자료 처리에 관한 함수의 그룹화
(C 구조체: 자료의 그룹화, 클래스: 자료와 처리 함수의 그룹화)
형식(선언)
              예시
class 클래스명{
              class ACT {
private:
              private:
              int ACTId; //계좌번호
변수1;
               char pw[5]; //비밀번호
변수2;
               char name[20]; //이름
                           //잔액
protected:
               int bal;
public:
              public:
변수3;
               void Init_act(int actid, const char* pw, const char* name, int bal);
변수 연산 함수1; void showData(); //계좌정보 출력
변수 연산 함수2;
            void deposit(); //입금 함수 & 출력
변수 연산 함수3;
              void withdraw();//출금 함수 & 출력
              };
};
private: 클래스 내에서만 접근허용
```

public: 어디서든 접근허용

protected: 상속관계에서 접근허용

Day05 - 2. 클래스와 객체

클래스

특징 (구조체와 차이점)

- 접근제어 지시자(private, protected, public)를 사용한다.
- 새로운 접근제어 지시자에 따라 데이터 or 함수의 사용이 달라진다.
- struct의 접근제어 지시자의 디폴트는 public : (형식은 case : 문과 같다)
- class의 접근제어 지시자의 디폴트는 private:(형식은 case: 문과 같다)
- 클래스 밖으로 함수를 정의 해도 클래스의 일부로 private 접근 가능하다.

객체의 정의

선언된 클래스에 변수를 선언하면 하나의 객체라고 한다. 자료형(class, 객체형) 변수(객체, 인스턴스 변수)

ACT act_1, act_2, act_3, act_4, act_5, act_6; 객체 1, 객체 2, 객체 3, 객체 4, 객체 5, 객체 6,

파일 분할 (header.h / source.cpp / main.cpp)

- 헤더 파일 (.h): 클래스의 선언부, inline 함수 작성
- 소스 파일 (.cpp) : 함수의 정의부 작성
- 메인 파일 (.cpp) : main()함수 작성

Account struct 작성 struct ACT { unsigned long long ACTID;//계좌번호 char pw[7];//비밀번호 char* name;//이름 long long balance;//자액 **}**; void menu(int& choice); void init();//초기화 void create(ACT& AC);//개설 void deletion(ACT& AC);//삭제 void deposit(ACT& AC);//입금 void withdraw(ACT& AC);//출금 void showData(ACT& AC);//작액조회 bool checkpw(ACT& AC);//비밀번호 확인 bool cinclear();

[계좌 관리 프로그램]

- 1. 개설
 - 이름 입력 받기
 - 비밀번호 6자리 입력 받기
 - 계좌번호 랜덤 생성
 - 잔액 0워 초기화
- 2. 입금
 - 입금 금액 입력 받기
 - 잔액 변경
- 3. 출금
 - 출금 금액 입력 받기
 - 액수 < 0, 잔고 < 0 경우 재입력
 - 잔액 변경
- 4. 잔액조회
 - 비밀번호 확인
 - 계좌 정보 출력
- 5. 삭제
 - 계좌 정보 초기화
- 10. 프로그램 종료

```
//구조체 멤버 변수 ver.
#include <iostream>
using namespace std;
struct ACT {
 unsigned long long ACTID://계좌번호
 char pw[7]://비밀번호
 char* name;//이름
 long long balance://잔액
void menu(int& choice);
void create(ACT& AC);//개설
void deletion(ACT& AC);//삭제
void deposit(ACT& AC);//입금
void withdraw(ACT& AC);//출금
void showData(ACT& AC);//잔액조회
bool checkpw(ACT& AC);//비밀번호 확인
bool cinclear();
void cinput(int& value);
int main() {
 srand(time(NULL));
 int select = 0;
 ACT account = { 0 };
 deletion(account);
 while (1) {
  system("cls");
  menu(select);
  switch (select) {
  case 1: create(account); break;
  case 2: deposit(account); break;
  case 3: withdraw(account); break;
  case 4: showData(account); break;
  case 5: deletion(account); break;
   cout << "\t [ 프로그램 종료. ] \n";
   exit(0);
  system("pause");
 return 0;
void menu(int& choice) {
 cout << "\t [ 계좌 관리 프로그램 ] \n\n";
 cout << "\t 1. 개설 \n";
 cout << "\t 2. 입금 \n";
 cout << "\t 3. 출금 \n";
 cout << "\t 4. 잔액조회 \n";
 cout << "\t 5. 삭제 \n";
 cout << "\t 10. 프로그램 종료 \n";
 cout << "\t 메뉴를 입력하세요.:";
 cin >> choice;
```

```
cinclear();
                                                                  } while (money < 0);</pre>
                                                                  AC.balance += money;
void create(ACT& AC) {
                                                                  cout << money << "원 입금완료.\n";
 deletion(AC);
                                                                  cout << AC.name << " 회원님의 잔액은 " << AC.balance << "원 입니
 char name[255], pw[10];
 cout << "이름을 입력하세요.:";
 cin >> name;
                                                                void withdraw(ACT& AC) {
 cinclear();
                                                                 long long money;
 AC.name = new char[strlen(name) + 1];
 if (AC.name == NULL) exit(1);
                                                                  cout << "출금금액 입력 : ";
 strcpy_s(AC.name, strlen(name) + 1, name);
                                                                   cin >> money;
                                                                  if (money < 0 | AC.balance - money < 0) {
 do {
  cout << "새 비밀번호를 입력하세요. (6자리):";
                                                                    cout << "잔액이 부족합니다. \n";
  cin >> pw;
                                                                    cout << AC.name << " 회원님의 잔액은 " << AC.balance << "원 입
  for (int i = 0; i < 6; i++) {
                                                                 니다.\n";
   if (pw[i] < 0' | | 9' < pw[i] | strlen(pw) != 6) {
                                                                    cin.setstate(cin.failbit);
    cin.setstate(cin.failbit);
                                                                  } while (cinclear());
    break;
                                                                  AC.balance -= money;
                                                                  cout << money << "원 출금완료.\n";
   else AC.pw[i] = pw[i];
                                                                  cout << AC.name << " 회원님의 잔액은 " << AC.balance << "원 입니
  AC.pw[6] = '\0';
 } while (cinclear());
 AC.ACTID = rand() * rand() * rand();//계좌번호
                                                                bool checkpw(ACT& AC) {
 AC.balance = 0;
                                                                  char pw[7];
 cout << "계좌번호: " << AC.ACTID << endl;
 cout << AC.name << "회원님의 잔액은 " << AC.balance << "원 입니
                                                                   cout << "6자리 비밀번호를 입력하세요. (\"End\" : 종료) :";
다.\n";
                                                                   cin >> pw;
                                                                   for (int i = 0; i < 6; i++) {
void deletion(ACT& AC) {
                                                                    if (AC.pw[i] != pw[i]) {
if (AC.name != NULL) delete[] AC.name;
                                                                     cin.setstate(cin.failbit);
 AC.ACTID = 0;
                                                                     break;
 AC.name = NULL;
 for (int i = 0; i < 6; i++) AC.pw[i] = '0';
 AC.balance = 0;
                                                                   if (cinclear()) {
                                                                    cout << "비밀번호가 다름니다. 다시 입력해 주세요. \n\n";
void showData(ACT& AC) {
 if (AC.name == NULL) {
  cout << "등록된 계좌가 없습니다. \n\n";
                                                                  else return true;
  return;
                                                                  } while (strcmp(pw, "End"));
                                                                  return false;
 if (checkpw(AC)) {
  cout << "계좌번호: " << AC.ACTID << endl;
                                                                bool cinclear() {
  cout << AC.name << "회원님의 잔액은 "<< AC.balance << "원 입니 if(cin.fail()) {
다.\n";
                                                                   cin.clear();
                                                                   while (cin.get() != '\n');
                                                                   return true;
void deposit(ACT& AC) {
long long money;
                                                                  else while (cin.get() != '\n');
 cout << "입금금액 입력:";
                                                                  return false:
  cin >> money;
  cinclear();
```

```
//ACT.cpp
//ACT.h
                       #include <iostream>
#pragma once
                       #include <cstring>
class ACT {
                       #include "psk.h"
private:
                       using namespace std;
                       void ACT::Init_act(int _actid, const char* _PW,
 int ACTId;
                       const char* _NAME, int _BAL) {
//계좌번호
                        ACTId = _actid;
 char pw[5];
                        strcpy_s(pw,strlen(_PW)+1, _PW);
//비밀번호
                        strcpy_s(name, strlen(_NAME) + 1, _NAME);
                        bal = _BAL;
 char name[20];
//이름
                       void ACT::showData() {
 int bal;
                        cout << "\n계좌번호: " << ACTId << endl;
                        cout << name << "회원님의 잔액은 " << bal << "입
//잔액
                       니다.\n";
public:
 void Init_act(int
                       void ACT::deposit() {
actid. const char*
                        int m;
                        cout << "\n입금금액 입력:";
pw, const char*
                        cin >> m;
name, int bal);
                        bal += m;
 void showData();
                        cout << "\t\t" << m << "원 입금완료.\n";
//계좌정보 출력
                       void ACT::withdraw() {
 void deposit(); //
                        int m;
입금 함수 & 출력
                        cout << "\n출금금액 입력:";
 void
                        cin >> m;
                        bal -= m:
withdraw();//출금
                        cout << "\t\t" << m << "원 출금완료.\n";
함수 & 출력
};
```

```
//main.cpp
#include <iostream>
#include "psk.h"
using namespace std;
int main() {
 ACT act_1;
 act_1.Init_act(1234, "5678", "christopher", 10000);
 act_1.showData();
                    //잔액조회 멤버 함수 호출
 act_1.deposit();
                   //입금 멤버 함수 호출
                    //출금 멤버 함수 호출
 act_1.withdraw();
 act_1.showData();
                    //잔액조회 멤버 함수호출
 return 0;
```

```
#include <iostream>
using namespace std;
class Triangle{
public:
 int base, height; //밑변, 높이
 void SetTriangle(int _base, int _height);
 double GetArea();
};
void Triangle::SetTriangle(int _base, int _height){
 base = _base;
 height = _height;
double Triangle::GetArea(){
 return base * height / 2.0;
int main(){
 cout << "밑변:";
 int base;
 cin >> base;
 cout << "높이 : ";
 int height;
 cin >> height;
 Triangle t; //삼각형 클래스
 t.SetTriangle(base, height);
 cout << "삼각형의 넓이는 " << t.GetArea() << "입니다.\n";
```

도형 그리기 코드 작성

```
class figure {
public:
    int width, height; //폭(가로), 높이(세로)
    char shape; // 출력 모양
    void Setfigure(int _width, int _height, char _shape); // 멤버 값 초기화
    void print_rectangular(); // 사각형 그리기
    void print_right_triangle(); // 오른쪽 직각 삼각형 그리기
    void print_left_triangle(); // 왼쪽 직각 삼각형 그리기
    void print_isosceles_triangle(); // 이등변 삼각형 그리기
};
```

- 1. class 작성
- 2. 폭(가로), 높이(세로), 출력모양 입력 받기
- 3. 각 멤버 변수의 값으로 도형 그리기
- 4. 사각형 그리기
- 5. 오른쪽 직각 삼각형 그리기
- 6. 왼쪽 직각 삼각형 그리기
- 7. 이등변 삼각형 그리기

```
cout << endl;
#include <iostream>
#include <cmath>
                                         void figure::print_right_triangle(){
                                           for (int i = 0; i < height; i++) {
using namespace std;
                                                                                     cout << endl << endl;
class figure {
                                            for (int j = 0; j < (i / (height / width));}
                                         j++) {
                                                                                   int main() {
public:
                                                                                     cout << "폭(가로): ";
 float width, height; //폭(가로), 높이(세
                                             cout << shape;
로)
                                                                                     float width;
 char shape;
                                            cout << endl;
                                                                                     cin >> width;
                                                                                     cout << "높이(세로): ";
 void Setfigure(int _width, int _height,
char _shape);
                                           cout << endl << endl;
                                                                                     float height;
 void print_rectangular();
                                                                                     cin >> height;
 void print_right_triangle();
                                                                                     cout << "출력 모양: ";
                                         void figure::print_left_triangle(){
 void print_left_triangle();
                                           for (int i = 0; i < height; i++) {
                                                                                     char shape;
 void print_isosceles_triangle();
                                            for (int j = 0; j < (width - ceil(i / 
                                                                                     cin >> shape;
                                         (height / width))); j++) cout << " ";
                                                                                     figure t; //도형 클래스
void figure::Setfigure(int_width, int
                                            for (int j = 0; j < (i / (height / width)); t.Setfigure(width, height, shape);
_height, char _shape) {
                                         j++) cout << shape;
                                                                                     t.print_rectangular();
 width = width;
                                            cout << endl;
                                                                                     t.print_right_triangle();
 height = _height;
                                                                                     t.print_left_triangle();
 shape = _shape;
                                           cout << endl << endl;
                                                                                     t.print_isosceles_triangle();
                                                                                     return 0;
void figure::print_rectangular(){
                                         void figure::print_isosceles_triangle(){}
                                           for (int i = 0; i < height; i++) {
 for (int i = 0; i < height; i++) {
  for (int j = 0; j < width; j++) {
                                            for (int j = 0; j < ceil((width/2 - ceil(i
                                         / (height / width)/2))); j++) cout << " ";
    cout << shape;
                                            for (int j = 0; j < (i / (height / width))
  cout << endl;
                                         / 2 ); j++) cout << shape;
                                            for (int j = 0; j < trunc(i / (height /
                                         width) / 2.0001); j++) cout << shape;
 cout << endl << endl;
```

Day06 - Contents

- 1. 클래스 맴버의 접근제어
- 2. OOP (Object Oriented Programming)
- 3. Encapsulation & Information hiding
- 4. 예제 코드

Day06 - 1. 클래스 맴버의 접근제어

```
private: 클래스 내에서만 접근허용
protected: 상속관계에서 접근허용
public: 어디서든 접근허용
                class point {
형식(선언)
                private:
class 클래스명{
                 int x, y; //x좌표, y좌표
private:
                public:
 변수1;
                 bool init_pos(int x_pos, int y_pos);
 변수2;
                 int get_x();
                 int get_y();
protected:
                 bool set_x(int x_pos);
public:
                 bool set_y(int y_pos);
변수3;
 변수 연산 함수1;
                class rectangle {
 변수 연산 함수2;
                private:
 변수 연산 함수3;
                 point up_left, low_right;//왼쪽 상당 좌표 1, 오른쪽 하단 좌표 2
                public:
                 bool init_rectangle(point& ul, point& lr);
                 void show_rectangle();
                };
```

```
클래스 코드 작성
class point {
private:
 int x, y; //x좌표, y좌표
public:
 bool init_pos(int x_pos, int y_pos); // x, y 값 초기화
 int get_x(); // x값 읽기
 int get_y(); // y값 읽기
 bool set_x(int x_pos); // x값 쓰기
 bool set_y(int y_pos); // y값 쓰기
class rectangle {
private:
  point up_left, low_right;//왼쪽 상당 좌표 1, 오른쪽 하단 좌표 2
  point diff;//low_right - up_left 차이 값
public:
  bool init_rectangle(point& ul, point& lr); // up_left, low_right 초기화
  void subtract_point();// diff = low_right - up_left 차이 값 구하는 함수
  void show_rectangle();//두 점 사이 사각형 그리기
};
```

Day06 - 2. OOP (Object Oriented Programming)

```
객체지향 패러다임
현실에 존재하는 사물과 대상, 그 대상의 행동을 실체화 하는 프로그래밍
세상을 객체로 바라보는 사고를 말한다.
객체 : 1개 이상의 데이터(자료)와 1개 이상의 기능(메소드:함수)
클래스(객체의 틀)
- 목적에 따라 대상을 설정하고 추상적인 개념을 수치화로 정의
- 수치화된 데이터의 연산을 메소드(함수)로 처리
class point {
                          변수(객체) 선언
private:
                          point pos_1, pos_2; // 객체 생성
int x, y; //x 좌 亚, y 좌 亚
                          point * ptr_pos_1 = new point; //동적할당
public:
bool init_pos(int x_pos, int y_pos);
int get_x();
int get_v();
bool set_x(int x_pos);
bool set_v(int y_pos);
```

Day06 - 3. Encapsulation & Information hiding

```
캡슐화 (Encapsulation)
객체의 속성(data)과 행위(methods)를 하나로 묶고 (클래스 설계),
실제 구현 내용을 외부에 숨기는 것
```

특징

- 외부 객체는 객체 내부의 구조를 얻지 못한다
- 객체가 노출해서 제공하는 필드와 메소드만 이용할 수 있다
- 외부의 잘못된 사용으로 인해 객체가 손상되지 않도록 캡슐화하여 보호
- 은닉과 노출은 접근제어 지시자로 분류
- 클래스의 구성 범위를 어디까지 정의할 것인가?

```
정보은닉 (Information hiding) - [private:] 클래스 멤버 변수를 외부에서 직접 수정 할 수 없으며,
멤버 함수로만 수정 하도록 만드는 방법으로 외부에서 접근을 막을 수 있다
```

```
접근제어 지시자(은닉 정도를 분류)
은닉이 필요한 속성(data)는 [ private : ]
상속이 필요한 경우 [ protected : ]
노출하는 모든 경우 [ public : ]
```

```
Account Class 작성
class ACT {
private:
 unsigned long long ACTID;//계좌번호
 char pw[7];//비밀번호
 char* name;//이름
 long long balance;//잔액
 bool cinclear();//cin 초기화
public:
 void init();//초기화
 void create();//개설
 void deletion();//삭제
 void deposit();//입금
 void withdraw();//출금
 void showData();//잔액조회
 bool checkpw();//비밀번호 확인
void menu(int& choice);
//메뉴 출력 & 입력 받기
bool cinclear();//cin 초기화
```

[계좌 관리 프로그램]

- 1. 개설
 - 이름 입력 받기
 - 비밀번호 6자리 입력 받기
 - 계좌번호 랜덤 생성
 - 잔액 0원 초기화
- 2. 입금
 - 입금 금액 입력 받기
 - 잔액 변경
- 3. 출금
 - 출금 금액 입력 받기
 - if(액수 < 0, 잔고 < 0 경우) 재입력
 - 잔액 변경
- 4. 잔액조회
 - 비밀번호 확인
 - 계좌 정보 출력
- 5. 삭제
 - 계좌 정보 초기화
- 10. 프로그램 종료

```
//class 멤버 변수 + 멤버 함수 선언ver.
#include <iostream>
using namespace std;
void menu(int& choice);
bool cinclear();
void cinput(int& value);
class ACT {
private:
 unsigned long long ACTID;//계좌번호
 char pw[7];//비밀번호
 char* name://이름
 long long balance://잔액
 bool cinclear()://cin 초기화
 void init();//초기화
 void create();//개설
 void deletion();//삭제
 void deposit()://입금
 void withdraw();//출금
 void showData();//잔액조회
 bool checkpw();//비밀번호 확인
int main() {
 srand(time(NULL));
 int select = 0;
 ACT account;
 account.init();
 while (1) {
  system("cls");
  menu(select);
  switch (select) {
  case 1: account.create(); break;
  case 2: account.deposit(); break;
  case 3: account.withdraw(); break;
  case 4: account.showData(); break;
  case 5: account.deletion(); break;
   cout << "\n\t [ 프로그램 종료. ] \n";
   exit(0)
  system("pause");
 return 0;
void menu(int& choice) {
 cout << "\t [ 계좌 관리 프로그램 ] \n\n";
 cout << "\t 1. 개설 \n";
 cout << "\t 2. 입금 \n";
 cout << "\t 3. 출금 \n";
 cout << "\t 4. 잔액조회 \n";
 cout << "\t 5. 삭제 \n";
 cout << "\t 10. 프로그램 종료 \n";
 cout << "\t 메뉴를 입력하세요.: ";
 cin >> choice;
 cinclear();
void ACT::init() {
 ACTID = 0;
 for (int i = 0; i < 7; i++) pw[i] = 0;
 name = NULL;
 balance = 0;
void ACT∷create() {
```

```
deletion();
char NAME[255], PW[10];
cout << "이름을 입력하세요. :";
cin >> NAME;
cinclear();
name = new char[strlen(NAME) + 1];
if (name == NULL) exit(1);
strcpy_s(name, strlen(NAME) + 1, NAME);
 cout << "새 비밀번호를 입력하세요. (6자리):";
  cin >> PW;
  for (int i = 0; i < 6; i++) {
   if (PW[i] < '0' || '9' < PW[i] || strlen(PW) != 6) {
    cin.setstate(cin.failbit);
    break;
   else pw[i] = PW[i];
 pw[6] = '\0';
} while (cinclear());
ACTID = rand() * rand() * rand();//계좌번호
cout << "계좌번호 : " << ACTID << endl;
cout << name << "회원님의 잔액은 " << balance << "원 입니다.\n";
void ACT::deletion() {
if (name != NULL) delete[] name;
ACTID = 0;
name = NULL;
for (int i = 0; i < 6; i++) pw[i] = '0';
cout << "계좌를 삭제 했습니다. \n\n";
void ACT::showData(){
if (name == NULL) {
 cout << "등록된 계좌가 없습니다. \n\n";
if (checkpw()) {
 cout << "계좌번호: " << ACTID << endl;
  cout << name << " 회원님의 잔액은 " << balance << " 원 입니다.\n";
void ACT::deposit() {
if (name == NULL) {
 cout << "등록된 계좌가 없습니다. \n\n";
 cout << "입금금액 입력:";
long long money;
do {
 cin >> money;
 cinclear();
} while (money < 0);
 balance += money;
cout << money << " 원 입금완료.\n";
cout << name << "회원님의 잔액은" << balance << "원 입니다.\n";
void ACT::withdraw() {
if (name == NULL) {
 cout << "등록된 계좌가 없습니다. \n\n";
 return;
```

```
long long money;
 do {
  cout << "출금금액 입력:";
  cin >> money;
  if (money < 0 || balance - money < 0) {
   cout << "잔액이 부족합니다. \n";
   cout << name << "회원님의 잔액은 " << balance << "원 입니다.\n";
   cin.setstate(cin.failbit);
 } while (cinclear());
 balance -= money;
 cout << money << " 원 출금완료.\n";
 cout << name << " 회원님의 잔액은 " << balance << "원입니다.\n";
bool ACT::checkpw() {
 char PW[7];
  cout << "6자리 비밀번호를 입력하세요. (\"End\": 종료):";
  cin >> PW;
  for (int i = 0; i < 6; i++) {
   if (pw[i] != PW[i]) {
    cin.setstate(cin.failbit);
    break;
  if (cinclear()) {
   cout << "비밀번호가 다름니다. 다시 입력해 주세요. \n\n";
   continue;
  else return true;
 } while (strcmp(PW, "End"));
 return false;
bool ACT::cinclear() {
 if (cin.fail()) {
  cin.clear();
  while (cin.get() != '\n');
  return true;
 while (cin.get() != '\n');
 return false;
bool cinclear() {
 if (cin.fail()) {
  cin.clear();
  while (cin.get() != '\n');
  return true;
 while (cin.get() != '\n');
 return false;
```

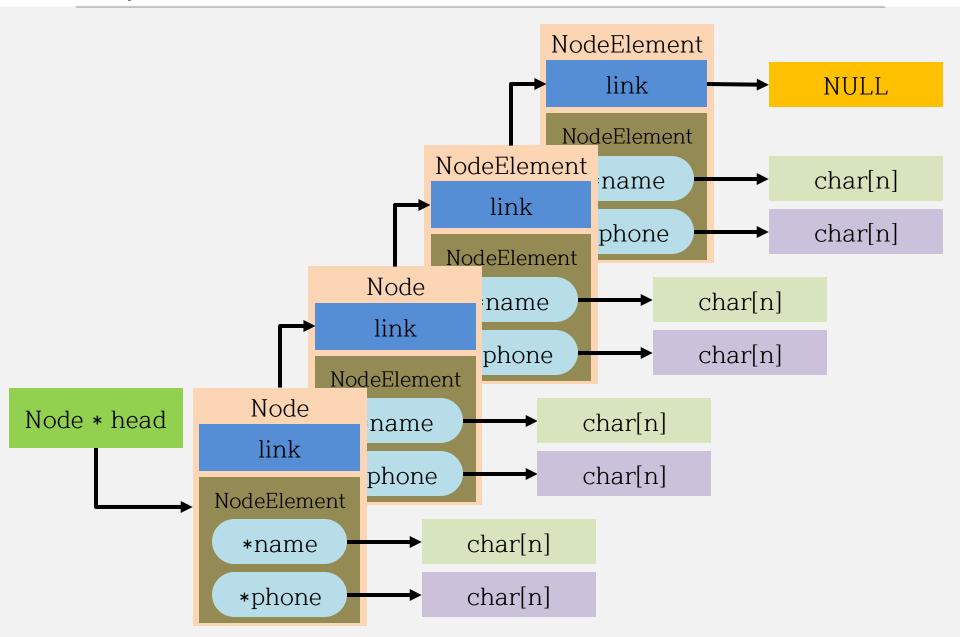
```
[ 네모 그리기 ]
1. point 1 (X,Y) 입력 받기
                                                    point 1 ( X , Y )
2. point 2 (X,Y) 입력 받기
3. 점 1과 2의 내부 면적 채우기
rectangle Class 작성
class point {
private:
  int x, y; //x좌표, y좌표
public:
  bool init_pos(int x_pos, int y_pos); // x, y 값 초기화
  int get_x(); // x값 읽기
  int get_y(); // y값 읽기
  bool set_x(int x_pos); // x값 쓰기
  bool set_y(int y_pos); // y값 쓰기
};
class rectangle {
private:
  point up_left, low_right;//왼쪽 상당 좌표 1, 오른쪽 하단 좌표 2
  point diff;//low_right - up_left 차이 값
public:
  bool init_rectangle(point& ul, point& lr); // up_left, low_right 초기화
  void subtract_point();// diff = low_right - up_left 차이 값 구하는 함수
  void show_rectangle();//두 점 사이 사각형 그리기
};
```

point 2 (X, Y)

```
#include <iostream>
                                          cout << "2번 좌표를 입력하세요 (x y, void rectangle::subtract_point()
                                                                                                                          return false;
                                       공백 구분, 범위:0~500, 종료:-1): ";
using namespace std;
class point {
                                          cin >> x >> y;
                                                                                 diff.set_x(low_right.get_x() -
                                                                                                                         x = x_pos;
private:
                                          if (x == -1 || y == -1) break;
                                                                                up_left.get_x());
                                                                                                                         y = y_pos;
int x, y; //x좌표, y좌표
                                          point pos2;
                                                                                 diff.set_y(up_left.get_y() -
                                                                                                                         return true;
                                          if (!pos2.init_pos(x, y)) {
                                                                                low_right.get_v());
public:
 bool init_pos(int x_pos, int y_pos); //
                                           cout << "pos2 초기화 실패" << endl; cout << diff.get_x() << ", "
                                                                                                                       int point∷get_x() {
x, y 값 초기화
                                           system("pause");
                                                                                << diff.get_v() << endl;
                                                                                                                         return x;
 int get_x(); // x값 읽기
                                           continue;
 int get_y(); // y값 읽기
                                                                                void rectangle::show_rectangle() {
                                                                                                                       int point::get_v() {
 bool set_x(int x_pos); // x값 쓰기
                                          rectangle rec1;
                                                                                 cout << endl << endl << endl
                                                                                                                        return y;
 bool set_y(int y_pos); // y값 쓰기
                                          if (!rec1.init_rectangle(pos1, pos2))
                                                                               << endl:
                                       cout << "rec1 초기화 실패" << endl;
                                                                                 for (int i = 0; i < up_left.get_x(); i++)
                                                                                                                       bool point::set_x(int x_pos) {
class rectangle {
                                          rec1.subtract_point();
                                                                                cout << " ";
                                                                                                                         if (0 > x_pos | 500 < x_pos) 
                                          rec1.show_rectangle();
                                                                                 cout << up_left.get_x() << ","
                                                                                                                          cout << "X좌표 범위를 벗어났습니
private:
 point up_left, low_right;//왼쪽 상당 좌
                                          system("pause");
                                                                                << up_left.get_y() << endl;
                                                                                                                        다.(0~500)" << endl;
표 1, 오른쪽 하단 좌표 2
                                                                                for (int i = 0; i < up_left.get_y() -
                                                                                                                          return false;
 point diff;//low_right - up_left 차이
                                         return 0;
                                                                                low right.get v(); i++) {
값
                                                                                  for (int i = 0; i < up left.get x(); i++)
                                                                                                                        x = x_pos;
                                                                               cout << " ";
public:
                                        bool rectangle::init_rectangle(point&
                                                                                                                         return true;
 bool init_rectangle(point& ul, point& ul, point& lr) {
                                                                                  for (int i = 0; i < low right.get x() -
                                         if ((ul.get_x() == lr.get_x()) || (ul.get_y()up_left.get_x(); i++) cout << "\equiv ";
lr); // up_left, low_right 초기화
                                                                                                                        bool point::set_y(int y_pos) {
 void subtract_point();// diff =
                                        == lr.get v())) return false;
                                                                                  cout << endl;
                                                                                                                         if (0 > y_pos || 500 < y_pos) {
low_right - up_left 차이 값 구하는 함수
                                                                                                                          cout << "Y좌표 범위를 벗어났습니
                                         if (ul.get_x() > lr.get_x()) {
 void show_rectangle();//두 점 사이 사
                                          low_right.set_x(ul.get_x());
                                                                                 for (int i = 0; i < low_right.get_x(); i++) \Box . (0~500)" << endl;
각형 그리기
                                          up_left.set_x(lr.get_x());
                                                                                cout << " ";
                                                                                                                          return false;
};
                                                                                 cout << low_right.get_x() << ","
                                                                               << low_right.get_y() << endl;
int main() {
                                         else {
                                                                                                                         y = y_pos;
                                          up_left.set_x(ul.get_x());
                                                                                 for (int i = 0; i < low_right.get_y() - 1; return true;
 while (1) {
                                                                                i++) cout << endl;
  system("cls");
                                          low_right.set_x(lr.get_x());
  cout << "1번 좌표를 입력하세요 (x y,
공백 구분, 범위:0~500, 종료:-1): ";
                                         if (ul.get_y() > lr.get_y()) {
                                                                                bool point::init_pos(int x_pos, int
  int x, y;
                                          up_left.set_y(ul.get_y());
                                                                               y_pos) {
                                                                                if (0 > x_pos | | 500 < x_pos) {
  cin >> x >> v;
                                          low_right.set_y(lr.get_y());
  if (x == -1 || y == -1) break;
                                                                                  cout << "X좌표 범위를 벗어났습니
                                                                                다.(0~500)" << endl;
  point pos1;
                                         else {
                                          low_right.set_y(ul.get_y());
  if (!pos1.init_pos(x, y)) {
                                                                                  return false;
   cout << "pos1 초기화 실패" << endl;
                                          up_left.set_y(lr.get_y());
                                                                                 if (0 > y_pos || 500 < y_pos) {
   system("pause");
                                                                                  cout << "X좌표 범위를 벗어났습니
   continue;
                                         return true;
                                                                                다.(0~500)" << endl;
```

```
[명단 관리 프로그램] [Directory Program]
1. 등록
                     1. REGIST
2. 삭제
                     2. REMOVE
                                           typedef student* NodeElement;
3. 리스트 출력
                     3. DISPLAY
                                           class Node {
4. 전체 삭제
             4. REMOVEALL
                                           private:
0. 프로그램 종료
                     0. TERMINATE
                                            NodeElement value:
메뉴 선택:
                     Select Menu:
                                            Node* link;
                                           public:
                                            void init();
                                            void Delete();
                                            inline NodeElement getValue();
                                            inline Node* getlink();
                                            inline void setValue(NodeElement data);
                                            inline void setlink(Node* _link);
                                           };
class student {
                                           class SingleList {
private:
 char* name;
                                           private:
                                            Node* head;
 char* phone;
public:
                                           public:
 inline void init();
                                            inline void init() { head = NULL; }
 inline void Delete();
                                            inline void Delete();
 inline char* getName();
                                            bool InsertNode(NodeElement data);
 inline char* getPhone();
                                            bool DeleteNode(NodeElement data);
 inline bool setName(char* _name);
                                            bool DeleteAll();
 inline bool setPhone(char* _phone);
                                            bool PrintList();
};
                                           };
```

Day06 - 4. 예제 코드



Header.H

```
#pragma once
#ifndef _REGISTER_H
                                                                                                       head = NewNode;
#define _REGISTER_H
                                                  typedef student* NodeElement;
                                                                                                       return true;
#include<iostream>
using namespace std;
                                                  class Node {
void cinclear() {
                                                  private:
if (cin.fail()) {
                                                    NodeElement value:
                                                                                                      if (head == NULL) {
  cin.clear();
                                                    Node* link;
  while (cin.get() != '\n');
                                                   public:
                                                                                                        return false;
                                                    void init() {
 while (cin.get() != '\n');
                                                     value = new student;
                                                     value->init();
void cinput(int& value) {
                                                     link = NULL;
 while (!(cin >> value)) {
  cin.clear();
                                                    void Delete() {
  while (cin.get() != '\n');
                                                     if (value != NULL) {
  cout << "\t 다시 입력 하세요.: ";
                                                      value->Delete();
                                                                                                     >getlink();
 while (cin.get() != '\n');
                                                     delete value;
                                                     value = NULL;
enum menuType {
                                                     link = NULL;
 TERMINATE, REGISTER, REMOVE, DISPLAY,
REMOVEALL
                                                    inline NodeElement getValue() { return value; }
                                                                                                         return true;
                                                    inline Node* getlink() { return link; }
                                                    inline void setValue(NodeElement_data) { value =
class student {
private:
                                                    inline void setlink(Node* link) { link = link; }
int age;
 char* name;
 char* phone;
                                                                                                       return false;
                                                  class SingleList {
 int no;
                                                   private:
public:
                                                   Node* head;
 inline void init() { name = phone = NULL; }
                                                   public:
 inline void Delete() {
                                                    inline void init() { head = NULL; }
                                                    inline void Delete() { delete head; head = NULL; } (Y/N) \n\n";
  if (name != NULL) delete[] name;
  if (phone != NULL) delete[] phone;
                                                    bool InsertNode(NodeElement data);
                                                    bool DeleteNode(NodeElement data);
                                                                                                       char answer = 'n';
 inline char* getName() { return name; }
                                                    bool DeleteAll();
                                                                                                       cin >> answer;
 inline char* getPhone() { return phone; }
                                                    bool PrintList();
                                                                                                       cinclear();
 inline bool setName(char* _name) {
  if ( name == NULL) return false;
                                                  void PrintMenu(int& menu);
                                                                                                      pNode != NULL;) {
  name = _name;
  return true;
 inline bool setPhone(char* _phone) {
                                                   bool SingleList::InsertNode(NodeElement data) {
                                                    Node* NewNode = new Node;
  if (_phone == NULL) return false;
                                                    if (NewNode == NULL) return false;
  phone = phone;
                                                    NewNode->init();
  return true;
                                                    NewNode->setValue(data);
                                                                                                        head = NULL;
```

```
NewNode->setlink(head);
                                                  cout << "\t\t [ 목록을 모두 지웠습니다.]\n\n";
                                                  return true;
                                                else {
                                                  cout << "\t\t [ 목록을 지우지 않고 메뉴로 돌아갑니
bool SingleList::DeleteNode(NodeElement data) { 다. ]\n\n";
                                                  return false;
  cout << "\t\t목록을 추가해 주세요. \n\n";
 Node* DeleteNode = head, * preDelete = NULL;
                                               bool SingleList::PrintList() {
 while (DeleteNode != NULL) {
                                                if (head == NULL) {
  if ((strcmp(DeleteNode->getValue()->getName(),
                                                 cout << "\t\t목록이 비어있습니다.\n\n";
data->getName()) == 0) && (strcmp(DeleteNode-
                                                  return false;
>getValue()->getPhone(), data->getPhone()) == 0))
                                                 for (Node* pNode = head; pNode != NULL; pNode
                                               = pNode->getlink()) {
   if (preDelete == NULL) head = DeleteNode-
                                                  cout << endl;
   else preDelete->setlink(DeleteNode->getlink()); cout << "\t\t 이름:" << pNode->getValue()-
   DeleteNode->Delete();
                                               >getName() << endl;
   delete DeleteNode;
                                                 cout << "\t\t 전화번호: " << pNode->getValue()-
   cout << "\t\t 삭제 성공! \n\n";
                                               >getPhone() << endl;
                                                return true;
  preDelete = DeleteNode;
  DeleteNode = DeleteNode->getlink();
                                               void PrintMenu(int& menu) {
 cout << "\t\t목록에 삭제 할 자료가 없습니다. \n\n"; cout << "\n\n\t\t[ 명단 관리 프로그램 \\n\n";
                                                cout << "\t\t1. 등록\n";
                                                cout << "\t\t2. 삭제\n";
                                                cout << "\t\t3. 리스트 출력\n";
bool SingleList::DeleteAll() {
                                                cout << "\t\t4. 전체 삭제\n";
 cout << endl << endl;
                                                cout << "\t\t0. 프로그램 종료\n\n";
 cout << "\t\t [ 목록을 모두 지우시겠습니까??
                                                cout << "\t\t메뉴 선택:";
                                                cinput(menu);
 cout << "\t\t 입력하세요. :";
                                               #endif // _REGISTER_H
 if (answer == 'Y' || answer == 'y') {
  for (Node* pNode = head. *DeleteNode = NULL;
   DeleteNode = pNode;
   pNode = pNode->getlink();
   DeleteNode->Delete();
   delete DeleteNode;
   cout << "\t\t 삭제 성공! \n\n";
```

Main.cpp

```
#include <iostream>
#include "register.h"
using namespace std;
int main() {
 int menu = 1;
 char buf[255];
 SingleList list1;
 list1.init();
 while (1) {
  system("cls");
  PrintMenu(menu);
  switch (menu) {
  case REGISTER:
   cout << "\t\t 이름 : ";
   cin >> buf;
   cinclear();
   NodeElement Register = new student;
   Register->init();
   Register->setName(new char[strlen(buf) + 1]);
   strcpy_s(Register->getName(), strlen(buf) + 1, buf);
   cout << "\t\t 전화번호: ";
   cin >> buf;
   cinclear();
   Register->setPhone(new char[strlen(buf) + 1]);
   strcpy_s(Register->getPhone(), strlen(buf) + 1, buf);
   list1.InsertNode(Register);
   list1.PrintList();
   break;
  case REMOVE:
   cout << "\t\t
                   이름:";
   cin >> buf:
```

```
cinclear();
  NodeElement Deletion = new student;
  Deletion->init();
  Deletion->setName(new char[strlen(buf) + 1]);
  strcpy_s(Deletion->getName(), strlen(buf) + 1, buf);
  cout << "\t\t 전화번호: ";
  cin >> buf;
  cinclear();
  Deletion->setPhone(new char[strlen(buf) + 1]);
  strcpy_s(Deletion->getPhone(), strlen(buf) + 1, buf);
  list1.DeleteNode(Deletion);
  list1.PrintList();
  Deletion->Delete();
  delete Deletion;
  break:
 case DISPLAY: list1.PrintList(); break;
 case REMOVEALL:
  list1.DeleteAll();
  list1.PrintList();
  break:
 case TERMINATE:
  cout << "\n\n\t\t[ 프로그램 종료 합니다. ]\n\n";
  exit(0);
 cout << endl << endl;
 system("pause");
return 0;
```

Day07 - Contents

- 1. Constructor & Destructor
- 2. 예제 코드

Day07 - 1. Constructor & Destructor

```
생성자 (Constructor)
객체의 생성과 동시에 호출하는 함수
클래스의 이름과 동일한 함수의 이름
소멸자 (Destructor)
객체의 소멸과 동시에 호출 되는 함수
'~' + 클래스의 이름과 동일한 함수의 이름
```

특징

- 함수 선언에 반환형이 없고, 실제 반환도 없음.
- 객체 생성시 딱 한번 호출
- 오버로딩(Overloading) 가능
- Default 가능
- 클래스의 각 멤버가 선언된 순서대로 초기화 된다.
- 생성자 뒤에 콜론을 사용해서 각 멤버를 초기화 할 수 있다.(일반 멤버함수에서는 사용 할 수 없다.)
- 생성자, 소멸자는 정의하지 않으면 자동으로 생성된다.
- 디폴트 생성자, 소멸자는 빈 껍데기만 존재한다.

Day07 - 1. Constructor & Destructor

```
변수(객체) 선언(생성)
point pos_1; // 객체 생성
point * ptr_pos_1 = new point; //동적할당
생성자 변수 선언
point pos_1(1,1); // 객체 생성
point * ptr_pos_1=new point(1,1); //동적할당
일반적인 초기화 방법
A(int _a, double _d, char _c)
a = _a;
d = d;
C = C;
콜론(:)을 이용한 초기화 방법
A(int _a, double _d, char _c) : a(_a), d(_d), c(_c)
```

```
#include <iostream>
using namespace std;
                                                   main() stack memory
class A {
private:
  int a;
                                                                                  6
                                                          소멸:~1
public:
  A(int _a) {
    a = _a;
                                                                                  5
                                                          소멸:~3
    cout << "생성자 " << a << "호출\n";
  ~A() {
                                                           생성:3
    cout << "~소멸자 " << a << "호출\n";
};
                                                    fun() stack memory
void fun() {
  A ob2(2);
                                                                                  3
                                                          소멸:~2
void main() {
  A ob1(1);
                                                           생성 : 2
                                                                                  2
  fun();
  A ob3(3);
                                                           생성:1
```

객체의 유효범위는 지역변수와 같다, 블록이 끝나면 메모리 상에서 소멸됨

```
#include <iostream>
using namespace std;
                                               main() stack memory
class A {
private:
  int* p;
                                                                           6
                                                    소멸:~*p=1
public:
  A(int _a) {
    p = new int;
                                                                           5
                                                    소멸:~*p=3
    *p = _a;
    생성: *p=3
                                                                           4
  ~A() {
    cout << "~소멸자 " << *p << "호출\n";
    delete p;
                                                fun() stack memory
  }};
void fun() {
                                                                           3
                                                    소멸:~*p=2
  A ob2(2);
void main() {
                                                     생성: *p=2
  A ob1(1);
  fun();
  A ob3(3);
                                                     생성 : *p=1
```

객체의 유효범위는 지역변수와 같다, 블록이 끝나면 메모리 상에서 소멸됨

Day07 - 1. Constructor & Destructor

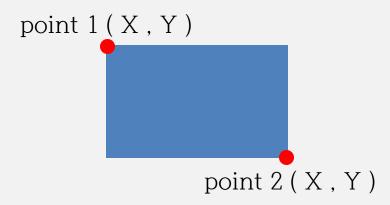
생성자 (Constructor) & 소멸자 (Destructor)

	생성자 (Constructor)	소멸자 (Destructor)	
호출시기	객체 생성 시 1번	객체 소멸 시 1번	
함 수 명	클래스명	~클래스명	
리 턴 값	없다.	없다.	
인 수 (매개변수)	가질 수 있다.	없다.	
	여러 개 존재 가능 (생성자 오버로딩)	오직 한 개만 존재	
활 용	멤버변수 초기화 시	동적 메모리 해제 시	

```
#include <iostream>
                                                   void Triangle::initialize(double _base, double
using namespace std;
                                                   _height) {
class Triangle {
                                                      base = _base;
public:
                                                     height = _height;
  double base, height;
  Triangle() {
                                                   double Triangle::getArea(){
     base = 0;
                                                     return base * height / 2;
     height = 0;
     cout << "1. constructor call\n";</pre>
                                                   int main() {
                                                     double b, h;
  Triangle(double _base, double _height) {
                                                     cout << "밑변 입력:";
     base = base;
                                                     cin >> b;
                                                     cout << "높이 입력:";
     height = _height;
     cout << "2. constructor call with
                                                     cin >> h;
parameter\n";
                                                     Triangle tri; //클래스 변수 선언 + 생성자
                                                     //Triangle tri(b, h); //클래스 생성자 변수 선언
  void initialize();
                                                     tri.initialize(b. h); //초기화
                                                     cout << "삼각형의 넓이는 " << tri.getArea() << "입
  void initialize(double _base, double _height);
                                                   니다.\n";
  double getArea();
};
                                                     return 0;
void Triangle::initialize() {
  base = 0;
  height = 0;
```

[네모 그리기]

- 1. point 1 (X,Y) 입력 받기
- 2. point 2 (X , Y) 입력 받기
- 3. 점 1과 2의 내부 면적 채우기



```
class point {
private:
    int x, y; //x좌표, y좌표
public:
    point(int _x = 0, int _y = 0);
    ~point() { }
    void init_pos(int x_pos, int y_pos); //초기화 함수
    int get_x(); //멤버 변수 x 확인(읽기)
    int get_y(); //멤버 변수 y 확인(읽기)
    void set_x(int x_pos); //멤버 변수 x 설정(쓰기)
    void set_y(int y_pos); //멤버 변수 y 설정(쓰기)
};
```

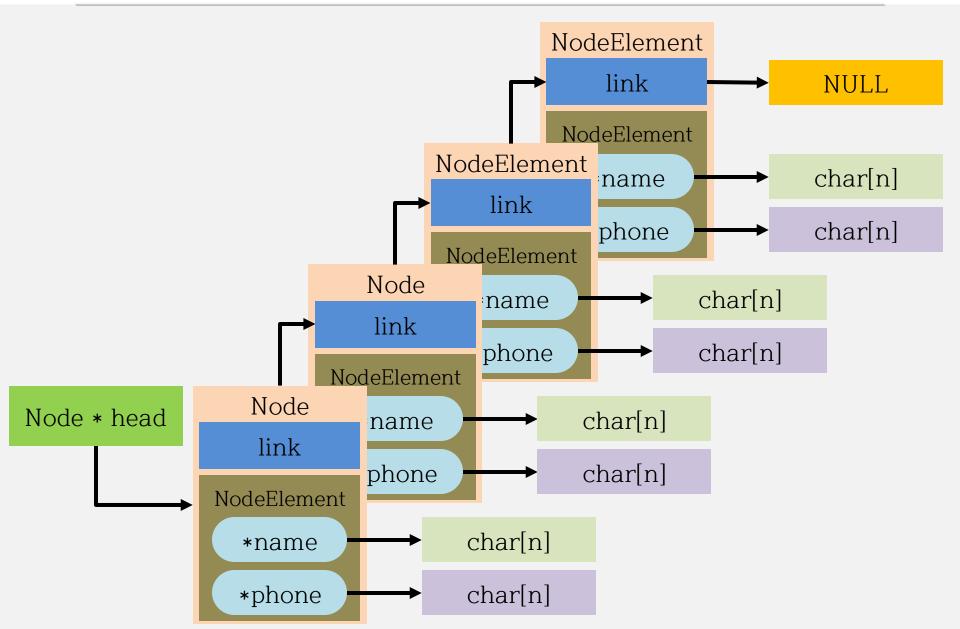
```
Class rectangle {
Private:
  point up_left, low_right;//왼쪽 상단, 오른쪽 하단
Public:
  rectangle(int& x1, int& y1, int& x2, int& y2);
  // 4개 값으로 생성
  rectangle(point& pos1, point& pos2);
  // point 2개로 생성
  ~rectangle() {}
  void init_rec(int& x1, int& y1, int& x2, int& y2);
  void init_rectangle(point& ul, point& lr);
  void show_rectangle();
};
```

```
//두 점으로 사각형 만들기
                                        int& y1, const int& x2, const int& y2);
#include <iostream>
                                           void init_rectangle(point& ul, point&
                                                                                  if (y1 > y2) {
                                                                                                                           for (int i = 0; i < low_right.get_x();
using namespace std;
                                        lr);
                                                                                      up_left.set_v(v1);
                                                                                                                         i++) cout << " ";
                                                                                     low_right.set_y(y2);
                                                                                                                           cout << low_right.get_x() << ","
class point {
                                           void show_rectangle();
private:
                                                                                                                         << low_right.get_y() << endl;
  int x, y; //x좌표, y좌표
                                        int main() {
                                                                                   else {
                                                                                                                           for (int i = 0; i < low_right.get_y() - 1;
public:
                                           while (1) {
                                                                                     low_right.set_v(v1);
                                                                                                                         i++) cout << endl;
  point(int _x = 0, int _y = 0) {
                                             //system("cls");
                                                                                     up_left.set_y(y2);
                                             cout << "1번 좌표를 입력하세요 (x y, }
                                                                                                                        void point∷init_pos(int x_pos. int
     x = _X;
                                        공백 구분, 범위:0~500, 종료:-1): ";
                                                                                                                         y_pos) {
     y = y;
     cout << "point Constructor call\n";</pre>
                                                                                                                           if (0 > x_pos | | 500 < x_pos) {
                                             int x, y;
                                                                                void rectangle::init_rectangle(point&
                                                                                                                             cout << "X좌표 범위를 벗어났습니
                                             cin >> x >> v;
                                                                                ul, point& lr) {
                                             if (x == -1 || v == -1) break;
                                                                                   if (ul.get_x() > lr.get_x()) {
                                                                                                                         다.(0~500)" << endl;
  ~point() {
     cout << "point Destructor call\n";</pre>
                                             point pos1(x, y);
                                                                                     low_right.set_x(ul.get_x());
                                                                                                                              return;
                                             cout << "2번 좌표를 입력하세요 (x y,
                                                                                     up_left.set_x(lr.get_x());
  void init_pos(int x_pos, int y_pos); 공백 구분, 범위:0~500, 종료:-1):";
                                                                                                                           if (0 > y_pos || 500 < y_pos) {
                                                                                                                             cout << "X좌표 범위를 벗어났습니
  int get_x();
                                             cin >> x >> v;
                                                                                   else {
                                                                                                                         다.(0~500)" << endl;
                                             if (x == -1 || v == -1) break;
  int get_y();
                                                                                      up_left.set_x(ul.get_x());
                                                                                     low_right.set_x(lr.get_x());
  void set_x(int x_pos);
                                             point pos2(x, y);
                                                                                                                              return;
  void set_y(int y_pos);
                                             rectangle rec1(pos1, pos2);
                                             rectangle rec2(10, 10, 5, 5);
                                                                                   if (ul.get_y() > lr.get_y()) {
                                                                                                                           x = x_pos;
                                             rec1.show_rectangle();
class rectangle {
                                                                                     up_left.set_y(ul.get_y());
                                                                                                                           y = y_pos;
                                                                                     low_right.set_y(lr.get_y());
private:
  point up_left, low_right;//왼쪽 상당
                                                                                                                         int point::get_x() { return x; }
좌표 1, 오른쪽 하단 좌표 2
                                        << endl;
                                                                                   else {
                                                                                                                         int point::get_y() { return y; }
                                             rec2.show rectangle();
                                                                                     low_right.set_y(ul.get_y());
                                                                                                                        void point::set_x(int x_pos) {
public:
                                             cout << "-----
  rectangle(const int& x1, const int&
                                                                                     up_left.set_y(lr.get_y());
                                                                                                                           if (0 > x_pos | | 500 < x_pos) {
y1, const int& x2, const int& y2) {
                                                                                                                             cout << "X좌표 범위를 벗어났습니
                                                                                                                         다.(0~500)" << endl;
     init_rec(x1, y1, x2, y2);
                                        << endl;
     cout << "rectangle constructor
                                             system("pause");
                                                                                void rectangle::show_rectangle() {
                                                                                                                             return;
call\n";
                                                                                   cout << endl << endl << endl << endl
                                           return 0;
                                                                                << endl;
                                                                                                                           x = x_pos;
  rectangle(point& pos1, point& pos2) }
                                                                                   for (int i = 0; i < up_left.get_x(); i++) }
                                        void rectangle::init_rec(const int& x1, cout << " ";
                                                                                                                         void point::set_y(int y_pos) {
     init_rectangle(pos1, pos2);
                                        const int& y1, const int& x2, const
                                                                                   cout << up_left.get_x() << ","
                                                                                                                           if (0 > y_pos || 500 < y_pos) {
     cout << "rectangle constructor
                                                                                                                             cout << "Y좌표 범위를 벗어났습니
                                        int\& y2){
                                                                                << up_left.get_y() << endl;
                                                                                                                         다.(0~500)" << endl;
call\n";
                                           if (x1 > x2) {
                                                                                   for (int i = 0; i < up_left.get_y() -
                                             low_right.set_x(x1);
                                                                                low_right.get_y(); i++) {
                                                                                                                             return;
                                             up_left.set_x(x2);
                                                                                     for (int i = 0; i < up_left.get_x();
  ~rectangle() {
                                                                                i++) cout << " ";
     cout << "rectangle Destructor</pre>
                                                                                                                           y = y_pos;
call\n";
                                                                                     for (int i = 0; i < low_right.get_x() }
                                           else {
                                                                                - up_left.get_x(); i++) cout << "■";
                                             up_left.set_x(x1);
  void init_rec(const int& x1, const
                                             low_right.set_x(x2);
                                                                                     cout << endl;
```

```
pos2.get_y(): pos1.get_y();
//두 점으로 사각형 만들기
                                          py2 = (pos1.get_y() > pos2.get_y())? << endl;
#include <iostream>
                                                                                                                       void point::init_pos(int x_pos, int
                                                                                  system("pause");
using namespace std;
                                       pos2.get_v() : pos1.get_v();
                                                                                                                       y_pos) {
class point {
                                          up_left = new point(px1, py1);
                                                                                                                        if (0 > x_pos | | 500 < x_pos) {
private:
                                          low_right = new point(px2, py2);
                                                                                return 0;
                                                                                                                          cout << "X좌표 범위를 벗어났습니
 int x, y; //x좌표, y좌표
                                          cout << "rectangle constructor
                                                                                                                       다.(0~500)" << endl;
                                                                               void rectangle::init_rec(int& x1, int&
public:
                                       call\n";
                                                                                                                          return;
 point(int _x = 0, int _y = 0) {
                                                                               v1, int& x2, int& v2) {
                                                                                                                         if (0 > y_pos || 500 < y_pos) {
  x = x;
                                         ~rectangle() {
                                                                                 int px1, py1, px2, py2;
                                                                                px1 = (x1 > x2) ? x2 : x1;
                                                                                                                          cout << "X좌표 범위를 벗어났습니
  y = _y;
                                          delete up_left;
  cout << "point Constructor call\n";</pre>
                                          delete low_right;
                                                                                 px2 = (x1 < x2) ? x2 : x1;
                                                                                                                       다.(0~500)" << endl;
                                          cout << "rectangle Destructor
                                                                                py1 = (y1 < y2) ? y2 : y1;
                                                                                                                          return;
 ~point() {
                                       call\n";
                                                                                 pv2 = (v1 > v2) ? v2 : v1;
  cout << "point Destructor call\n";</pre>
                                                                                up_left = new point(px1, py1);
                                                                                                                        x = x_pos;
                                        void init_rec(int& x1, int& y1, int& x2, low_right = new point(px2, py2);
                                                                                                                        y = y_pos;
 void init_pos(int x_pos, int y_pos);
                                                                                                                       int point::get_x() { return x; }
 int get_x();
                                         void init_rectangle(point& ul, point& void rectangle:init_rectangle(point&
                                                                               pos1, point& pos2) {
                                                                                                                       int point::get_y() { return y; }
 int get_y();
                                        lr);
 void set_x(int x_pos);
                                         void show_rectangle();
                                                                                int px1, py1, px2, py2;
                                                                                                                       void point::set_x(int x_pos) {
 void set_y(int y_pos);
                                                                                 px1 = (pos1.get_x() > pos2.get_x())?
                                                                                                                        if (0 > x_pos | 500 < x_pos)
                                                                                                                          cout << "X좌표 범위를 벗어났습니
                                       int main() {
                                                                               pos2.get_x() : pos1.get_x();
                                                                                px2 = (pos1.get_x() < pos2.get_x())?
                                                                                                                       다.(0~500)" << endl;
class rectangle {
                                         while (1) {
                                          //system("cls");
                                                                               pos2.get_x() : pos1.get_x();
private:
                                                                                                                          return;
 point* up_left;// 왼쪽 상단 좌표 1
                                          cout << "1번 좌표를 입력하세요 (x y,
                                                                                py1 = (pos1.get_y() < pos2.get_y())?
 point* low_right;// 오른쪽 하단 좌표 2 공백 구분, 범위:0~500, 종료:-1):";
                                                                               pos2.get_y() : pos1.get_y();
                                                                                                                        x = x_pos;
public:
                                                                                py2 = (pos1.get_y() > pos2.get_y())?
                                          int x1, y1;
 rectangle(int& x1, int& y1, int& x2,
                                          cin >> x1 >> y1;
                                                                               pos2.get_y() : pos1.get_y();
                                                                                                                       void point::set_y(int y_pos) {
int& v2) {
                                          if (x1 == -1 || v1 == -1) break;
                                                                                up_left = new point(px1, py1);
                                                                                                                        if (0 > y_pos || 500 < y_pos) {
                                                                                                                          cout << "Y좌표 범위를 벗어났습니
  int px1, py1, px2, py2;
                                          point pos1(x1, y1);
                                                                                 low_right = new point(px2, py2);
                                          cout << "2번 좌표를 입력하세요 (x y,
                                                                                                                       다.(0~500)" << endl;
  px1 = (x1 > x2) ? x2 : x1;
  px2 = (x1 < x2) ? x2 : x1;
                                       공백 구분, 범위:0~500, 종료:-1): ";
                                                                               void rectangle::show_rectangle() {
                                                                                                                          return;
  py1 = (y1 < y2) ? y2 : y1;
                                          int x2, y2;
                                                                                 cout << endl << endl << endl
  py2 = (y1 > y2) ? y2 : y1;
                                          cin >> x2 >> v2;
                                                                               << endl:
                                                                                                                         y = y_pos;
  up_left = new point(px1, py1);
                                          if (x2 == -1 || y2 == -1) break;
                                                                                for (int i = 0; i \le 500; i++) {
  low_right = new point(px2, py2);
                                                                                  for (int j = 0; j <= 500; j++) {
                                          point pos2(x2, y2);
                                                                                   if (up_left->get_x() <= j \&\& j <=
                                          rectangle rec1(pos1, pos2);
 rectangle(point& pos1, point& pos2) {
                                          rectangle rec2(x1, y1, x2, y2);
                                                                               low_right->get_x() && low_right-
                                          rec1.show_rectangle();
  int px1, py1, px2, py2;
                                                                               > get_y() <= i \&\& i <= up_left-> get_y())
  px1 = (pos1.get_x() > pos2.get_x())?
                                                                               -cout << "■";
pos2.get_x() : pos1.get_x();
                                                                                   else cout << " ";
  px2 = (pos1.get_x() < pos2.get_x())? << endl;
                                          rec2.show_rectangle();
pos2.get_x() : pos1.get_x();
                                                                                  cout << endl;
  py1 = (pos1.get_y() < pos2.get_y())?
```

```
[명단 관리 프로그램] [Directory Program]
                                                 typedef student* NodeElement;
                                                 class Node {
1. 등록
                        1. REGIST
                                                 private:
2. 삭제
                        2. REMOVE
                                                  NodeElement value;
3. 리스트 출력
                        3. DISPLAY
                                                  Node* link:
4. 전체 삭제
                        4. REMOVEALL
                                                 public:
0. 프로그램 종료
                        0. TERMINATE
                                                  Node();
메뉴 선택:
                        Select Menu:
                                                  ~Node();
                                                  void init();
                                                  void Delete();
                                                  inline NodeElement getValue();
                                                  inline Node* getlink();
                                                  inline void setValue(NodeElement data);
                                                  inline void setlink(Node* _link);
                                                 };
class student {
private:
                                                 class SingleList {
 char* name;
                                                 private:
 char* phone;
                                                   Node* head;
                                                 public:
public:
 student();
                                                  SingleList();
 ~student();
                                                  ~SingleList();
 inline void init();
                                                  inline void init() { head = NULL; }
                                                   inline void Delete() { delete head; head = NULL; }
 inline void Delete();
                                                   bool InsertNode(NodeElement data);
 inline char* getName();
 inline char* getPhone();
                                                   bool DeleteNode(NodeElement data);
 inline bool setName(char*_name);
                                                   bool DeleteAll();
 inline bool setPhone(char* _phone);
                                                   bool PrintList();
};
```

Day07 - 2. 예제 코드



Header.H

```
#pragma once
#ifndef _REGISTER_H
#define _REGISTER_H
#include<iostream>
using namespace std;
void cinclear() {
if (cin.fail()) {
  cin.clear();
  while (cin.get() != '\n');
 while (cin.get() != '\n');
void cinput(int& value) {
 while (!(cin >> value)) {
  cin.clear();
  while (cin.get() != '\n');
  cout << "\t 다시 입력 하세요.: ";
 while (cin.get() != '\n');
enum menuType {
 TERMINATE, REGISTER, REMOVE, DISPLAY,
REMOVEALL
class student {
private:
 char* name;
 char* phone;
public:
 student():name(NULL), phone(NULL) {}
 ~student() {
  if (name != NULL) delete[] name;
  if (phone != NULL) delete[] phone;
 inline void init() { name = phone = NULL; }
 inline void Delete() {
  if (name != NULL) delete[] name;
  if (phone != NULL) delete[] phone;
 inline char* getName() { return name; }
 inline char* getPhone() { return phone; }
 inline bool setName(char* _name) {
  if (_name == NULL) return false;
  name = name;
  return true;
 inline bool setPhone(char* _phone) {
  if (phone == NULL) return false;
  phone = _phone;
  return true;
```

```
typedef student* NodeElement;
class Node {
private:
 NodeElement value;
 Node* link;
public:
 Node(){
  value = new student;
  link = NULL:
 Node(NodeElement _value) :value(_value).
link(NULL) {}
 ~Node() {
  if (value != NULL) { delete value; }
  value = NULL;
  link = NULL;
 void init() {
  value = new student:
  link = NULL;
 void Delete() {
  if (value != NULL) { delete value; }
  value = NULL:
  link = NULL;
 inline NodeElement getValue() { return value; }
 inline Node* getlink() { return link; }
 inline void setValue(NodeElement_data) { value =
 inline void setlink(Node* _link) { link = _link; }
class SingleList {
private:
 Node* head;
public:
 SingleList():head(NULL) {}
 SingleList(Node* node) :head(node) {}
 ~SingleList() {
  DeleteAll();
 inline void init() { head = NULL; }
 inline void Delete() { delete head; head = NULL; }
 bool InsertNode(NodeElement data);
 bool DeleteNode(NodeElement data);
```

```
bool DeleteAll();
                                                 if (answer == 'Y' || answer == 'y') {
                                                  for (Node* pNode = head, *DeleteNode = NULL;
 bool PrintList();
                                                pNode != NULL;) {
                                                   DeleteNode = pNode;
                                                    pNode = pNode->getlink();
void PrintMenu(int& menu);
                                                   DeleteNode->Delete();
                                                   delete DeleteNode;
bool SingleList::InsertNode(NodeElement data) {
                                                   cout << "\t\t 삭제 성공! \n\n";
 Node* NewNode = new Node;
 if (NewNode == NULL) return false;
                                                  head = NULL;
 NewNode->init();
                                                  cout << "\t\t [ 목록을 모두 지웠습니다.]\n\n";
 NewNode->setValue(data);
                                                  return true;
 NewNode->setlink(head);
 head = NewNode;
 return true;
                                                  cout << "\t\t [ 목록을 지우지 않고 메뉴로 돌아갑니
                                                다. ]\n\n";
                                                  return false;
bool SingleList::DeleteNode(NodeElement data) {
 if (head == NULL) {
  cout << "\t\t목록을 추가해 주세요. \n\n";
                                                bool SingleList::PrintList() {
  return false;
                                                 if (head == NULL) {
 Node* DeleteNode = head, * preDelete = NULL;
                                                  cout << "\t\t목록이 비어있습니다.\n\n";
 while (DeleteNode != NULL) {
                                                  return false;
  if ((strcmp(DeleteNode->getValue()->getName(),
data->getName()) == 0) && (strcmp(DeleteNode-
                                                 for (Node* pNode = head; pNode != NULL; pNode
>getValue()->getPhone(), data->getPhone()) == 0)) = pNode->getlink()) {
                                                  cout << endl;
   if (preDelete == NULL) head = DeleteNode-
                                                  cout << "\t\t 이름: " << pNode->getValue()-
>getlink();
                                                >getName() << endl;
   else preDelete->setlink(DeleteNode->getlink()); cout << "\t\t 전화번호: " << pNode->getValue()-
   delete DeleteNode;
                                                >getPhone() << endl;
   cout << "\t\t 삭제 성공! \n\n";
   return true;
                                                 return true;
  preDelete = DeleteNode;
  DeleteNode = DeleteNode->getlink();
                                                void PrintMenu(int& menu) {
                                                 cout << "\n\n\t\t[ 명단 관리 프로그램 ]\n\n";
 cout << "\t\t목록에 삭제 할 자료가 없습니다. \n\n"; cout << "\t\t1. 등록\n";
 return false;
                                                 cout << "\t\t2. 삭제\n";
                                                 cout << "\t\t3. 리스트 출력\n";
                                                 cout << "\t\t4. 전체 삭제\n";
bool SingleList::DeleteAll() {
                                                 cout << "\t\t0. 프로그램 종료\n\n";
 cout << endl << endl;
                                                 cout << "\t\t메뉴 선택:";
 cout << "\t\t [ 목록을 모두 지우시겠습니까??
                                                 cinput(menu);
(Y/N) ]\n\n";
 cout << "\t\t 입력하세요.:";
                                                #endif // _REGISTER_H
char answer = 'n';
 cin >> answer;
 cinclear();
```

Main.cpp

```
#include <iostream>
#include "register.h"
using namespace std;
int main() {
 int menu = 1;
 char buf[255];
 SingleList list1;
 while (1) {
  system("cls");
  PrintMenu(menu);
  switch (menu) {
  case REGISTER:
   cout << "\t\t 이름: ";
   cin >> buf;
   cinclear();
   NodeElement Register = new student;
   Register->setName(new char[strlen(buf) + 1]);
   strcpy_s(Register->getName(), strlen(buf) + 1, buf);
   cout << "\t\t 전화번호:";
   cin >> buf;
   cinclear();
   Register->setPhone(new char[strlen(buf) + 1]);
   strcpy_s(Register->getPhone(), strlen(buf) + 1, buf);
   list1.InsertNode(Register);
   list1.PrintList();
   break;
  case REMOVE:
   cout << "\t\t 이름: ";
   cin >> buf;
   cinclear();
   NodeElement Deletion = new student:
```

```
Deletion->setName(new char[strlen(buf) + 1]);
  strcpy_s(Deletion->getName(), strlen(buf) + 1, buf);
  cout << "\t\t 전화번호 : ";
  cin >> buf;
  cinclear();
  Deletion->setPhone(new char[strlen(buf) + 1]);
  strcpy_s(Deletion->getPhone(), strlen(buf) + 1, buf);
  list1.DeleteNode(Deletion);
  list1.PrintList();
  delete Deletion;
  break;
 case DISPLAY: list1.PrintList(); break;
 case REMOVEALL:
  list1.DeleteAll();
  list1.PrintList();
  break;
 case TERMINATE:
  cout << "\n\n\t\t[ 프로그램 종료 합니다. ]\n\n";
  exit(0);
 cout << endl << endl;</pre>
 system("pause");
return 0:
```

Day08 - Contents

- 1. this
- 2. 상수화(const)
- 3. 정적 멤버 (static)
- 4. 예제 코드

Day08 - 1. this

this포인터

- 멤버 함수에서 호출한 객체의 주소를 가리키는 숨겨진 (상수) 포인터
- 모든 멤버 함수는 객체의 주소(this 포인터)를 숨겨진 매개변수로 가진다.
- 컴파일 과정에서 멤버 함수에 this 포인터 인수가 추가 된다.
- 함수가 실행되는 동안에만 스택에 this 포인터가 매개변수로 쌓인다.

객체선언 시 멤버 변수만 각각 할당하고, 멤버 함수는 공유한다. 멤버 함수 호출 시 어떤 객체가 멤버 함수를 호출했는지 알아야 한다. 이 때, 함수의 인수(매개변수)로 객체의 주소를 전달한다.

		색제 ╽	색제 2		
객체 1	객체 2		멤버 변수 1.1	멤버 변수 2.1	
멤버 변수 1.1	멤버 변수 2.1		급이 한구 1.1	급미 친구 2.1	
			멤버 변수 1.2	멤버 변수 2.2	
멤버 변수 1.2	멤버 변수 2.2				
멤버 함수 1.1	멤버 함수 2.1		멤버 함수 1 (객체* ptr=this)		
0-1 0 1 1.1	0-1 0 1 2.1				
멤버 함수 1.2	멤버 함수 2.2		멤버 함수 2 (객체* ptr=this)		
			고이		

OIT

```
#include <iostream>
using namespace std;
class B{
private:
   int i, j;
public:
   void Set(int _i, int _j){
                                                                       void Set(B* const this, int _i, int _j) {
      i = _i;
                                                                          this->i = i;
      j = _j;
                                                                          this-> j = _j;
   void Out(){
      cout << i << ", " << j << endl;
int main(){
   B ob1, ob2, ob3;
   ob1.Set(1, 10);
ob2.Set(2, 20);
ob3.Set(3, 30);
                                                                       Set(&ob1, 1, 10);
Set(&ob2, 2, 20);
Set(&ob3, 3, 30);
   ob1.Out();
   ob2.Out();
   ob3.Out();
   return 0;
```

Day08 - 4. 예제 코드

this포인터 활용

- 멤버 변수와 이름이 같은 매개 변수를 가진 경우
- 멤버 함수에서 객체를 반환하는 경우

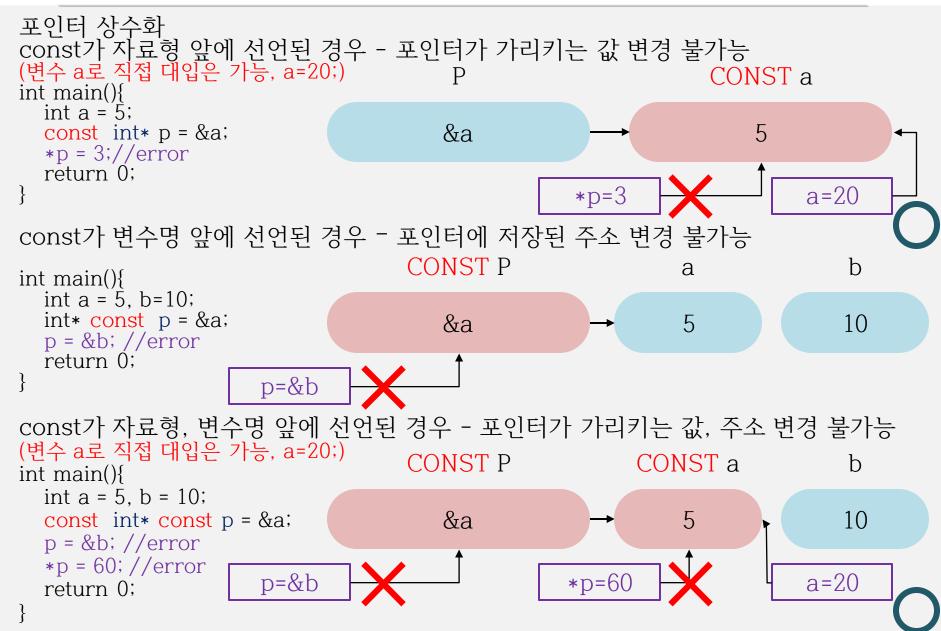
```
#include <iostream>
using namespace std;
class B {
private:
  int i, j;
public:
  void Set(int i, int j) {
     this->i = i; //m_i
     this->i = i; //m_i
  void Out() {
     cout << i << ", " << j << endl;
int main() {
  B ob1, ob2, ob3;
  ob1.Set(1, 10);
  ob2.Set(2, 20);
  ob3.Set(3, 30);
  ob1.Out();
  ob2.Out();
  ob3.Out();
  return 0;
```

```
#include <iostream>
using namespace std;
class Calc {
private:
  int m_Value = 0;
public:
  Calc& Add(int value) { m_Value += value; return *this; }
  Calc& Sub(int value) { m_Value -= value; return *this; }
  Calc& Mul(int value) { m_Value *= value; return *this; }
  int GetValue() { return m Value; }
int main(){
  Calc obj1;
  obj1.Add(10).Sub(5).Mul(4);
        [obj1.Add(10)];
        [obj1.Add(10)] .Sub(5)];
        [obi1.Add(10)] .Sub(5)] .Mul(4)];
  cout << obj1.GetValue() << endl; // 20
  return 0;
```

Day08 - 4. 예제 코드

```
#include <iostream>
#include <iomanip>// setw
using namespace std;
class Telephone {
private:
int no;
 char name[20];
 char phone[20];
public:
 Telephone(): no(0), name(""), phone("") {}
 Telephone(int no. const char* const name, const char* const phone) {
  this->no = no;
  strcpy_s(this->name, 20, name);
  strcpy_s(this->phone, 20, phone);
  cout << "Constructor complete" << endl;</pre>
 int GetNo() const { return this->no; }
 const char* GetName() const { return this->name; }
 const char* GetPhone() const { return this->phone; }
int main() {
 char name[20], ph[20];
 cout << "** 전화번호부 **\n \n";
 cout << "이 름:";
 cin >> name;
 cout << "전화번호:";
 cin >> ph;
 Telephone p(1, name, ph); //객체 생성 시 인수 전달
 cout << "\n \n \t** 결과출력 **\n \n";
 cout << std::left << setw(5) << "No." << setw(20) << "이름" << setw(20) << "전화번호" << endl;
 cout << setw(5) << p.GetNo() << setw(20) << p.GetName() << setw(20) << p.GetPhone() << endl;
 return 0;
```

```
상수화
- 변하지 않는 데이터
- 자료를 수정 할 수 없도록 대입 연산을 금지하는 것
변수의 상수화
변수 앞에 const 키워드를 붙이면 상수가 된다.
(값 변경 시 에러 발생)
                                               CONST PI
int main(){
  const double PI = 3.141592;
                                               3.141592
 PI = -5; //error
 return 0;
                                  PI=-5
상수는 반드시 선언과 초기화를 동시에 한다.
변수는 지역변수 선언 시 쓰레기 값으로 초기화되기 때문이다.
int main(){
                                               CONST PI
  const double PI;//초기화 필요
 PI = -5; //error
                                                 ???
 return 0;
                                  PI=-5
```



멤버변수 상수화 메모리 할당 후, 쓰레기 값으로 초기화 되기 때문에 오류발생

```
#include <iostream>
                                                        CONST number
#include <cstring>
class Student {
                                                              ???
private:
  const int number;
  char name[20];
                                     number=Num
  int age;
public:
  Student(int Num, const char* pN, int a) {
    number = Num; // const 초기화 error
    strcpy(name, pN);
    age = a;
int main() {
  Student s1(20221234, "michelin", 21);
  Student s2(20231234, "christopher", 20);
  return 0;
```

```
멤버변수 상수화
":"콜론 초기화 <초기화 리스트(Initialize List)> 생성자의 body 부분보다 먼저 수행된다.
                                                              CONST number
#include <iostream>
#include <cstring>
                                                                    Num
class Student {
private:
  const int number;
  char name[20];
  int age;
                                                //선언과 동시 초기화
public:
  Student(int Num, const char* pN, int a): number(Num) {
     //number = Num; const 초기화 error
     strcpy_s(name, strlen(pN) + 1, pN);
     age = a;
int main() {
  Student s1(20221234, "michelin", 21);
Student s2(20231234, "christopher", 20);
  return 0;
```

```
멤버함수 상수화
- 멤버변수의 수정 불가능하다. (읽기전용)
- 상수화된 함수는 일반함수 호출이 불가능
- 상수화된 함수는 멤버변수의 주소 리턴 불가능
```

```
#include <iostream>
#include <iomanip>// setw
#include <cstring>
using namespace std;
class Student {
private:
 const int number;
 char name[20];
 int age;
public:
 Student(int Num, const char* pN, int a):
number(Num) {
  //number = Num;
  strcpy_s(name, strlen(pN) + 1, pN);
  age = a;
 void init() {
  cin >> name;
  cin >> age;
  Show();
 // 멤버변수 주소 리턴 불가능
int getage() const { return age; }
// 멤버변수 값 리턴 가능
 void print() const { cout << age; }</pre>
```

```
void Show() const // {}
  //age = 22; //멤버변수 수정 불가능
  //init(); // 일반함수 호출 불가능
  //getname();
  getage();
  print(); // 상수화 멤버 함수 호출 가능
cout << setw(10) << number << endl;
  cout << setw(15) << name << endl;
  cout << setw(5) << age << endl;
int main() {
 Student s1(20221234, "michelin", 21);
 Student s2(20231234, "christopher", 20);
 s1.Show();
 s2.Show();
 return 0;
```

Day08 - 3. 정적 멤버 (static)

정적 멤버 (static) - static을 붙여 선언한 멤버

- 정적 멤버 외부 정의 시 staitc 키워드는 선언부에만 쓴다.
- 정적 멤버 함수는 정적 멤버 변수, 정적 멤버 함수만 접근할 수 있다.
- 정적 멤버는 동일한 클래스의 모든 객체가 공유한다.
- main 함수가 호출되기 전에 메모리 공간에 올라가서 초기화 (전역변수 특징) < 객체 생성과는 관계 없이 먼저 할당, 사용 가능 >

객체 2 객체 1 멤버 변수 1.1 멤버 변수 2.1 멤버 변수 1.2 멤버 변수 2.2 멤버 함수 1.1 멤버 함수 2.1 멤버 함수 2.2 멤버 함수 1.2



Day08 - 3. 정적 멤버 (static)

```
정적 멤버 ( static )
  static을 붙여 선언한 멤버
 정적 멤버 외부 정의 시 staitc 키워드는 선언부에만 쓴다.
정적 멤버 함수는 정적 멤버 변수, 정적 멤버 함수만 접근할 수 있다.
정적 멤버는 동일한 클래스의 모든 객체가 공유한다.
- main 함수가 호출되기 전에 메모리 공간에 올라가서 초기화 (전역변수 특징)
  < 객체 생성과는 관계 없이 먼저 할당, 사용 가능 >
형식
class Point_1 {
private:
 int x, y;
 static int creat cnt;
public:
 static int call_cnt;
 Point_1(int _x, int _y) : _x(_x), _y(_y) {
  creat_cnt++;
 static void OutPointCount(); //선언부
int Point_1::creat_cnt = 0, Point_1::call_cnt = 0;
//초기화 필수 - 정적 멤버 지명은 변수명 앞에 < 클래스명 :: >
void Point_1::OutPointCount() {//정의부
 call cnt++;
 cout << "생성된 Point_1 는 " << creat_cnt << "개 입니다.\n";
 cout << "Point_1 호출 횟수는 " << call_cnt << "개 입니다.\n";
```

Day08 - 3. 정적 멤버 (static)

```
정적 멤버 (static)
- static을 붙여 선언한 멤버
- 정적 멤버 외부 정의 시 staitc 키워드는 선언부에만 쓴다.
- 정적 멤버 함수는 정적 멤버 변수, 정적 멤버 함수만 접근할 수 있다.
- 정적 멤버는 동일한 클래스의 모든 객체가 공유한다.
- main 함수가 호출되기 전에 메모리 공간에 올라가서 초기화 (전역변수 특징) < 객체 생성과는 관계 없이 먼저 할당, 사용 가능 >
int main(){
  cout<<Point_1::call_cnt<<"회 호출"<<endl; //객체 생성 전에 사용 가능한 멤버
 // 정적 멤버 지명은 변수명 앞에 <클래스명 :: >
                                             class Point_1 {
  Point_1 p1(10, 10);
                                             private:
  p1.OutPointCount();
                                              int x, y;
                                              static int creat cnt;
  Point_1 p2(20, 20);
                                             public:
  p2.OutPointCount();
                                              static int call_cnt;
                                              Point_1(int _x, int _y) : _x(_x), _y(_y) {
                                                creat_cnt++;
  Point_1 p3(30, 30);
  p3.OutPointCount();
                                              static void OutPointCount(); //선언부
  p1.OutPointCount();
                                             int Point_1::creat_cnt = 0, Point_1::call_cnt = 0;
  cout << endl;
                                             void Point_1::OutPointCount() {//정의부
  return 0;
                                              call cnt++;
                                              cout << "생성된 Point_1 는 " << creat_cnt << "개 입니다.\n"; cout << "Point_1 호출 횟수는 " << call_cnt << "개 입니다.\n";
```

Day08 - 4. 예제 코드

★과제★ student 클래스 설계

private멤버변수 - 학번 - 이름 - 나이

- 학생수 (static)

public 멤버함수

- 생성자 (20230000부터 차례로 1씩 증가 되도록 구현) print_info: 학생 정보 출력 (const) print_std_cnt: 학생 수 출력 (static)

단, 멤버 변수의 변경이 필요 없는 함수는 멤버함수를 상수화 시켜 읽기 전용 함수로 구현

Day08 - 4. 예제 코드

```
name) { strcpy_s(this->name, 20,
                                                                               Student st2("christopher", 21);
#include <iostream>
#include <iomanip>
                                      name); }
                                                                            st1.print_std_cnt();
                                        void SetAge(int age) { this->age =
                                                                               Student st3("charlie puth", 22);
using namespace std;
                                                                            st1.print_std_cnt();
class Student {
                                      age; }
                                      };
private:
                                      int Student::studentNum = 0; //정적멤
                                                                               st1.print_info(); st2.print_info();
  int no;
  char name[20];
                                      버변수 초기화
                                                                            st3.print_info();
                                      int Student::GetStudentNum(){
  int age;
                                        return studentNum;
                                                                               st1.SetName("bruno mars");
  static int studentNum; //학생수(정적
멤버변수)
                                                                            st2.SetAge(23);
                                                                               cout << "\n\n정보를 수정했습니다.\n";
public:
                                      void Student::print_info() const {
  static int GetStudentNum(); //정적멤
                                      cout << std::left << std::setw(10) << "
                                                                               st1.print_info();
                                      학번" << std::setw(20) << "이름"
버함수
                                                                               st2.print_info();
  Student(): no(0), name(""), age(0) { << std::setw(5) << "나이" << endl;
                                                                               st3.print_info();
                                        cout << std::setw(10) << this->GetNo()
//initializer lists
                                                                              return 0:
  Student(const char* const name, int<< std::setw(20) << this->GetName()
                                      << std::setw(5) << this->GetAge()
age) {
     this->no = 20230000 +
                                      << endl << endl;
studentNum:
     ++studentNum:
                                      void Student::print_std_cnt() const {
                                        cout << "전체 학생 수는 "
     strcpy_s(this->name, 20, name);
     this->age = age;
                                      << Student::GetStudentNum() << "명 입
     cout << "생성자 초기화\n";
                                      니다.\n";
  int GetNo() const { return no; }
                                      int main() {
  const char* GetName() const
                                        cout << "전체 학생 수는 "
                                      << Student::GetStudentNum() << "명 입
{ return name; }
                                      니다.\n"; //정적멤버는 클래스명으로접근
  int GetAge() const { return age; }
  void print_info() const;
  void print_std_cnt() const;
                                        Student st1("michelin", 20);
  void SetName(const char* const
                                      st1.print_std_cnt();
```

Day09 - Contents

- 1. Copy Constructor
- 2. 객체와 배열, 포인터
- 3. 예제 코드

Day09 - 1. Copy Constructor

Copy Constructor(복사 생성자)

- 주 사용 목적 : 객체간의 멤버 대 멤버 복사
- 자신과 같은 타입의 객체를 인수로 받을 수 있는 생성자 (컴파일 자동 생성)
- 반드시 Reference나 Pointer를 객체의 매개변수로 사용 (주로 Reference)

Document(const Document&) {......} // 멤버 : 멤버 복사 생성자

복사 생성자 호출 시점

};

- (1) 객체를 선언하면서 객체로 초기화 하는 경우
- (2) 함수의 매개변수로 객체를 값으로 전달한 경우 (call by value)
- (3) 객체를 리턴 시 값에 의해 리턴 하는 경우 (call by value - 반환 시 값을 복사 / C++17 개정)

Day09 - 1. Copy Constructor

```
#include <iostream>
using namespace std;
                                            복사 생성자 호출 시점
class A {
private:
                                            (1) 객체를 선언하면서 객체로 초기화 하는 경우
  int a;
                                             (2) 함수의 매개변수로 객체를 값으로 전달한 경우
  double d:
  char c;
                                                 (call by value)
public:
  A(int _a=0, double _d=0, char _c=0);//생성자
 A(A\& r);
                       //복사생성자
  void Show();
                        //출력함수
void display_value(A a) { a.Show(); } //call by value
void display_ref(A& a) { a.Show(); } //call by reference
void display_ptr(A* a) { a->Show(); } //call by reference
A creat_obj() { A obj(2, 6.87, 'B'); return obj; } //return by value ( C++17 개정으로 때에 따라 다름 )
int main() {
  A ob1(1, 3.54, 'A');
  A ob2(ob1); //(1) 객체를 선언과 동시에 객체로 초기화 하는 경우 복사생성자가 호출됨
  A ob3 = ob1; //(1) 객체를 선언과 동시에 객체로 초기화
  A ob4:
  ob4 = creat obi();
  ob1.Show(); ob2.Show(); ob3.Show();
  display_value(ob1); //(2) 함수의 매개변수로 객체를 값으로 전달한 경우 call by value
  display_ref(ob2); display_ptr(&ob3); display_ref(ob4); //(2) 함수의 매개변수로 객체를 레퍼런스(주소)로 전달한 경우 call by reference
  return 0;
A::A(int _a, double _d, char _c) : a(_a), d(_d), c(_c) { cout << "생성자 호출\n"; }
A::A(A& r) {//반드시 레퍼런스나 포인터로 받아야 한다. 멤버 대 멤버 복사(객체간의 복사)
  this->a = r.a:
  this->d = r.d;
  this->c = r.c;
  cout << "복사 생성자 호출\n";
void A::Show() {
  cout << a << ' ' << d << ' ' << c << endl:
```

Day09 - 1. Copy Constructor



소멸자에서 동적 메모리 해제 시 같은 메모리를 두 번 해제하며 오류 발생.

깊은 복사 (: 포인터가 가리키는 메모리 만큼 할당하여 복사하는 방식)
 포인터는 주소값을 복사하는 방식이 아닌
 포인터가 가리키는 메모리만큼 동적 할당 받아 복사해야 한다.

Day09 - 3. 예제 코드

```
//얕은 복사 포인터 오류 코드
#include <iostream>
using namespace std;
class MyString {
private:
  char* pStr;
public:
  MyString(char* p) {
    pStr = new char[strlen(p) + 1];
    strcpy_s(pStr,strlen(p)+1, p);
    cout << "생성자 호출\n";
  //MyString(MyString& r);
  ~MyString() {
    delete[]pStr;
    cout << "동적메모리 해제\n";
  void Show() const {
    cout << pStr << endl;
int main() {
  MyString s1 = (char*)"Hello World";
  MyString s2 = (char*)"C++ language";
  MyString s3 = s1; //default 복사 생성자
  s1.Show();
  s2.Show();
  s3.Show();
  return 0;
               얕은 복사
   s1* pStr
                                   Α
   s3* pStr
```

```
//깊은 복사 코드
#include <iostream>
using namespace std;
class MyString {
private:
  char* pStr;
public:
  MyString(char* p);
  ~MyString();
  MyString(const MyString& MyString);
  void Show() const;
MyString::MyString(char* p) {//생성자
                                                      깊은 복사
  pStr = new char[strlen(p) + 1];
  strcpy_s(pStr, strlen(p) + 1, p);
                                                                        string
  cout << "생성자 호출\n";
                                         s1* pStr
                                                                         .pStr
MyString::~MyString() {//소멸자
                                                                        string
  delete[]pStr;
                                         s3* pStr
  cout << "동적메모리 해제\n";
                                                                         .pStr
MyString::MyString(const MyString& string) { //깊은 복사 생성자
  pStr = new char[strlen(string.pStr) + 1];
  strcpy_s(pStr, strlen(string.pStr) + 1, string.pStr);
  cout << "pStr 포인터가 가리키는 동적메모리 모두 복사 -> 깊은 복사 수행하는 복사생성자\n";
void MvString::Show() const {
  cout << pStr << endl;
int main() {
  MyString s1 = (char*)"Hello World"; //MyString s1((char*)"Hello World");
  MyString s2 = (char*)"C++ language"; //MyString s1((char*)"C++ language");
  MyString s3 = s1; //깊은 복사 생성자
  s1.Show();
  s2.Show();
  s3.Show();
  return 0;
```

```
/*
  객체배열
  - 객체가 여러 개 필요할 경우 배열로 선언
#include <iostream>
using namespace std;
class A {
private:
  int a;
public:
  A() { cout << "생성자 호출\n"; }
  void Set(int _a) { a = _a; cout << "값 설정 "; }
  void Show(int i) { cout << "a[" << i << "] = " << a << endl; }</pre>
int main() {
  A ob1, ob2, ob3, ob4, ob5;
  A ob[5]; //객체배열
  for (int i = 0; i < 5; i++) {
     ob[i].Set(i + 1);
     ob[i].Show(i);
  return 0;
```

```
/*
  객체배열과 생성자
   - 생성자의 인수가 한개인 경우
#include <iostream>
using namespace std;
class A {
private:
  int a;
public:
  A() { cout << "생성자 호출\n"; }
  A(int _a) { a = _a; cout << _a<< "생성자 호출\n"; }
  ~A(){ cout << "소멸자 호출\n"; }
  void Set(int _a) { a = _a; cout << "값 설정 "; }
  void Show(int i) { cout << "a[" << i << "] = " << a << endl; }</pre>
};
int main() {
  //A \text{ ob1}(1), \text{ ob2}(2), \text{ ob3}(3), \text{ ob4}(4), \text{ ob5}(5);
  //A obi[5] = { 1, 2, 3, 4, 5 }; //객체배열 간략화된 초기화 방법
  A obi[5] = { A(6), A(7), A(8), A(9), A(10) }; //객체배열 초기화 방법
  for (int i = 0; i < 5; i++) obj[i]. Show(i);
  for (int i = 0; i < 5; i++) obj[i]=A(i*10);
  for (int i = 0; i < 5; i++) obj[i]. Show(i);
  return 0;
```

```
/*
 객체배열과 생성자
 - 생성자의 인수가 두개 이상인 경우
*/
#include <iostream>
using namespace std;
class A {
private:
 int a. b;
public:
 A(int _a, int _b) {
  a = _a;
  b = b;
  cout << a<< ""<< b << "생성자 호출\n";
 ~A() { cout << "소멸자 호출\n"; }
 void Show(int i) { cout << "a[" << i << "] : " << "a = " << a << ", b = " << b << endl; }
int main() {
 A ob1(1, 10), ob2(2, 20), ob3(3, 30), ob4(4, 40), ob5(5, 50);
 A ob[5] = { {1, 10}, {2, 20}, {3, 30}, {4, 40}, {5, 50} }; //객체배열 간략화된 초기화 방법
 //A ob[5] = { A(1, 10), A(2, 20), A(3, 30), A(4, 40), A(5, 50) }; //객체배열 초기화 방법
 for (int i = 0; i < 5; i++) ob[i]. Show(i);
 for (int i = 0; i < 5; i++) ob[i] = A(i+10,i * 10);
 for (int i = 0; i < 5; i++) ob[i]. Show(i);
 return 0;
```

```
/*
 객체 포인터
 - 객체의 주소를 저장하는 포인터
#include <iostream>
using namespace std;
class A {
private:
int a, b;
public:
A() { cout << "생성자 호출\n"; }
 void Set(int _a, int _b) {
  a = a;
  b = b;
 void Show() { cout << "a = " << a << ", b = " << b << endl; }</pre>
int main() {
A* p; //객체 포인터 생성 - 객체의 주소를 저장하는 포인터 - 8 byte
 p = new A; //동적 할당된 객체 메모리를 객체 포인터가 가리킴
 p->Set(1, 10); //객체 포인터로 멤버 접근 시 "->"로 접근
 p->Show();
 delete p; //동적객체 해제
 return 0;
```

```
객체 포인터 배열
  - 객체포인터가 여러개 필요한 경우 사용
#include <iostream>
using namespace std;
class A {
private:
  int a, b;
public:
  A(int _a, int _b) {
    a = _a;
    b = _b;
    cout << "생성자 호출\n";
  void Show(int i) { cout << "a[" << i << "] : " << "a = " << a << ", b = " << b << endl; }</pre>
int main() {
  A* p[5]; //객체 포인터 배열(포인터 5개 = 8byte * 5개)
  for (int i = 0; i < 5; i++) p[i] = new A(i, i * 10); //객체 생성 시 인수 전달
  for (int i = 0; i < 5; i++) p[i]->Show(i); //객체포인터->멤버함수
  for (int i = 0; i < 5; i++) {
    delete p[i]; //delete 포인터;
    cout << "동적메모리 해제(객체)\n";
  return 0;
```

Day09 - 3. 예제 코드

```
포인터 배열을 사용해 객체를 만들어 봅니다.
class Telephone {
private:
  char* name;
  char* phone;
public:
  Telephone(char* pN, char* pP); // 동적 할당 (깊은 복사) ~Telephone(); // 동적 할당 해제
  void Show() const;
int Main(){
 Telephone* ptr[10];
 // ~~~
```

Day09 - 3. 예제 코드

```
//5명의 객체 예제 프로그램
#include <conio.h>
#include <iostream>
using namespace std;
class Telephone {
private:
  char name[20], phone[20];
public:
  Telephone(char* pN, char* pP);
  void Show() const;
};
Telephone::Telephone(char* pN, char* pP) {
  strcpy_s(name,strlen(pN)+1, pN);
  strcpy_s(phone.strlen(pN)+1, pP);
void Telephone::Show() const {
  cout << "이름: " << name << "\t전화번호: " << phone
<< endl;
int main() {
  Telephone* pT[5]; //객체 포인터 배열
  //포인터명 - pT[0], pT[1], pT[2], pT[3], pT[4]
  char name[20], phone[20];
  cout << "\n ** 전화번호 입력 ** \n";
  for (int i = 0; i < 5; i++) {
    cout << "\n이 름:";
    cin >> name;
    cout << "전화번호: ";
    cin >> phone;
     pT[i] = new Telephone(name, phone); //동적객체 생성
시 인수 전달
```

```
}
cout << "\n\n\t** 전화번호 출력**\n\n";
for (int i = 0; i < 5; i++) pT[i]->Show(); //객체주소->멤버
(객체.멤버)
cout << endl << endl;
for (int i = 0; i < 5; i++) { // [] - 포인터가 가리키는 메모리
가 배열인 경우
delete /*[]*/ pT[i]; //delete 포인터;
cout << "동적 메모리 해제\n";
}
return 0;
}
```

Day10 - Contents

- 1. string class
- 2. 예제 코드

String : 문자열 Class

- C style char str[15] = "Hello World!(\0)"; (큰 따옴표 : "문자열 상수(\0)", 작은 따옴표 : '문자 상수 값')

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
str	Н	е	l	l	O		W	0	r	1	d	!	\0		

- C++ style string str1("Hello World!"), str2="Hello World!";

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
str1, 2	Н	е	1	1	0		W	0	r	1	d	!			

```
#include <string>
namespace std {
  class string {
    unsigned __int64 size; // 저장된 문자열의 개수
    unsigned __int64 capacity; // 저장 가능한 총 용량
    allocator<char> allocator; // 배열에 대한 메모리 할당 및 해제를 관리
  };
}
```

String : 문자열 Class

- Constructor (생성자)

Constructor(생성자)	의미					
객체명()	크기가 0인 default 생성자					
객체명(const char * s)	"문자열 상수(\0)"					
객체명(const string& str)	string 객체의 복사 생성자					
객체명(size_type n, char c)	문자 C를 n개 구성한 문자열					
객체명(const char * s, size n)	문자열 상수에서 n개까지 초기화 s의 문자열 길이 < n 가능					
template <class iter=""> 객체명(Iter begin, Iter end)</class>	begin ≤ 문자열 < end 초기화 Iterater : 반복자, 포인터					
객체명(const string& str, size pos, size n=npos)	str[pos] ≤ 문자열 < str[n]					

```
#include <iostream>
#include <string>
using namespace std;
int main() {// 생성자
 char ary[] = "The sun will shine on us again.";
 string str1;// string()
 string str2("hi!");//string(const char * const s)
 string str_2(ary);//string(char * s)
 string str3(str2);// string(const string &str)
 string str4(5, 'k');// string(size_type n, char c);
 string str5("Korea", 10);// string(const char* s, size_type n)
 string str6(ary + 4, ary + 24);// string(Iter begin, Iter end),(char* begin, char* end)
 string str7(str_2, 4, 14);//string(const string& str, size pos, size n=npos)
 string str_7("new world string", 4, 5);//string(const char * str, size pos, size n=npos)
 cout << "=======" << endl;
 cout << "str 01" << str1 << endl;
 cout << "str 02" << str2 << endl;
 cout << "str 002" << str 2 << endl;
 cout << "str 03" << str3 << endl;
 cout << "str 04" << str4 << endl;
 cout << "str 05" << str5 << endl;
 cout << "str 06" << str6 << endl;
 cout << "str 07" << str7 << endl;
 cout << "str 007" << str_7 << endl;
 cout << "=======" << endl:
 return 0;
```

String : 문자열 Class

String(문자열) 입력

```
- C style
char name[15];
 cin >> name; // 공백문자 전까지 14(+'\0')개 저장 가능. [ 큐에 '\n' 남김 ]
while (cin.get() != '\n');
 cin.get(name, 15); // 1행 입력 받기 [ 큐에 '\n' 남김 ]
 cin.getline(name, 15); // 1행 입력 받기 [ '\n'(Enter) 버림 ]
 cin.getline(name, 15, ':'); // 1행 입력 받기 [ ':' 버림 ]
- C++ style
입력 받은 문자열 자동 크기 조절
 string fullname;
 cin >> fullname; // 공백문자 전까지 저장 가능. [ 큐에 '\n' 남김 ]
 while (cin.get() != '\n');
 getline(cin, fullname); // 1행 입력 받기 [ '\n'(Enter) 버림 ]
 getline(cin, fullname, ':'); // 1행 입력 받기 [ ':' 버림 ]
```

제한 요소 : unsigned $_{-}$ int64 (2^{64})저장 가능한 문자열 개수, 메모리 크기

String : 문자열 Class

String(문자열) Method(멤버 함수) 객체. Method();

문자열 위치

- char& at (size_t n): 문자열 n칸 반환 (0 ≤ 문자열 < End) char& operator[] (size_t n): 문자열 n칸 반환 (0 ≤ 문자열 < End)
- char& front (): 문자열 [0]칸 반환
- char& back () : 문자열 [End]칸 반환 문자열 길이
- size_t size () const == size_t length () const : 문자열 길이 반환
- void resize (size_t n): 문자열 길이 n개 맞춤 (str.size() < n: 공백, n < str.size(): 삭제) void resize (size_t n, char c): 공백을 문자 c로 설정
- string substr (size_t st = 0, size_t len = npos) const : st부터 len개 반환
- string& replace (size_t st, size_t len, const string& str): st~len 위치로 str 끼워 넣기
- bool empty () const : 비었는지 확인하는 함수 (size == 0 : true) 수정
- void clear (): 문자열을 지우는 함수 (capacity 유지)
- iterator erase (iterator st, iterator end): st~end 문자열 제거
- iterator erase (iterator iter) : 문자열[iter] 제거
- string& erase (size_type offset = 0, size_type count = npos): offset부터 count개 제거
- void push_back (char c): 맨 뒤에 문자 c 추가 함수
- void pop_back (): 맨 뒤에 있는 문자 1개 제거 함수

String : 문자열 Class

String(문자열) Method(멤버 함수) 객체. Method();

문자열 메모리

- size_t capacity () const; 문자열 저장 가능한 용량(메모리 크기, bytes) 반환
- void shrink_to_fit (): capacity(메모리)를 줄이는 함수
- void reserve (size_t n = 0): capacity를 할당 함수 (메모리 확보)

C style 변환

- const char* c_str () const : 문자열(+ \0) 상수 반환 [C style]
- const value_type * data () const noexcept; 문자열을 null로 끝나는 문자 배열로 변환
- value_type * data () noexcept;

문자열 처리

- int compare (const string& str2) const : 문자열 비교 (객체.변수 : > +, = 0, < -반환)
- int compare (size_t st, size_t len, const string& str2) const
- int compare (size_t st, size_t len, const string& str2, size_t st2, size_t len2) const
- void swap (string& str1, string& str2) : str1과 str2를 교환
- size_t copy (char* arr, size_t len, size_t st = 0) const : 복사, 복사된 길이 반환
- size_t find (const string& str, size_t st = 0) const : 객체 내 문자열 검색, 인덱스 반환
- size_t find (const char* arr, size_t st = 0) const : 객체 내 문자열 검색, 실패 npos 반환

- const char& operator (const size_t n) const :

String : 문자열 Class

```
String(문자열) Method(멤버 함수) 객체. Method();
문자열 위치 접근 iterator (반복자, 포인터 주소)
 const_iterator begin () const : 객체[0]의 주소 반환 [반복자(iterator 포인터)]
- const_iterator end () const : 객체[End+1] 주소 반환 [반복자(iterator 포인터)]
문자열 연산자 오버로딩
String + String
- string& operator+(char _ch):
- string& operator+(const char * const _ptr):
- string& operator+(const string & _Right):
- string& operator+(initializer_list<char> & _ilist):
String = String
- string& operator=(char _ch):
- string& operator=(const char * const _ptr) :
- string& operator=(const string & _Right):
  string& operator=(initializer_list<char> & _ilist):
String[n]
 char& operator (const size_t n):
```

Hangman Game (String class)

- 1. 제시 단어 랜덤 선택
- 2. 글자 수만큼 밑줄(_) 표시
- 3. 알파벳 26개 중 1개 입력 받기
- 4. 알파벳이 있을 경우 알파벳 표시, 없을 경우 기회(life) 1개 소모.
- 5. 게임 진행하는 동안 찾기 실패한 알파벳 목록 표시 중복된 입력의 경우 예외처리, 재입력
- 6. 총 기회(life) 10번

단어 입력 [string 배열 or 파일 입출력] 제시 단어 랜덤 선택 & 숨김(_) 기능 알파벳 1개를 입력 받아 제시 단어에서 일치하는 위치 표시 게임 진행 상황 밑줄 단어로 표시 찾기 실패한 단어 목록 출력

알파벳을 모두 찾아 맞추면 성공 10번의 기회를 다 소모하면 실패

```
#include <string>
using namespace std;
class hangman {
private:
 static string* words; // 단어 목록 동적 할당
 static int wordsCnt; // 단어 목록 총 수량
 string answer; // 게임 진행 답안
 string hideAnswer; // 게임 진행 숨긴 답안
 string bad; // 실패 기록 답안 목록
 int Life; // 총 기회 10번
public:
 ~hangman() { } // words 동적 할당 해제
 void init() { } // 멤버 변수 초기화
 int lifeCheck() { } // life 값 확인
 void printAnswer() { } // 답안 진행 상태 출력
 void menu(int& menu) { } // 게임 메뉴판 출력
 bool isSolve() { } // 정답 확인 함수
 bool guesses(char letter) { } // 입력받은 알파벳 & 답안 비교
 // 알파벳을 실패 기록에서 유무 체크
 // 알파벳을 정답 단어에서 유무 체크
 // 정답 단어에 있으면 단어 위치에 표시 [ true 반환 ]
 // 정답 단어에 없으면 실패 기록에 추가 & Life - 1 [ false 반환 ]
```

string word[] = { "inform", "advise", "service", "provide", "relationship", "behave", "improve", "individual", "require", "million",

common de la control de la con social", "amount", "employ", "terrible", "attitude", "research", "audience", "volunteer", "influence", "international",

```
/////Hangman Game(String class)
/////1. 제시 단어 랜덤 선택
/////2. 글자 수만큼 밑줄(_) 표시
//////3. 알파벳 26개 중 1개 입력 받기
/////4. 알파벳이 있을 경우 알파벳 표시. 없을 경우 기회(life) 1개 소모.
/////5. 게임 진행하는 동안 찾기 실패한 알파벳 목록 표시
///// 중복된 입력의 경우 예외처리, 재입력
/////6. 총 기회(life) 10번
#include <iostream>
#include <cstdlib>
#include <string>
using namespace std;
enum menu { Play = 1, Terminate, Cheat, ReplaceWord };
string wordlist[] = { "apiary", "beetle", "cereal"
"danger", "ensign", "florid", "garage", "health", "insult", 
"jackal", "keeper", "loaner", "manage", "nonce", "onset",
 "plaid", "quilt", "remote", "stolid", "train", "useful",
 "valid", "whence", "xenon", "yearn", "zippy" };
class hangman {
 static string* words: // 단어 목록 동적 할당
static int wordsCnt: // 단어 목록 총 수량
string answer: // 게임 진행 답안
 string hideAnswer; // 게임 진행 숨긴 답안
 string bad: // 실패 기록 답안 목록
int Life; // 총 기회 10번
 ~hangman() { // words 동적 할당 해제
 if (words != NULL) delete[] words;
 void init() { // 멤버 변수 초기화
 if (wordsCnt == 0) {
   wordsCnt = sizeof(wordlist) / sizeof(wordlist[0]);
   words = new string[wordsCnt];
   for (int i = 0; i < wordsCnt; i++) {
     words[i] = wordlist[i];
  answer = words[rand() % wordsCnt];
  string temp(answer.length(), '_');
  hideAnswer = temp;
  Life = 10;
  bad.clear();
 int lifeCheck() { return Life; } // life 값확인
 void printAnswer(int n = 0) { // 답안 진행 상태 출력
 if (n == 3) cout << answer << endl;
  cout << hideAnswer << endl;
 void menu(int& menu) { // 게임 메뉴판 출력
  cout << "[ Hangman Game !! ] \n";
  cout << " 1. Play \n";
  cout << " 2. Exit \n";
  cout << " 3. Cheat\n";
  cout << " 메뉴를 입력하세요. : ";
  do {
   while (!(cin >> menu)) {
    cin.clear();
    while (getchar() != '\n');
    cout << " 메뉴를 입력하세요. : ";
   while (getchar() != '\n');
  } while (menu <= 0 || 4 <= menu);
 bool isSolve() { // 정답 확인 함수
 if (answer.compare(hideAnswer) == 0) {
   return true;
  return false;
```

```
bool guesses(char letter) {// 입력받은 알파벳 & 답안 비교
  if (bad.find(letter) != string::npos || hideAnswer.find(letter) != string::npos) {
   cout << " 이미 시도한 알파벳 입니다. \n";
   return false;
  else {
   int pos = answer.find(letter);
   if (pos == string::npos)
    cout << " 틀렸습니다. \n\n";
    bad += letter; // add to string
    else
    cout << " 찾았습니다. \n\n";
     hideAnswer[pos] = letter;
     pos = answer.find(letter, pos + 1);
    } while (pos != string::npos);
    if (!isSolve()) {
    if (bad.length() > 0)
    cout << "실패한 알파벳 : " << bad << endl:
cout << Life << " 번 기회가 남았습니다.\n\n";
    return false;
   return true;
string* hangman::words;
int hangman::wordsCnt;
int main() {
srand(time(0)):
 hangman game;
int menu = 0:
 while (1) {
  system("cls");
  game.menu(menu);
  switch (menu) {
  case Cheat:
  case Play:
   game.init();
   while (!game.isSolve() && game.lifeCheck() > 0) {
    game.printAnswer(menu);
    char letter:
    cin >> letter;
    while (cin.get() != '\n');
    game.guesses(letter);
   break:
  case Terminate:
   cout << "\n [ 프로그램 종료 합니다. ] \n";
   exit(0);
  if (game.isSolve()) {
   game.printAnswer();
   cout << "축하합니다. 정답을 맞추셨습니다.!! \n";
  else cout << " ㅠ.ㅠ 정답을 맞추지 못 했습니다. \n";
  system("pause");
return 0
```

```
/////Hangman Game(String class)
/////1. 제시 단어 랜덤 선택
/////2. 글자 수만큼 밑줄(_) 표시
/////3. 알파벳26개중 1개 입력 받기
/////4. 알파벳이 있을 경우 알파벳 표시, 없을 경우 기회(life) 1개 소모.
/////5. 게임 진행하는 동안 찾기 실패한 알파벳 목록 표시
///// 중복된 입력의 경우 예외처리, 재입력
/////6. 총 기회(life) 10번
#include <fstream>
#include <iostream>
#include <cstdlib>
#include <string>
using namespace std;
enum menu { Play = 1, Terminate, Cheat, ReplaceWord };
class hangman {
private:
static string filename;
 static int wordsCnt;
 string words[1000];
 string answer;
 string hideAnswer:
 string bad;
int Life;
public:
hangman() {
  srand(time(0));
  cout << "게임에 사용할 단어 파일 이름을 입력하세요.(.txt 확장자 포함, End 종료):";
  while (getline(cin. filename)) {
  ifstream readFile;
   string end("end"), End("End");
  if ((filename.compare(end) == 0) || (filename.compare(End) == 0)) exit(0);
  readFile.open(filename); //파일열기
   if (readFile.is_open()) { //파일이 열렸는지 확인
    for (wordsCnt = 0; !readFile.eof(); wordsCnt++) { //파일 끝까지 읽었는지 확인
     string str:
     getline(readFile, str);
     words[wordsCnt] = str;
     //cout << "단어 저장 " << wordsCnt + 1 <<" : " << words[wordsCnt] << end];
    readFile.close(); //파일 닫기
    break
   else {
    cout << "게임에 사용할 단어 파일 이름을 입력하세요.( .txt 확장자 포함, End 종료) :";
    continue;
 void init() {
  Life = 10;
  bad.clear();
  printFile();
  randomSelect();
 int lifeCheck() { return Life; }
 static void printFile() {
  cout << "참고하는 단어 파일은" << filename << "입니다.\n";
  cout << " 단어 개수 : " << wordsCnt << endl << endl;
 void printAnswer(int n=0) {
 if (n == 3) cout << answer << endl;
  cout << hideAnswer << endl;
 void menu(int& menu) {
 cout << "[ Hangman Game !! ] \n";
  cout << " 1. Play \n";
  cout << " 2. Exit \n";
  cout << " 3. Cheat\n";
  cout << " 4. ReplaceWord\n";
  cout << "메뉴를 입력하세요. : ";
```

```
do{
  while (!(cin >> menu)) {
   cin.clear();
   while (getchar() != '\n');
   cout << " 메뉴를 입력하세요. : ";
  while (getchar() != '\n');
 } while (menu <= 0 || 5 <= menu);
void saveWord() {
 cout << "게임에 사용할 단어 파일 이름을 입력하세요.( .txt 확장자 포함, End 종료) :";
 while (getline(cin, filename)) {
  ifstream readFile;
  string end("end"), End("End");
  if ((filename.compare(end) == 0) || (filename.compare(End) == 0))
  readFile.open(filename); //파일열기
  if (readFile.is_open()) { //파일이 열렸는지 확인
   for (wordsCnt = 0; !readFile.eof(); wordsCnt++) { //파일 끝까지 읽었는지 확인
    getline(readFile, str);
    words[wordsCnt] = str;
    //cout << "단어 저장 " << wordsCnt + 1 <<" : " << words[wordsCnt] << endl;
   readFile.close(); //파일 닫기
   break;
   cout << "게임에 사용할 단어 파일 이름을 입력하세요.( .txt 확장자 포함, End 종료) :";
   continue;
string& randomSelect() {
 answer = words[rand() % wordsCnt];
 int length = answer.length();
 string temp(length, '_');
 hideAnswer = temp;
 return answer:
bool isSolve() {
if (answer.compare(hideAnswer) == 0) {
  return true;
 return false;
bool guesses(char letter) {
 if (bad.find(letter) != string::npos || hideAnswer.find(letter) != string::npos) {
  cout << " 이미 시도한 알파벳 입니다. \n";
  return false;
 else {
  int pos = answer.find(letter);
  if (pos == string::npos)
   cout << " 틀렸습니다. \n\n";
   bad += letter; // add to string
   Life--:
  else
   cout << " 찾았습니다. \n\n";
   do {
    hideAnswer[pos] = letter;
    pos = answer.find(letter, pos + 1);
   } while (pos != string::npos);
  if (answer != hideAnswer) {
   if (bad.length() > 0)
```

```
cout << "실패한 알파벳 : " << bad << endl;
    cout << Life << " 번 기회가 남았습니다.\n\n";
    return false;
   return true;
int hangman::wordsCnt;
string hangman::filename;
int main() {
hangman game;
 int menu = 0;
 while (1) {
  system("cls");
  game.menu(menu);
  switch (menu) {
  case Cheat:
  case Play:
   game.init();
   while (!game.isSolve() && game.lifeCheck() > 0) {
    game.printAnswer(menu);
    char letter;
    cin >> letter;
    while (cin.get() != '\n');
    game.guesses(letter);
   break;
  case Terminate:
   cout << "\n [ 프로그램 종료 합니다. ] \n";
   exit(0);
  case ReplaceWord:
   game.saveWord();
   continue;
  if (game.isSolve()) {
   game.printAnswer();
   cout << "축하합니다. 정답을 맞추셨습니다.!! \n";
  else cout << " ㅠ.ㅠ 정답을 맞추지 못 했습니다. \n";
  system("pause");
return 0;
```

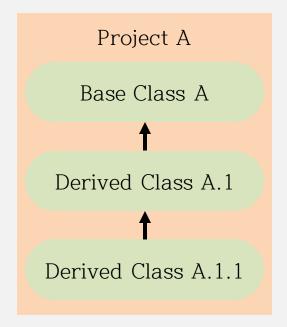
Day11 - Contents

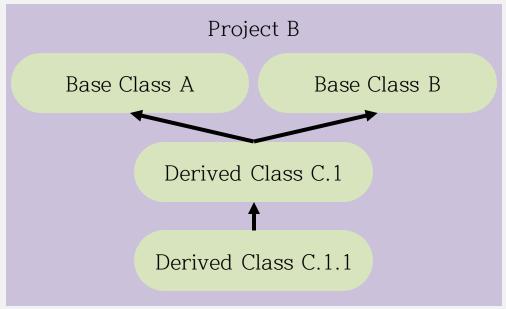
- 1. 상속(Inheritance)
 - 개념
 - 상속을 위한 조건
 - protected 멤버
- 2. 예제 코드

클래스 상속(class inheritance) (: 클래스의 재사용) 기존의 클래스에서 (모든 멤버 변수, 함수)를 물려받고, 새로운 기능(함수)이나 속성(데이터)을 추가해 새로운 클래스를 구현

베이스, 상위, 슈퍼, 기초, 부모 <mark>클래스</mark> (base, super, parent class) : 기존에 정의한 클래스

파생, 하위, 자식 <mark>클래스</mark> (derived, sub, child class) :상속을 통해 기초 클래스의 모든 특성을 물려받아 새롭게 작성된 클래스



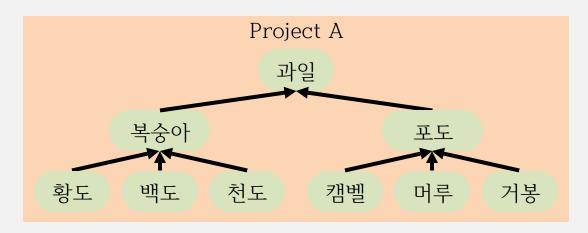


```
형식
Class 파생클래스이름: 접근제어지시자 기초클래스이름 [, 접근제어지시자
기초클래스이름, …(다중 상속)]{
 // 파생 클래스 멤버 리스트…
특징
(1) 기존에 작성된 클래스를 효과적으로 재활용 할 수 있다.
(2) 클래스 간의 계층적 관계를 구성함 (모듈화, 다형성의 기반)
(3) 공통적인 부분은 기초 클래스에 설계
  새로운 데이터, 연산은 파생 클래스에 설계 (중복이 없도록)
(4) 접근 제어 지시자: private, protected, public
  (접근 제어 지시자를 생략하면, private로 기본 설정)
(5) 기초 클래스가 하나인 파생 클래스는 단일 상속(single inheritance)
(6) 여러 개의 기초 클래스를 상속받으면 다중 상속(multiple inheritance)
  ( 쉼표(,)를 사용하여 상속받을 기초 클래스를 여러 개 가능, 권장 X )
(7) 파생 클래스: 반드시 자신만의 생성자를 필요하다
(8) 파생 클래스: 기초 클래스의 접근할 수 있는 모든 멤버 변수들이 저장
(9) 파생 클래스: 기초 클래스의 접근할 수 있는 모든 멤버 함수를 사용 가능
(10)파생 클래스: 필요한 만큼 멤버, 함수를 추가 가능
```

상속을 적용 할 수 있는 조건

(1) Is-A 상속관계 (A는 B 이다. - 역은 성립하지 않는다.)

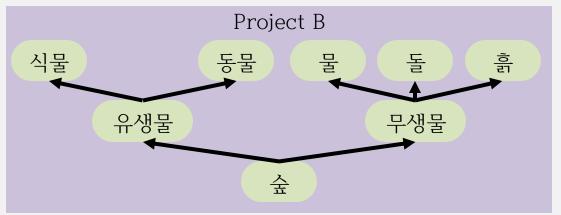
학생은 사람이다 사과는 과일이다. 토끼는 동물이다.



(2) Has-A 상속관계 (A는 B를 포함한다. - 포함관계)

교실에 책상도 있다.

숲에 나무도 있다. 상품에 제조일자도 있다.



```
#include <iostream>
#include <cstring>
using namespace std;
class Person {
private:
  char* name_;
  int age_;
public:
  Person(char* name, int age) {
    int len = strlen(name) + 1;
    name_ = new char[len];
    strcpy_s(name_,strlen(name)+1, name);
    age_ = age;
    cout << "베이스 클래스 생성 완료\n";
  ~Person() {
    delete[] name;
     cout << "베이스 클래스 소멸 완료\n";
  void ShowPersonInfo() { cout << "name :" << name_ << "\nage :" << age_ << endl; }</pre>
class Student : public Person {
private:
  int student_id_;
public:
  Student(int sid, char* name, int age) :Person(name, age) {
     student_id_ = sid;
    cout << "파생 클래스 생성 완료\n";
  ~Student() { cout << "파생 클래스 소멸 완료\n"; }
  void ShowStdInfo() { cout << "Student number :" << student_id_ << endl; this->ShowPersonInfo(); }
};
int main() {
  Student std_1(20230000, (char*)"michelin", 20);
  std_1.ShowStdInfo();
  return 0:
```

Project A

Base Class: Person



Derived Class: Student

class 접근제어지시자

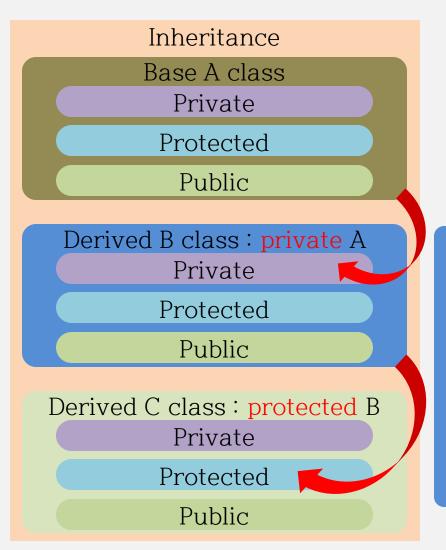
private: 클래스 내에서만 접근허용 protected: 상속관계에서 접근허용

public: 어디서든 접근허용

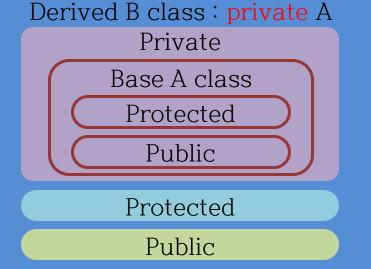
Program Class Private Protected 멤버가 선언된 클래스 ~ 파생 클래스 외부에서 접근 불가 Public 어디서나 접근 가능

```
#include <iostream>
using namespace std;
class Base {
private: //선언된 클래스
  int a;
protected: //선언된 클래스 + 상속된 클래스
  int b:
public:
  void Set_a(int_a) \{ a = a; \}
  void Set_b(int_b) \{ b = b; \}
  int Get_a() { return a; }
  int Get_b() { return b; }
class Derived: public Base {
private:
  int c;
public:
  void Set(int _a, int _b, int _c) {
    C = C;
    //a = _a; //a는 private 멤버로 상속받아도 접근 불가능
    Set_a(_a); //a를 초기화 해주는 Set_a함수를 통해 a값 초기화
    b = _b; //b는 protected 멤버로 상속 받아 접근 가능
  int Mul() {
    //return a * b * c; //a는 private 멤버로 상속받아도접근 불가능
    return Get_a() * b * c; //a * b * c 값을 리턴하는 함수호출
};
int main() {
  Derived ob:
                 //객체 생성 시 상속된 멤버까지 모두 메모리 할당
  ob.Set(5, 4, 5);
  cout << ob.Mul() << endl;
  return 0;
```

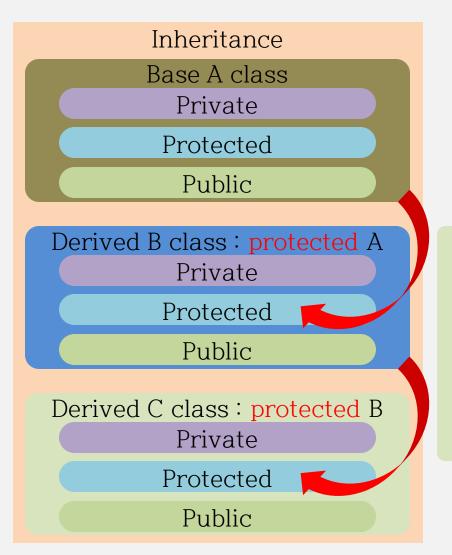
상속(Inheritance) 접근제어지시자 (private, protected, public)



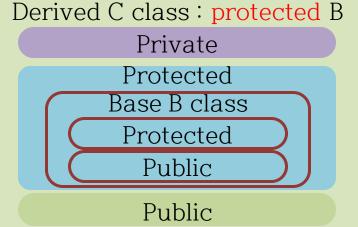
Base A class
Private



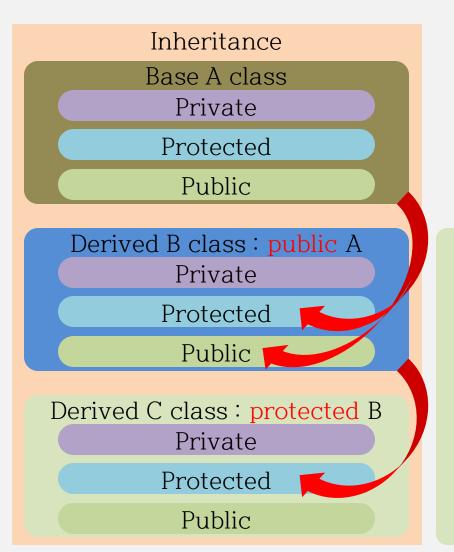
상속(Inheritance) 접근제어지시자 (private, protected, public)



Base A class
Private



상속(Inheritance) 접근제어지시자 (private, protected, public)



Base A class
Private

Private

Protected

Base B class

Protected

Public

Base B class

Public

Public

Product Class 를 활용한 제품 등록 프로그램 작성, 수량 입력 -> 제품 등록

```
class Product : public Date {
class Date {
                                protected:
protected:
                                 char* product_name;
 //년, 월, 일
                                 char* manufacturer;
 int year, month, day;
                                 int price;
public:
                                public:
 //년, 월, 일 설정 함수 set
                                 ~Product() {}
 void Set_year(int _year) {}
                                 //init_Product : 상품명, 제조사, 가격, 년, 월, 일 초기화
 void Set_month(int _month) {} void init_Product(char* name, char* manufact, int
 void Set_day(int _day) {}
                                _price, int year, int month, int day) {}
                                 //상품명, 제조사, 가격, 제조일자 설정 함수 set
 //년, 월, 일 리턴 함수 get
                                 void Set_product_name(char* name) {}
 int Get_year() {}
                                 void Set_manufacturer(char* name) {}
 int Get_month() {}
                                 void Set_price(int _price) {}
 int Get_day() {}
                                 void Set_date(int year, int month, int day) {}
                                 //상품명, 제조사, 가격 리턴 함수 get
                                 char* Get_product_name() {}
                                 char* Get_manufacturer() {}
                                 int Get_price() {}
                                 //상품정보 출력 함수
                                 void show_info() {}
                                };
```

```
#include <iostream>
                                                     string Get manufacturer() { return manufacturer; } getlineclear();
#include <string>
                                                    int Get_price() { return price; }
using namespace std;
                                                    void show info() {
                                                      cout << endl << endl;
class Date {
                                                      cout << "상 품 명 : " << Get_product_name()
private:
 int year, month, day;
                                                      cout << "제 조 사:" << Get manufacturer()
public:
                                                   << endl:
 void Set_year(int _year) { year = _year; }
                                                      cout << "가 격: " << Get_price() << endl;
 void Set_month(int _month) { month = _month; }
                                                      cout << "제조일자: " << Get_year() << ","
 void Set_day(int _day) { day = _day; }
                                                   << Get_month() << "," << Get_day() << endl << endl;
 int Get_year() { return year; }
 int Get_month() { return month; }
 int Get day() { return day; }
                                                   void cinclear() {
class Product : Date {
                                                    if (cin.fail()) {
                                                      cin.clear();
private:
                                                      while (cin.get() != '\n');
 string product_name;
 string manufacturer;
 int price;
                                                     while (cin.get() != '\n');
public:
 void init_Product(string name, string manufact,
                                                   void getlineclear() {
int _price, int year, int month, int day) {
                                                    if (cin.fail()) {
  product_name = name;
                                                      cin.clear();
                                                      while (cin.get() != '\n');
  manufacturer = manufact;
  price = _price;
  Set_year(year);
  Set_month(month);
                                                   int main()
  Set_day(day);
                                                    int price, year, month, day, num;
                                                    cout << "등록하실 제품의 수량을 적어주세요.:";
 void Set_product_name(string product)
{ product_name = product; }
                                                    cin >> num;
 void Set_manufacturer(string _manufacturer)
                                                    cinclear();
{ manufacturer = _manufacturer; }
                                                    Product* product = new Product[num]://객체생성
 void Set_price(int _price) { price = _price; }
                                                    for (int i = 0; i < num; i++) {
 void Set_date(int year, int month, int day) {
                                                      string product_name, manufacturer;
  Set_year(year);
                                                      cout << "[ 제품 등록 ] \n\n";
                                                      cout << "등록하실 제품의 이름:";
  Set_month(month);
  Set_day(day);
                                                      getline(cin, product_name);
                                                      getlineclear();
                                                      cout << "등록하실 제품의 제조사:";
 string Get_product_name() { return
                                                      getline(cin, manufacturer);
product_name; }
```

```
} getlineclear();
cout << "등록하실 제품의 가격:";
cin >> price;
cinclear();
cout << "등록하실 제품의 제조일자(년, 월, 일):";
cin >> year >> month >> day;
cinclear();
product[i].init_Product(product_name,
manufacturer, price, year, month, day);
}
for (int i = 0; i < num; i++) {
product[i].show_info();
}
delete[] product; //객체소멸
return 0;
}
```

```
#include <iostream>
                                                    manufacturer = new char[strlen(manufact) + 1]; };
                                                    strcpy_s(manufacturer, strlen(manufact) + 1.
#include <cstring>
                                                                                                   int main()
using namespace std;
                                                 manufact);
                                                    price = _price;
class Date {
                                                                                                    int price, year, month, day;
private:
                                                    Set_year(year);
                                                                                                    char buf[100];
int year, month, day;
                                                    Set_month(month);
                                                                                                    char* product_name, * manufacturer;
                                                                                                    Product* p1 = new Product; //객체생성
public:
                                                    Set dav(dav);
                                                                                                    Product* p2 = new Product; //객체생성
 void Set_year(int _year) { year = _year; }
                                                  void Set_product_name(char* name) {
                                                                                                    p1->Set_Product("촉촉한 초코칩", "오리온", 13900,
 void Set_month(int _month) { month = _month; }
 void Set_day(int _day) { day = _day; }
                                                    delete product_name;
                                                                                                   2023, 2, 27);//상품명, 제조사, 가격, 제조년월일 출력
int Get_year() { return year; }
                                                    product_name = new char[strlen(name) + 1];
                                                                                                    p1->show_info();
                                                    strcpy_s(product_name, strlen(name) + 1, name); cout << "상 품 명: ";
 int Get_month() { return month; }
int Get day() { return day; }
                                                                                                    cin >> buf;
                                                  void Set_manufacturer(char* name) {
                                                                                                    product_name = new char[strlen(buf) + 1];
class Product : Date {
                                                    delete manufacturer;
                                                                                                    strcpy_s(product_name, strlen(buf) + 1, buf);
                                                    manufacturer = new char[strlen(name) + 1];
                                                                                                    cout << "제 조 사:";
private:
                                                                                                    cin >> buf;
 char* product_name;
                                                    strcpy_s(manufacturer, strlen(name) + 1, name);
 char* manufacturer;
                                                                                                    manufacturer = new char[strlen(buf) + 1];
int price;
                                                   void Set_price(int _price) {
                                                                                                    strcpy_s(manufacturer, strlen(buf) + 1, buf);
                                                                                                    cout << "가 격:";
public:
                                                    price = _price;
 ~Product() {
                                                                                                    cin >> price;
                                                                                                    cout << "제조일자: (띄어쓰기, 공백: 년, 월, 일)";
  delete product_name;
                                                  void Set_date(int year, int month, int day) {
  delete manufacturer;
                                                    Set_year(year);
                                                                                                    cin >> year >> month >> day;
  cout << "파생 클래스 동적 할당 해제 완료" << endl;
                                                    Set_month(month);
                                                                                                    p2->Set_Product(product_name, manufacturer,
                                                                                                   price, year, month, day);//상품명, 제조사, 가격, 제조
                                                    Set_day(day);
                                                                                                   년월일 출력
 void Set_Product(const char* name, const char*
manufact, int _price, int year, int month, int day) {
                                                  char* Get_product_name() { return
                                                                                                    cout << endl << endl;
  product_name = new char[strlen(name) + 1];
                                                 product_name; }
                                                                                                    cout << "객체 2 정보 표시";
  strcpy_s(product_name, strlen(name) + 1, name); char* Get_manufacturer() { return manufacturer; } p2->show_info();
                                                  int Get_price() { return price; }
                                                                                                    p1->Set_product_name(product_name);
  manufacturer = new char[strlen(manufact) + 1];
                                                                                                    p1->Set_manufacturer(manufacturer);
  strcpy_s(manufacturer, strlen(manufact) + 1,
manufact);
                                                  void show_info() {
                                                                                                    p1->Set_price(price);
                                                   cout << endl << endl;
  price = _price;
                                                                                                    p1->Set_date(year, month, day);
                                                    cout << "상 품 명:" << Get_product_name()
                                                                                                    cout << "객체 1 정보 수정";
  Set_year(year);
  Set_month(month);
                                                                                                    p1->show_info();
                                                    cout << "제 조 사:" << Get_manufacturer()
  Set_day(day);
                                                                                                    delete p1; //객체소멸
                                                                                                    delete p2;
                                                 << endl;
                                                    cout << "가 격: " << Get_price() << endl;
 void Set_Product(char* name, char* manufact,
                                                                                                    return 0;
                                                    cout << "제조일자: " << Get_year() << ","
int _price, int year, int month, int day) {
                                                 << Get_month() << "," << Get_day() << endl << endl;
  product_name = new char[strlen(name) + 1];
  strcpy_s(product_name, strlen(name) + 1, name); }
```

Day12 - Contents

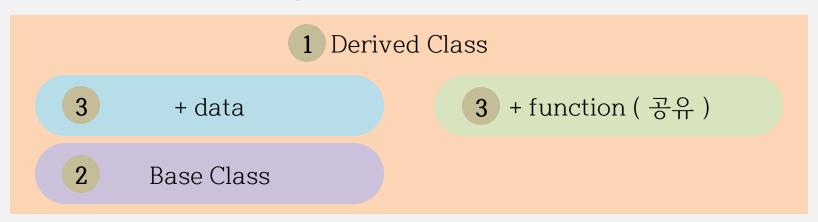
- 1. 상속(Inheritance)
 - 상속관계에서 생성자 호출 순서
 - 다중상속
- 2. 예제 코드

상속관계에서 생성자 호출 순서

파생 클래스의 구성 요소

- Base 클래스
- 추가 속성 (data)
- 추가 기능 (function)

파생 클래스의 객체를 생성한다면 구성 요소를 모두 메모리 할당 Derived_Class obj_example;



파생 객체 접근 -> 베이스 객체 접근 -> 베이스 생성 -> 파생 객체 생성 파생 객체 소멸 -> 베이스 객체 소멸: 생성자 동적 할당 시, 소멸자 해제!! 파생 클래스 생성은 베이스 생성이 필수로 진행된다.

```
#include <iostream>
using namespace std;
class Base {
private:
  int _base;
public:
  Base(): _base(20) { cout << "default 베이스 생성" << endl; }
  ~Base() { cout << "베이스 소멸" << endl << endl; }
  Base(int n): _base(n) { cout << "매개변수 베이스 생성" << endl; }
  void ShowBaseData() { cout << _base << endl; }</pre>
class Derived: public Base {
private:
  int _derived;
public:
  Derived(): _derived(30) { cout << "default 파생 생성" << endl; }
  ~Derived() { cout << "파생 소멸" << endl; }
  Derived(int n): _derived(n) { cout << "매개변수 1 파생 생성" << endl; }
  Derived(int n1, int n2): Base(n1), _derived(n2)
  { cout << "매개변수 2 파생 생성" << endl; }
  void ShowDerviedData() { ShowBaseData(); cout << _derived << endl; }</pre>
int main() {
  Derived dr1;
                      //case1 : default 객체 생성
  dr1.ShowDerviedData(); cout << endl << endl;</pre>
                     //case2 : 매개변수 1개 객체 생성
  Derived dr2(12);
  dr2.ShowDerviedData(); cout << endl << endl;</pre>
  Derived dr3(23, 24); //case2 : 매개변수 2개 객체 생성
  dr3.ShowDerviedData(); cout << endl << endl;
  return 0:
```

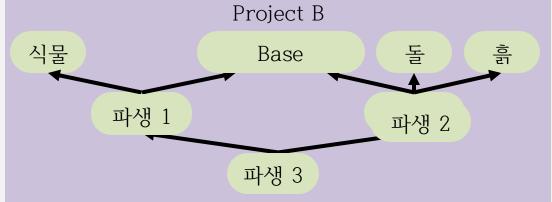
Derived Class 생성 Stack 6 Derived dr3 5 Base Class dr3 2. 4 Derived dr2 3 Base Class dr2 3 4 2 Derived dr1 5 Base Class dr1 6 소멸 순서 생성 순서

```
다중상속 두 개 이상의 베이스 클래스 멤버를 상속받아 파생 클래스를 생성하는 것 형식 Class 파생클래스이름: 접근제어지시자 기초클래스이름, 접근제어지시자 기초클래스이름, "(다중 상속){ // 파생 클래스 멤버 리스트… }
```

상속을 적용 할 수 있는 조건

(1) Has-A 상속관계 (A는 B를 포함한다. - 포함관계) 교실에 책상도 있다.

숲에 나무도 있다. 상품에 제조일자도 있다.



다중 상속의 문제점

여러 개의 기초 클래스가 가진 멤버를 모두 상속받을 수 있다는 점에서 유용한 방법이지만 단일 상속에 비해 다양한 문제를 발생.

- (1) 상속받은 여러 기초 클래스에 같은 이름의 멤버가 존재할 가능성 (A::a = _a; B::a=_a; C::a=_a;)
- (2) 하나의 클래스를 간접적으로 두 번 이상 상속받을 가능성 (virtual public Base)
- (3) 가상 클래스가 아닌 기초 클래스를 다중 상속하면, 기초 클래스 타입의 포인터로 파생 클래스를 가리킬 수 없습니다.

점점 복잡해지는 프로그램에서 다중 상속은 활용도에 비해 유지 관리가 매우어렵다. (실용성이 낮다.)

잘 설계된 클래스는 고객사의 요청 사항을 유연하게 대응할 수 있다. (약간의 수정으로 요청사항을 반영 할 수 있다.)

```
<< endl; }
#include <iostream>
using namespace std;
                                                          ~Derived() { cout << "파생 소멸" << endl; }
                                                          Derived(int n): _derived(n){
class Base 1 {
private:
                                                           cout << "매개변수 1 파생 생성" << endl; }
 int base 1;
                                                          Derived(int n1, int n2): Base_1(n1), _derived(n2){
                                                           cout << "매개변수 2 파생 생성" << endl; }
public:
 Base_1(): _base_1(20){ cout << "default 베이스 1 생성"
                                                          Derived(int n1, int n2, int n3): Base_1(n1), Base_2(n2),
<< endl; }
                                                         _derived(n3){
 ~Base_1() { cout << "베이스 1 소멸" << endl << endl; }
                                                           cout << "매개변수 3 파생 생성" << endl; }
 Base_1(int n): _base_1(n) { cout << "매개변수 베이스 1 생성
                                                          void ShowDerviedData(){
" << endl; }
                                                           Base_1::ShowBaseData(); Base_2::ShowBaseData(); cout
 void ShowBaseData() { cout << _base_1 << endl; }</pre>
                                                         << derived << endl; }
};
class Base_2 {
                                                         int main() {
private:
                                                          Derived dr1;
                                                                              //case1 : default 객체 생성
 int_base_2;
                                                          dr1.ShowDerviedData(); cout << endl << endl;</pre>
                                                          Derived dr2(12);
                                                                               //case2 : 매개변수 1개 객체 생성
public:
 Base_2(): _base_2(40) { cout << "default 베이스 2 생성"
                                                          dr2.ShowDerviedData(); cout << endl << endl;</pre>
<< endl; }
                                                          Derived dr3(23, 24); //case2 : 매개변수 2개 객체 생성
 ~Base_2() { cout << "베이스 2 소멸" << endl; }
                                                          dr3.ShowDerviedData(); cout << endl << endl;
 Base_2(int n): _base_2(n) { cout << "매개변수 베이스 2 생성
                                                          Derived dr4(23, 24, 25); //case3 : 매개변수 3개 객체 생성
" << endl; }
                                                          dr4.ShowDerviedData(); cout << endl << endl;</pre>
 void ShowBaseData() { cout << _base_2 << endl; }</pre>
                                                          return 0;
};
class Derived: public Base_1, public Base_2 {
private:
 int _derived;
public:
 Derived(): _derived(30){ cout << "default 파생 생성"
```

```
상속 생성자 호출 순서
상속에서 접근지시자 의미 (private 상속, protected 상속, public 상속)
class human { //생성자 작성, 멤버 함수 작성
public: (1), protected: (2), private: (3)
 string name;
 string phone;
class student: public ①, protected ②, private ③ human { //생성자 작성, 멤버 함수 작성
public: ①, protected: ②, private: ③
 unsigned long studentId;
 string major;
class graduate: public ①, protected ②, private ③ student { //생성자 작성, 멤버 함수 작성
public: ①, protected: ②, private: ③
 string job;
 void showStudentInfo() { }
};
int main() {
 graduate s("michelin", "010-1234-5678", 202312345, "Economics", "worker");
 s.showStudentInfo();
 return 0;
```

```
#include <iostream>
                                                                         protected:
#include <string>
                                                                          string job;
using namespace std;
                                                                          public:
                                                                          graduate(): job("") { cout << "파생 파생 클래스: 기본 생성자\n\n"; }
class human {
protected:
                                                                          graduate(const char* pN, const char* pP, unsigned long pId, const char*
                                                                         pMajor, const char* pjob): student(pN, pP, pId, pMajor), job(pjob) { cout << "
 string name;
                                                                         파생 파생 클래스: 매개변수 생성자\n\n"; }
 string phone;
                                                                          graduate(const graduate& s) : student(s), job(s.job) { cout << "파생 파생 클래
public:
human(): name(""), phone("") { cout << "베이스 클래스 : 기본 생성자\n"; }
                                                                         스 : 복사 생성자\n\n"; }
 human(const char* pName, const char* pPhone): name(pName),
                                                                          ~graduate() { cout << "~파생 파생 클래스 소멸자\n\n"; }
phone(pPhone) { cout << "베이스 클래스 : 매개변수 생성자\n\n"; }
                                                                           void showStudentInfo() {
human(const human&r):name(r.name), phone(r.phone) { cout << "베이스클
                                                                            cout << "이 름: " << name << endl;
                                                                            cout << "연락처:" << phone << endl;
래스: 복사 생성자\n\n"; }
 ~human() { cout << "~베이스 클래스 소멸자\n\n"; }
                                                                            cout << "학 번: " << studentId << endl;
                                                                            cout << "전 공: " << major << endl;
 inline const string getName() const { return name; }
 inline const string getPhone() const { return phone; }
                                                                            cout << "직 업: " << iob << endl << endl;
class student : protected human { //human클래스 상속
protected:
                                                                         int main() {
 unsigned long studentId;
                                                                          graduate s("michelin", "010-1234-5678", 202312345, "Economics", "worker");
 string major;
                                                                         //객체 생성
public:
                                                                          s.showStudentInfo();
 student(): studentId(0), major("") { cout << "파생 클래스 : 기본 생성자\n\n"; }
                                                                          graduate s2 = s;
 student(const char* pN, const char* pP, unsigned long pId, const char*
                                                                          s2.showStudentInfo();
pMajor): human(pN, pP), studentId(pId), major(pMajor) { cout << "파생 클래스: return 0;
매개변수 생성자\n\n"; }
 student(const student& s): human(s), studentId(s.studentId), major(s.major)
{ cout << "파생 클래스 : 복사 생성자\n\n"; }
 ~student() { cout << "~파생 클래스 소멸자\n\n"; }
 inline const unsigned long getstudentId() const { return studentId; }
 inline const string getmajor() const { return major; }
 inline const string getbaseName() const { return getName(); }
 inline const string getbasePhone() const { return getPhone(); }
 void showStudentInfo() {
  cout << "이 름: " << getName() << endl;
  cout << "연락처: " << getPhone() << endl;
  cout << "학 번: " << studentId << endl;
  cout << "전 공: " << major << endl;
class graduate: protected student { //student클래스 상속
```

Day13 - Contents

- 1. 상속(Inheritance) & 다형성(Polymorphism)
 - 객체포인터
 - Override
- 2. 예제 코드

객체포인터 객체의 주소 값을 저장하는 포인터 변수 (동적 할당된 메모리는 같은 타입의 포인터가 가리킬 수 있다.)

정수형 메모리를 가리키는 포인터

int *p = new int;

int *p : 정수형 포인터 변수 p

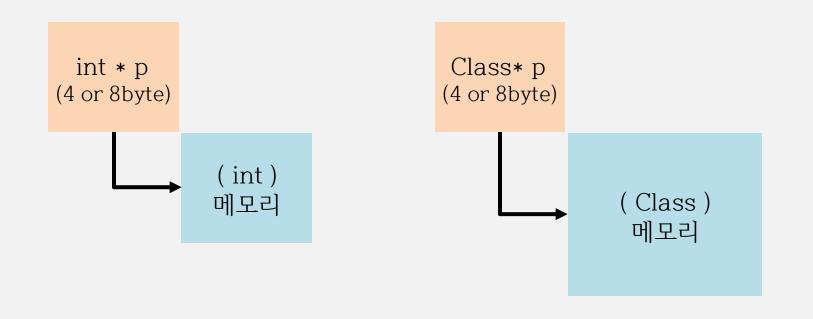
new int: 정수형 메모리 할당 후 주소 값 리턴

객체 메모리를 가리키는 포인터

Class *p = new Class;

Class *p : 객체 포인터 변수 p

new Class: 객체 메모리 할당 후 주소 값 리턴



객체포인터 특징

- 부모의 객체포인터로 자식 객체의 주소를 저장할 수 있다.
- 부모포인터로 자식고유의 멤버에는 접근 불가능하다.

```
#include <iostream>
                                                                         A* p1
using namespace std;
                                                Class 상속 구조
class A {
                                                                         (4 byte)
public:
  void func1() { cout << "call : A\n"; }</pre>
                                                    Class: A
};
                                                                                       Class A
class B:public A {
                                                                                        메모리
public:
  void func2() { cout << "call : B\n"; }</pre>
                                                    Class: B
};
                                                                         A* p2
class C:public B {
                                                                         (4 byte)
public:
  void func3() { cout << "call : C\n"; }</pre>
                                                    Class: C
                                                                                       Class B
int main() {
                                                                                        메모리
  A* ap = new A;
  A* bp = new B;
  A* cp = new C;
                                                                         A* p3
  ap->func1();
                                                                         (4 byte)
  bp->func1(); //bp->func2(); 베이스 포인터 변수로 파생 클래스 멤버 접근 불가
  cp->func1(); //cp->func3(); 베이스 포인터 변수로 파생 클래스 멤버 접근 불가
  delete ap;
                                                                                       Class C
  delete bp;
                                                                                        메모리
  delete cp;
  return 0;
```

02

B

В

03

04

객체포인터 특징

- 부모의 객체포인터로 자식 객체의 주소를 저장할 수 있다.
- 부모포인터로 자식고유의 멤버에는 접근 불가능하다.

```
#include <iostream>
using namespace std;
class A {
public:
                                                                 00
                                                                         01
  void func1() { cout << "call : A\n"; }</pre>
};
class B:public A {
                                                   A* ap
                                                                          Α
public:
  void func2() { cout << "call : B\n"; }</pre>
};
                                                   A* bp
                                                                          Α
class C:public B {
public:
  void func3() { cout << "call : C\n"; }</pre>
                                                   A* cp
                                                                          Α
int main() {
  A* ap = new A;
  A* bp = new B;
  A* cp = new C;
  ap->func1();
  bp->func1(); //bp->func2(); 베이스 포인터 변수로 파생 클래스 멤버 접근 불가
  cp->func1(); //cp->func3(); 베이스 포인터 변수로 파생 클래스 멤버 접근 불가
  delete ap;
  delete bp;
  delete cp;
  return 0;
```

```
오버라이드 (Override, 함수의 재 정의)
함수의 원형(선언문)이 완벽히 일치하는 함수를 다른 기능으로 내용 수정
상속에서 사용하며 부모클래스의 대부분의 기능이 일치하는 경우
일치하지 않은 기능만 재 정의해서 사용할 수 있다.
[ 파생 클래스에서 함수를 덮어쓰기 ]
오버로딩 (Overloading, 함수의 중복정의)
함수의 이름만 같고, 매개변수의 타입, 개수가 다르면 다른 함수로 인식한다.
                           오버라이딩 (Override)
오버로딩 (Overloading)
void fun (int x);
                           void fun(int x) { 정의부분 1 }
void fun (char c);
                           void fun(int x) { 정의부분 2 }
void fun (float x);
                           void fun(int x) { 정의부분 3 }
void fun (int x, int y);
                           void fun(int x) { 정의부분 4 }
```

```
#include <iostream>
using namespace std;
class A {
public:
  void func() { cout << "A::func" << endl; }</pre>
};
                                                                    A* ap
                                            Class 상속 구조
class B: public A {
                                                                   (4 byte)
public:
                                                Class: A
  void func() { cout << "B::func" << endl; }</pre>
                                                                                Class C
                                                                                 메모리
class C: public B {
public:
                                                Class: B
                                                                    B* bp
  void func() { cout << "C::func" << endl; }</pre>
                                                                   (4 byte)
};
int main() {
                                                Class: C
  C*cp = new C;
                                                                                Class C
  B*bp = cp;
                                                                                 메모리
  A*ap = bp; //ap = bp = cp = new C
  ap->func(); //객체 포인터 타입의 멤버 함수로 호출
                                                                    C* cp
  bp->func();
  cp->func();
                                                                   (4 byte)
  delete cp;
  return 0;
                                                                                Class C
                                                                                 메모리
```

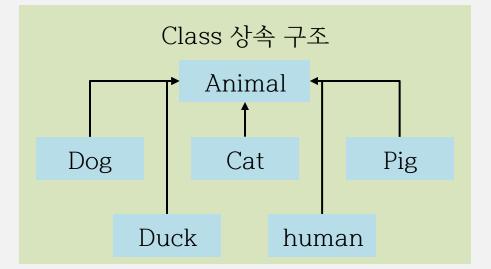
```
#include <iostream>
using namespace std;
class A {
public:
  void func() { cout << "A::func" << endl; }</pre>
};
class B: public A {
                                                           00
                                                                  01
                                                                          02
                                                                                 03
                                                                                         04
public:
  void func() { cout << "B::func" << endl; }</pre>
                                               A* ap
                                                                   Α
class C: public B {
public:
                                               B* bp
                                                                           B
                                                                   Α
  void func() { cout << "C::func" << endl; }</pre>
};
int main() {
                                               C* cp
                                                                           В
                                                                   Α
  C*cp = new C;
  B*bp = cp;
  A*ap = bp; //ap = bp = cp = new C
  ap->func(); //객체 포인터 타입의 멤버 함수로 호출
  bp->func();
  cp->func();
  delete cp;
```

return 0;

```
각각의 동물의 소리와 걸음을 오버라이딩(Override), 객체 포인터를 사용
class Animal {
public:
  void speak()
  void walk()
class Dog: public Animal {
public:
  void speak() //override
};
class Cat: public Animal {
public:
  void speak() //override
class Pig: public Animal {
public:
  void speak() //override
};
class Duck: public Animal {
public:
  void speak() //override
  void walk() //override
```

};

```
class human : public Animal {
public:
  void speak() //override
  void walk() //override
};
 선택한 동물의 소리와 걸음 함수 출력하기
 0. exit
    dog
 2. cat
    pig
 4. duck
    human
```



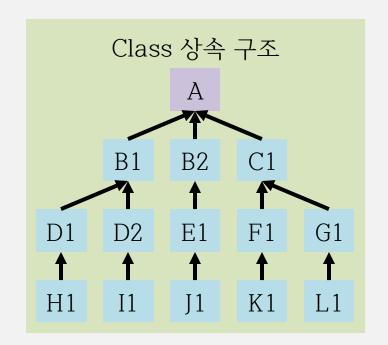
```
#include <iostream>
                                                                                                             phuman->walk();
                                                  int main() {
using namespace std;
                                                     Dog* pDog;
                                                                                                             delete phuman;
class Animal {
                                                     Cat* pCat;
                                                                                                             break;
public:
                                                     Pig* pPig;
                                                                                                          case 0:
  void speak() { cout << "[베이스 클래스] 각각의 동물
                                                     Duck* pDuck;
                                                                                                             exit(0);
의 소리를 여기에 출력합니다.\n"; }
                                                     human* phuman;
  void walk() { cout << "[베이스 클래스] 네발로 걷습니
                                                                                                          cout << "\n\n\t\t";
                                                     while (true) {
다.\n";}
                                                       system("cls");
                                                                                                          system("pause");
};
                                                       cout << "1. dog 2. cat 3. pig 4. duck 5.
class Dog: public Animal {
                                                  human 0. exit\n";
                                                                                                        return 0;
                                                       cout << "choice: ";
public:
  void speak() { cout << "[override] 멍멍!!!\n"; }
                                                       int choice;
//override
                                                       cin >> choice;
                                                       switch (choice) {
class Cat: public Animal {
                                                       case 1:
public:
                                                          pDog = new Dog;
  void speak() { cout << "[override] 야옹!!!\n"; }
                                                          pDog->speak();
//override
                                                          pDog->walk();
                                                          delete pDog;
class Pig: public Animal {
                                                          break;
                                                       case 2:
public:
  void speak() { cout << "[override] 꿀꿀!!!\n"; }
                                                          pCat = new Cat;
                                                          pCat->speak();
//override
};
                                                          pCat->walk();
class Duck: public Animal {
                                                          delete pCat;
                                                          break;
public:
  void speak() { cout << "[override] 꽥꽥!!!\n"; }
                                                       case 3:
//override
                                                          pPig = new Pig;
  void walk() { cout << "[override] 두 발로 걷습니
                                                          pPig->speak();
다.\n";}
                                                          pPig->walk();
};
                                                          delete pPig;
class human : public Animal {
                                                          break;
                                                       case 4:
public:
  void speak() { cout << "[override] 안녕하세요!!!\n"; }
                                                          pDuck = new Duck;
//override
                                                          pDuck->speak();
  void walk() { cout << "[override] 두 발로 걷습니
                                                          pDuck->walk();
다.\n";}
                                                          delete pDuck;
};
                                                          break;
                                                       case 5:
                                                          phuman = new human;
                                                          phuman->speak();
```

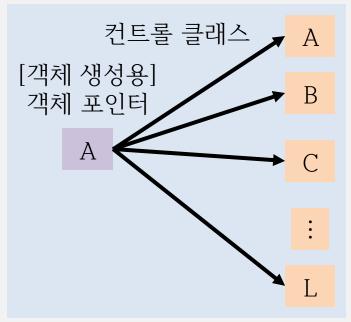
Day14 - Contents

- 1. 상속(Inheritance) & 다형성(Polymorphism)
 - Virtual
 - Virtual 소멸자
- 2. 예제 코드

컨트롤 클래스 (Control Class)

- 프로젝트 內 다양한 클래스의 종류를 모두 총괄적 제어하기 위한 클래스
- 객체 생성용으로 상속의 최상위 클래스를 객체 포인터 타입으로 사용
 (1) 객체 포인터는 파생 클래스를 모두를 가리킬 수 있다. [타입의 제한 X]
 (2) 객체의 추가 & 삭제 할 경우 컨트롤 클래스의 수정이 거의 없다.
 < Up-casting: 상위 클래스 * 변수명= new 하위 클래스; >
- 문제점 객체 포인터 타입에 따라 멤버 함수의 재정의(override)가 다르다.





Virtual

객체 포인터가 가리키는 객체의 클래스를 보고 호출할 함수 결정 최상위 클래스에 virtual 선언시 하위 클래스 모두 가상함수 적용 <객체 포인터 타입과 상관없이!! > 형식 virtual 반환 타입 함수명(매개변수) { 정의부 ···; }

바인딩(Binding): 함수 호출과 정의를 연결하는 방법 정적 바인딩 (Static Binding) 객체나 포인터의 타입을 보고 호출할 함수를 결정한다.

- 컴파일 과정에서 타입에 대한 정보, 호출될 함수가 결정. (속도가 빠르다.)
- 타입 에러로 인한 문제를 조기에 발견할 수 있어 안정적이다.
- 컴파일 과정에서 결정이 되고 그 이후 변경이 불가능

동적 바인딩(Dynamic Binding) 객체의 가상함수 Table을 사용한다.

- 프로그램 실행 중 필요한 객체의 함수를 호출. (객체의 타입에 유연성)
- 변수의 예상치 못한 타입으로 인해 안정성이 저하된다.

C++ 은 기본적으로 정적 바인딩으로 적용되고, 동적 바인딩을 수행 하려면 'virtual'을 사용.

```
//Override 예제
#include <iostream>
using namespace std;
class A{
public:
  void func() { cout << "A::func" << endl; }</pre>
class B: public A{
public:
  void func() { cout << "B::func" << endl; }</pre>
class C: public B{
public:
  void func() { cout << "C::func" << endl; }</pre>
};
int main(){
  C * cp = new C;
  B*bp = cp;
  A*ap = bp; //ap = bp = cp = new C
  //객체 포인터 타입의 멤버 함수로 호출
  ap->func();
  bp->func();
  cp->func();
  delete cp;
  return 0;
```

```
//virtual 예제
#include <iostream>
using namespace std;
class A {
public:
  virtual void func() { cout << "A::func" << endl; }</pre>
class B: public A {
public:
  void func() { cout << "B::func" << endl; }</pre>
};
class C: public B {
public:
  void func() { cout << "C::func" << endl; }</pre>
};
int main() {
  C*cp = new C;
  B*bp = cp;
  A*ap = bp; //ap = bp = cp = new C
  //객체 포인터가 가리키는 클래스의 멤버 함수로 호출
  ap->func();
  bp->func();
  cp->func();
  delete cp;
  return 0;
```

객체의 가상함수 Table			1 2 3	1 //virtual 예제 2 class Animal { 3 public:
Class	Function	Line (함수 주소)	4	1
Animal	speak()	4	$\frac{1}{7}$	
	A∷walk()	5 /	8 9	
Dog	speak()	9 —	10	}; class Cat: public Animal {
	A∷walk()	5	12	public: virtual void speak() { cout << "야용!!!\n"; }
Cat	speak()	13	$\begin{array}{c} -10 \\ 14 \\ 15 \end{array}$	
	A∷walk()	5	16	16 public: 17 → virtual void speak() { cout << "꿀꿀!!!\n"; } 18 }; 19 class Duck : public Animal { 20 public: 21 → virtual void speak() { cout << "꽥꽥!!!\n"; } 22 → virtual void walk() { cout << "두 발로 걷습니다.\n"; } 23 }; 24 class human : public Animal {
Pig	speak()	17 -		
	A∷walk()	5	1	
Duck	speak()	21 -	22	
	walk()	22 -		
human	speak()	26 —		public: → virtual void speak() { cout << "안녕하세요!!!\n"; }
	walk()	27 —	28	→ virtual void walk() { cout << "두 발로 걷습니다.\n"; } };
			29 30	

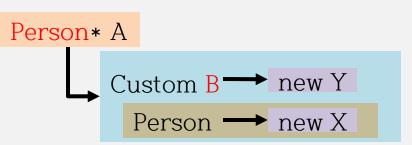
Day14 - 1. 상속(Inheritance) & 다형성(Polymorphism)

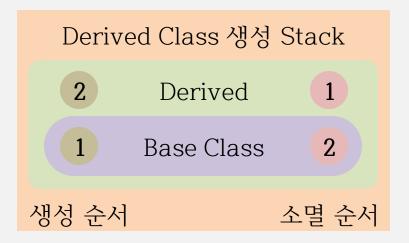
상속에서 소멸자 문제점 < Up-casting : 상위 클래스 * 변수명= new 하위 클래스; > Up-casting에서 delete 하면, 하위클래스의 소멸자는 호출되지 않는다.

- 최상위 클래스의 객체 포인터 활용
- 베이스와 파생 클래스의 생성자 & 소멸자 호출 순서
- 동적 할당 & 해제

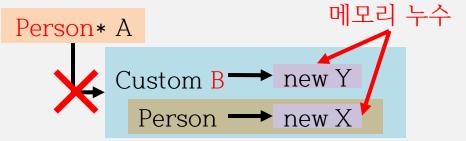


Person * A = new Custom();





delete A;



```
#include <iostream>
                                                         ~Custom() {
using namespace std;
                                                          cout << "\nCustom 소멸";
class Person {
                                                          delete[]phone, nationality;
private:
 char* name;
public:
                                                         virtual void show() {
 Person(const char* pN) {
                                                          Person::show();
                                                          cout << "\n번호: " << phone << "\n주소: " << nationality
  name = new char[strlen(pN) + 1];
  strcpy_s(name, strlen(pN) + 1, pN);
                                                        << endl;
  cout << "\nPerson 생성";
 ~Person() {
                                                        int main() {
  cout << "\nPerson 소멸\n";
                                                         Person* c1 = new Custom("michelin", "010-1234-5678",
                                                        "South Korea");
  delete[]name;
                                                         c1 \rightarrow show();
 virtual void show() {
                                                         delete c1;
  cout << "\n\n이름: " << name << endl;
                                                         return 0;
class Custom: public Person {
                                                                                             03
                                                                         00
                                                                                       02
                                                                                                    04
private:
 char* phone;
                                                              A* ap
                                                                                       B
 char* nationality;
public:
                                                              B* bp
                                                                                       B
 Custom(const char* pN, const char* pP, const char*
pNa): Person(pN) {
                                                              C* cp
                                                                                       B
  phone = new char[strlen(pP) + 1];
  strcpy_s(phone, strlen(pP) + 1, pP);
  nationality = new char[strlen(pNa) + 1];
                                            하위클래스 (Custom)의 소멸자는 호출되지 않는다.
  strcpy_s(nationality, strlen(pNa) + 1, pNa);
  cout << "\nCustom 생성";
```

Day14 - 1. 상속(Inheritance) & 다형성(Polymorphism)

Virtual 소멸자

계층구조상 최하위 클래스 소멸자부터 최상위 클래스 소멸자까지 모두 호출

형식

virtual + 소멸자 형식 [~ 클래스이름 () { ···; }]

특징

- 최상위 소멸자만 virtual 선언시 하위 클래스 모두 가상 소멸자 적용

예제 코드 소멸자 호출 순서		
2 Derived 메모리 누수		
1	Base Cla	ss 1
생성 순서	1	소멸 순서

Virtual 소멸자 호출 순서		
2	Derived	1
1	Base Class	2
생성 순/	4	소멸 순서

```
#include <iostream>
using namespace std;
class Person {
private:
 char* name;
public:
 Person(const char* pN) {
  name = new char[strlen(pN) + 1];
  strcpy_s(name, strlen(pN) + 1, pN);
  cout << "\nPerson 생성";
 virtual ~Person() {
  cout << "\nPerson 소멸\n";
  delete[]name;
 virtual void show() {
  cout << "\n\n이름 : " << name << endl;
class Custom: public Person {
private:
 char* phone;
 char* nationality;
public:
 Custom(const char* pN, const char* pP, const char*
pNa): Person(pN) {
  phone = new char[strlen(pP) + 1];
  strcpy_s(phone, strlen(pP) + 1, pP);
  nationality = new char[strlen(pNa) + 1];
  strcpy_s(nationality, strlen(pNa) + 1, pNa);
  cout << "\nCustom 생성";
```

```
}
        ~Custom() {
            cout << "\nCustom 소멸";
            delete[] phone, nationality;
        }
        virtual void show() {
            Person::show();
            cout << "\n번호:" << phone << "\n주소:" << nationality
        << endl;
        }
    };
    int main() {
        Person* c1 = new Custom("michelin", "010-1234-5678",
        "South Korea");
        c1->show();
        delete c1;
        return 0;
    }
```

Day14 - 1. 상속(Inheritance) & 다형성(Polymorphism)

순수가상함수

- 선언부만 있고 정의부가 없는 가상함수
- 파생 클래스에서 재정의 목적으로 정의되는 함수
- 함수 선언 시 0을 지정 virtual void Speak() = 0

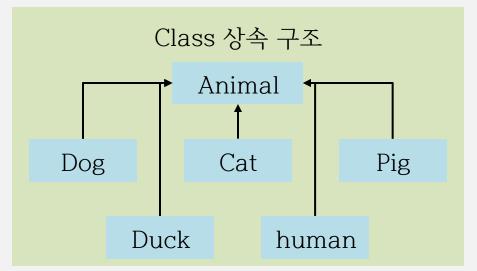
추상 클래스

- 순수가상함수를 포함하고 있는 클래스
- 객체 생성을 할 수 없다.

```
추상 클래스 Animal을 사용해 동물 클래스, 프로그램 작성 [ Up-Casting을 활용 ]
class Animal {
public:
  void speak()
  void walk()
class Dog: public Animal {
public:
  void speak() //override
};
class Cat: public Animal {
public:
  void speak() //override
class Pig: public Animal {
public:
  void speak() //override
};
class Duck: public Animal {
public:
  void speak() //override
  void walk() //override
```

};

```
class human : public Animal {
public:
  void speak() //override
  void walk() //override
};
 선택한 동물의 소리와 걸음 함수 출력하기
 0. exit
    dog
 2. cat
 3. pig
 4. duck
    human
```



```
#include <iostream>
                                            void Walk() { cout << "두발로 걷습니다.\n"; }
                                                                                         ani = new human;
using namespace std;
                                           ~Duck() { cout << "[Duck 소멸자]\n"; }
                                                                                         break;
class Animal { //추상클래스 : 순수 가상함수를 Duck() { cout << "[Duck 생성자]\n"; }
                                                                                        default:
한 개 이상 포함한 클래스
                                                                                         exit(0);
public:
                                           class human: public Animal {
virtual void Speak() = 0; //순수 가상함수 : 파 public:
                                                                                        ani->Speak();
생 클래스에서 오버라이딩 할 수 있도록 함수 원형 void Speak() { cout << "안녕하세요.\n"; }
                                                                                        ani->Walk();
                                            void Walk() { cout << "두발로 걷습니다.\n"; }
                                                                                        delete ani;
만 제공
virtual void Walk() { cout << "네발로 걷습니
                                            ~human() { cout << "[human 소멸자]\n"; }
                                                                                        cout << "\n\n\t\t";
다.\n"; }
                                            human() { cout << "[human 생성자]\n"; }
                                                                                        std::system("pause");
virtual ~Animal() { cout << "[베이스 소멸
자]\n"; }
                                          int main() {
                                                                                      return 0;
 Animal() { cout << "베이스 생성자]\n"; }
                                            Animal* ani = NULL;
                                            //Animal obj; //추상 클래스 객체 선언 불가
class Dog: public Animal {
                                            while (true) {
                                             system("cls");
public:
void Speak() { cout << "멍멍\n"; }
                                             cout << "1. dog 2. cat 3. pig 4. duck
 ~Dog() { cout << "[Dog 소멸자]\n"; }
                                           5. human (1~5 제외: exit) \n";
Dog() { cout << "[Dog 생성자]\n"; }
                                             cout << "choice: ";
                                             int choice;
class Cat: public Animal {
                                             cin >> choice;
                                             switch (choice) {
public:
void Speak() { cout << "야옹\n"; }
                                             case 1:
~Cat() { cout << "[Cat 소멸자]\n"; }
                                              ani = new Dog;
 Cat() { cout << "[Cat 생성자]\n"; }
                                              break;
                                             case 2:
class Pig: public Animal {
                                              ani = new Cat;
public:
                                              break;
void Speak() { cout << "꿀꿀\n"; }
                                             case 3:
~Pig() { cout << "[Pig 소멸자]\n"; }
                                              ani = new Pig;
Pig() { cout << "[Pig 생성자]\n"; }
                                              break;
                                             case 4:
class Duck: public Animal {
                                              ani = new Duck;
public:
                                              break;
void Speak() { cout << "꽥꽥\n"; }
                                             case 5:
```

List Class: 이중 연결 리스트 doubly linked list

- Constructor (생성자) list< [Data Type] > 객체명;

Constructor(생성자)	의미
list()	리스트 기본 생성자 노드 없음
template <class inputiterator=""> list(Iterator First, Iterator Last)</class>	begin ≤ Iterator < end 노드 생성 Iterater : 반복자, 포인터
list(list& Right)	복사 생성자 (깊은 복사)
list(list&& Right)	이동 생성자 (얕은 복사)
list(size Count, Type& Val)	Count 개, val 값 노드 생성
list(size Count, Type& Val, Allocator& Al)	Count 개, val 값 노드 생성
list(initializer_list <type> IList, const Allocator& Al)</type>	initializer_list 생성자 ({ })
template <class inputiterator=""> list(Iterator First, Iterator Last, Allocator& Al)</class>	begin ≤ Iterator < end 노드 생성 Iterater : 반복자, 포인터
list(_Alloc& _Al)	할당자 복사 생성자
list(size Count)	val 0으로 Count 개 노드 생성

```
#include <iostream>
#include <list>
using namespace std;
int main()
 list <int> list01; // list()
 list <int> list02(7); // list(size Count)
 list <int> list03(6, 3);//list(size Count, Type& Val)
 list <int> list04(3, 7, list03.get_allocator()); //list(size Count, Type & Val, Allocator & Al)
 list <int> list05(list03.get_allocator()); //list(_Alloc & _Al)
 list <int> list06(list04); // list(list& Right)
 list <int>∷iterator list2 Iter = list02.begin();
 list2 Iter++;
 list2_Iter++;
 list <int> list07(list02.begin(), list2_Iter); //list(Iterator First, Iterator Last)
 int ary[] = \{ 9, 5, 1, 7, 5, 3, 4, 5, 6, 8, 5, 2, 0 \};
 list <int> list08(arv + 2, arv + 6); //list(Iterator First, Iterator Last)
 list <int>::iterator list3_Iter = list03.begin();
 list3 Iter++;
 list3 Iter++;
 list3 Iter++;
 list <int> list09(list03.begin(), list3_Iter, list03.get_allocator()); // list(Iterator First, Iterator Last, Allocator& Al)
 list <int> list10(ary + 1, ary + 8, list03.get_allocator()); // list(Iterator First, Iterator Last, Allocator& Al)
 list <int>∷iterator Iter;
 cout << "list01 " << list01.size() << " ="; for (list <int>::iterator _Iter = list01.begin(); _Iter!= list01.end(); _Iter++) cout << " " << *_Iter; cout << endl;
 cout << "list03 " << list03.size() << " ="; for (list <int>::iterator _Iter = list03.begin(); _Iter!= list03.end(); _Iter++) cout << " " << *_Iter; cout << endl;
 cout << "list04 " << list04.size() << " ="; for (list <int>::iterator _Iter = list04.begin(); _Iter!= list04.end(); _Iter++) cout << " " << *_Iter; cout << endl;
 cout << "list05 " << list05.size() << " ="; for (list <int>::iterator _Iter = list05.begin(); _Iter!= list05.end(); _Iter++) cout << " " << *_Iter; cout << endl;
 cout << "list06 " << list06.size() << " ="; for (list <int>::iterator _Iter = list06.begin(); _Iter!= list06.end(); _Iter++) cout << " " << *_Iter; cout << endl;
 cout << "list07 " << list07.size() << " ="; for (list <int>::iterator    Iter = list07.begin();    Iter!= list07.end();    Iter++) cout << " " << * Iter; cout << endl;
 cout << "list08." << list08.size() << " ="; for (list <int>::iterator _Iter = list08.begin(); _Iter!= list08.end(); _Iter++) cout << " " << *_Iter; cout << endl;
 cout << "list09 " << list09.size() << " ="; for (list <int>::iterator _Iter = list09.begin(); _Iter!= list09.end(); _Iter++) cout << " " << *_Iter; cout << endl;
 cout << "list10 " << list10.size() << " ="; for (list <int>::iterator _Iter = list10.begin(); _Iter!= list10.end(); _Iter++) cout << " " << *_Iter; cout << endl;
 list<int> list11({ 1, 2, 3, 4 }); // list(initializer_list<Type> IList, const Allocator& Al)
 cout << "list11 " << list11.size() << " ="; for (list <int>::iterator _Iter = list11.begin(); _Iter!= list11.end(); _Iter++) cout << " " << *_Iter; cout << endl;
```

Method (멤버 함수) 객체.Method();	의미	
(const) reference front() (const);	첫 번째 요소 참조 반환	
(const) reference back() (const);	마지막 요소 참조 반환	
iterator <pre>begin();</pre>	첫 번째 요소의 주소	
iterator end();	마지막 요소 다음의 주소	
clear();	모든 요소 삭제	
void remove(const Type& val);	val 값과 일치하는 요소 모두 삭제	
<pre>iterator erase(iterator Where); iterator erase(iterator first, iterator last);</pre>	지정된 위치의 요소 제거 first~last 범위 요소 제거	
<pre>void pop_front();</pre>	목록 시작에 있는 요소를 삭제	
<pre>void pop_back();</pre>	목록의 끝에 있는 요소를 삭제	
<pre>void push_front(Type&& val);</pre>	목록 시작에 요소 추가	
void <pre>push_back(Type&& val);</pre>	목록 끝에 요소 추가	
<pre>insert(iterator Where, Type&& Val);</pre>	목록의 지정된 위치에 삽입	
emplace(iterator Where, Type&& val);	목록의 지정된 위치에 삽입	
assign(InputIterator First, InputIterator Last);	목록 대체 삽입(기존 요소 삭제)	
https://learn.microsoft.com/ko-kr/cpp/standard-library/list-class?view=msvc-170		

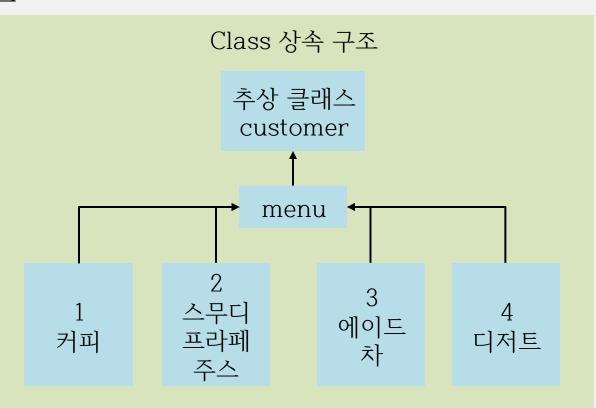
Method (멤버 함수) 객체. Method();	의미
bool empty() const;	목록이 비어 있는지 여부
<pre>void swap(list<type, allocator="">& right);</type,></pre>	두 목록의 요소를 교환
size_type size() const;	요소 수를 반환
<pre>get_allocator() const;</pre>	list의 할당자 복사본 반환
<pre>void reverse();</pre>	list 요소를 역순으로 설정
<pre>void sort();</pre>	목록의 요소를 정렬 (오름차순)
void merge(list <type, allocator="">& right);</type,>	리스트 합병 정렬 (오름차순)

추상 클래스, 상속을 활용해 커피메뉴 프로그램 작성 [Up-Casting을 활용]

선택한 모든 메뉴의 주문 내역(list), 총 합계 출력 초기 고객정보 입력, 이름, 전화번호, (주문 번호 부여) [메가 커피 Kiosk]

- 1. 커피
- 2. 스무디 & 프라페 & 주스
- 3. 에이드 & 차
- 4. 디저트
- 5. 계산할 총 금액
- 6. 종료

메뉴를 입력하세요.:



```
Unicornfrappolytring _name = "Maga", string _phone = "010-0000-0000", string _topping = "Nothing", string _product ); class EarlGreyFound: public menu (
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      void login() {
  cout << "[ 메가 커피 Kiosk ] \n";
  cout << "(이름과 언락치를 남겨주세요.) \n";
  cout << "이 등: ";
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      - ons.cen trappe , int_proce = 4800)

'menu(_name,_phone,_product,_price), topping(_topping)

const string& gettopping()const { return topping(} }
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      protected
string packaging:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              cout << 이 등:
getline(cin, name);
cout << "연락처: ";
getline(cin, phone);
callnum++;
    class customer {//추상 클래스
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        void order() {
cout<< "상품명: "<< product << end):
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           EarlGreyPound(string _name = "Mega", string _phone = "010-0000-0000", string _packaging = "Packaging", string
         urus:
virtual ~customeri) { cout << "회상의 클래스 소면자\n"; }
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                cout<< "가 격:" << price << endl:
cout<< "추가토핑:" << topping << endl< endl:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        _product = "Earl Grey Pound", int_price = 2500)

:menu(_name, _phone, _product, _price), packaging(_packaging) {

const string& getpackaging()const { return packaging; }
         virtual void customerinfo() = 0;
virtual const intgetprice() const = 0;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           void order() {
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           Cutist stillings appearance (see a second control of courts '항 품명: "< product <=ndi: courts '항 금명: "< product <=ndi: courts '항 구명' <=ndi: courts <="하기 구기" <=pre>courts <=ndi: courts <
         virtual void login() = 0:
virtual const string getname() const = 0:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                cout << "상품명: " << product << endl:
cout << "가 격:" << price << endl:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      class StrawberryCheeseholic: public menu {
         virtual const string getphone() const = 0:
void MainMenu(int& main) {
system("cls"):
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               public ....

Straterry-Cheesehdictering name * Mapr, string phone * 100-0000-0007, string lepping * hall of smederry-Cheesehdictering phone, product, preds toping lepping * fall of mental name, phone, product, preds toping lepping | 0 and order | 0 person phone, product, preds toping lepping | 0 and order | 0 person phone, product, preds toping | 0 person phone, product, preds toping | 0 person phone, product, preds toping | 0 person phone | 0 
            system(cts);
cout << "[ 메가 커피 Kiosk] \n";
cout << "1. 커피 \n";
cout << "2. 스무디 & 프라페 & 주스 \n";
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              cout < "인 를 :" < name << endi:
cout < "인 락 처 : " << phone << endi:
cout << "호출번호 :" << callnum << endi << endi:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           class Potatobread : public menu {
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        string packaging;
public:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           cout < "상 등 명 : " < product << end:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      public "Heiderwicking name = Mega", string phone = "010-0000-0000", string pockaging = Tackaging", string product = Tackaging = Tackaging = string product = Tackaging = Tackaging = string product = Tackaging = 
              cout << '5. 계산할 총 금액 \n':
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  cout << "가 격:" << price << endl:
cout << "추가토핑:" << topping << endl << endl:
                   cout << "메뉴를 입력 하세요.:"
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    string variety: //커피 품종 Arabica, Robusta, Liberica public:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               void croem) {
cout << "창품명:" << product << endl:
cout << "가 경:" << price << endl:
cout << "시사&포장:" << packaging << endl << endl:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         American o(string _name = "Mega", string _phone = "010-0000-0000", string _variety = "Arabica", string _product =
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    class CherryCoke: public menu (
                 while (cin.set() != "\n");
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              void order() {
cout< "상품명:"<< product < endi:
cout<< "가 격:"<< price << endi:
cout<< "가비품종:"<< variety << endi << endi:
            void SubMenul(int&subl){
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        CherryCoke(string _name = "Mega", string _phone = "010-0000-0000", string _size = "Grande", string _product =
                   system("cls");
cout << "[메카커피 Kiosk] \n";
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             customer* KIOSK = new menu
list< customer* > list1:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             void order() {
cout < "상 품 명 : " << product << nd:
cout < "가 격 :" << price << end:
cout << "가 격 :" << price << end:
cout << "음료용량 :" << size << end! << end:
                 cout << "(3), 카페모카 \n";
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         class Cappuccino: public menu (
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           lists customer >: iterator itr
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           string variety: //커리 품종 Arabica, Robusta, Liberica
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           tublic:
Cappuccino(string _name = "Mega", string _phone = "010-0000-0000", string _variety = "Robusta", string _product = }:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  Capputcincipting_name= Maga; string_phone= 1010000-000

AmericanG, int_price= 27000

menu(name_phone_product_price) variety(variety) {
vaid order() {
void o
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        class Bluelemonade : public menu {
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            wing gince

"Bushemonskeiting _name "Mogs", string _thone "1010-0000-0000", string _size = Venii", string _product = Tibue
|lemonseiv_int_price = 2500|
|monseiv_name_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_price_pr
                 while (cin.set() != "\n");
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    KIOSK->SubMenul(menul)
            void SubMenu2(int&sub1){
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    KIOSK-SystiManu | Imenuizz
switch | nemur2|
case | 1-Newcrder | new American (RIOSK-systemane), RIOSK-systemane)| list! | push_backNewcrder| treak
case | 3-Newcrder | new Caspuccin (RIOSK-systemane), RIOSK-systemane)| list! | push_backNewcrder| treak
case | 3-Newcrder | new Caladken hall (DISK-systemane), RIOSK-systemane)| list! | push_backNewcrder| treak
case | 4-Newcrder | new Caladken hall (DISK-systemane)| RIOSK-systemane)| list! | push_backNewcrder| treak
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           dass CafeMocha : public menu (
              cout << "(2). 플레인종크러쉬 \n":
cout << "(3). 유니콘프라페 \n":
cout << "(4). 스트로베리치즈홍릭 \n":
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           string variety: //커리 품종 Arabica, Robusta, Liberica
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      CafeMocha(string _name = "Mega", string _phone = "010-0000-0000", string _variety = "Liberica", string _product = "CafeMocha", int _price = 3700)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        class EarlGrev: public menu (
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         switch (menuz) (
case 1: Neworder = new Realch ocofrappe(KIOSK->getname(), KIOSK->getph one(); list1.pu sh_back(Neworder);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    Earding-vighting, name = "Maga", string_phone = "010-0000-0000", string_size = "Grande", string_product = "Earl created and the string product = "Earl created and the string 
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      Grey'. int_price = 2500)
menuf_name_phone_product__price), size(_size) {|
const string&getisze()const {return size:}
cout < "성통형:" < product <<endi:
cout < "성통형:" < product <<endi:
cout < "성통형:" < price < endi:
cout < "성통형:" < endi <<endi:
                 while (cin.set() != "\n");
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   creas.

cas e 3: N eworder = new Unicornfra ppe(KIOSK > 3ge hname(), KIOSK > 3ge hphon e(i): list it, push_back(New order): brea & case & Neworder = new StrawberryCheesehclic(KIOSK > 3gehname(), KIOSK > 3gehnone(i): list(push_back(Meworder): brea & list(push_back(
            unid SubMonu Wints mubl) /
              system("cls"):
cout << "[ 메가 커피 Kiosk] \n":
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           rosecsed.
string variety: //커리 중중 Arabica, Robusta, Liberica
              cout << "(1). 세리꼭 \n".
cout << "(2). 블루레몬에이드 \n".
cout << "(3). 얼그레이 \n".
cout << "(4). 허니자몽블랙티 \n".
cout << "(10). 이전 메뉴로....\n".
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           Coldbrew(string _name = "Mega", string _phone = "010-0000-0000", string _variety = "Robusta", string _product =
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         "Coldbrew", int_price = 3300)
:menu(_name, _phone, _product, _price), variety(_variety) {}
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         switch (menu2) {
    case 1: Neworder = new CherryCoke(KIOSK->getname(), KIOSK->getphone(); list1.push_back(Neworder); break:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              const string& getvariety()const {return variety; }
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 case 3. New order = new Build manadel(DSS 2550 hamman), KDSS 2550 period en eff juit 101, Debuglederen). The Res
case 3. New order = new Build manadel(DSS 2550 hamman), KDSS 2550 period en eff juit 101, Debuglederen). The Res Le
case 4. New order = new Build res(NDSS 2550 hamman), KDSS 2550 period en eff juit 101, Debuglederen et al. (DSS 2550 period). (Build public publ
                   cout << "메뉴를 입력 하세요.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        string size:
              while (lcin >> subl)) {
cin.clear();
while (cin.get() != '\n');
cout <<" 다시 일찍 하세요.:";
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           void order() {
cout<<"항 품명:"</pre> product << endi:
cout<<"항 품명:"</pre> price << endi:
cout<< "키리종종:"</p>
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      Graphuilfleneyfdeddaidstring _name="Mogo", string_three="010-0000-0000", string_size="Ven6", string_
_routate="Graphuilfleneyfdeddaidstring.neme="5000"
_menu(_name__phone__product__prine), size(_size) {\( \) }
const string\( \) genferm(constring) for the principle of the princi
                 while (cin.get() != "\n"):
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        waid crider() {
cout < "상등 명: "< product < endi:
cout < "가 격:" < price < endi:
cout < "음료용당:" < size < endi < endi:
            void SubMenu4(int& subl){
system("cls");
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         switch (m on 23) {
case 1: Neworder = new Plaincroiffle(KIOSK->getname(), KIOSK->getphone()): list1.push_back(Neworder): break:
                 system (cis ).
cout << "[ 메가 커피 Kiosk l \n":
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         string topping:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       case 2:N eworder = new Tiramisu Cale(KIOSK ->geth ame(), KIOSK ->getph one()): list t.p.ush_bock(Neworder): break:
case 3:Neworder = new EarlGreyPound(KIOSK ->gethame(), KIOSK ->getph one()): list1.p.ush_bock(Neworder):
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      prone:

Reach-coxfrappt/string _name="Mega", string_phone="010-0000-0000", string_topping="More Whippingc
string_roduct="Real-choodule frappe", int_price==3000
menu[name_phone_product_triviel topping_topping]
const string& gettopping)const [return topping]
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 ase 4: Neworder = new Potatobread(KIOSK->eetname(). KIOSK->eetohone(); list1.ou.sh back(Neworder); break
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           string packaging:
            cout << "[10]. 이전 메뉴로...'n":
cout << "메뉴를 일력 하세요.:"
while (l/cin >> sub1)) {
cin.cloar():
while (cin.got) != "\n"):
cout <<" 다시 일력 하세요.:";
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      public

Plain craffie(string _name = Maga*, string_phone = "010-0000-0000", string_puckaging = "Packaging*, string_product = "Main craffie", int_price = 2500

menu(name_phone_product_price) packaging_packaging] 
const string$ getpackaging@const (return packaging) }
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                cout < "상품명:" << product << end:
cout << "가 격:" << price << end:
cout << "추가토핑:" << topping << end! << end:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            KIOSK->cu stomerin fo():
for (itr = list1.begin(): itr != list1.en d(): itr++)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                cout << "가 걸:" << price << endl:
cout << "식사&포장:" << packaging << endl << endl:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         ,
cout <<" 총 계산하실 금액은 :"<<total << "원 입니다. \n \n":
total = 0:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    protected:
string topping:
         virtual void order() = 0;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      Pain progravahleting, name = Mega", string_thone = "010-0000-0000", string_tepping = "Nothing", string_product|;
="Pain progravah", int_price = 25000)
= menu(_name_thone_product_price), tepping|tepping) ()
= conststrings|tepping|const| fruitur tepping| = s
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         system ("pau se"):
break:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         ase 10:
cout <<" [프로그램 종료 합니다.]\n\n\n":
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              void crose() 1
cout< "창품명: "</pre> product <<pre>endi:
cout '가 격:" cont '* ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **</
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      pinter mention Calestering , name: "Mages", string , phone: "D10-0000-0000", string , packaging a "Packaging", string , proceedings Calestering , proceedings Calestering , proceedings Calestering , proceedings () mention (, name, , phone, , product, , price), pockaging (, packaging () a count strings () geophysical pin() proceedings () a count strings () geophysical pin() proceedings () and count strings () a proceedings () and count strings () and country () and cou
public:
    menu(string_name = "Mega", string_phone = "010-0000-0000", string_product = "water", int_price = 0)
    name(_name), phone(_phone), product(_product), price(_price) {
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        const stringly geopackaging (const 'Feturu purvagury
void order() {
cout<''가 등 명:"<< price < endi:
cout<''가 결:"</price < endi:
cout<''식사を포장:"<< packaging < endi << endi:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  protected:
string topping:
public:
```

Day15 - Contents

- 1. 연산자 오버로딩 (Operator Overloading)
 - friend 함수
 - 연산자 오버로딩 개념
 - 단항 연산자 오버로딩
 - cin, cout, endl
- 2. 예제 코드

Day15 - 1. 연산자 오버로딩 (Operator Overloading)

friend

외부에서 클래스 멤버로 접근하지 못하도록 private, protected 제한 friend 선언으로 다른 클래스 (or 함수) 에 private, protected 접근 허용

- friend 키워드로 명시하지 않은 경우 접근 불가 (개별적으로 접근하고 싶은 곳 모두 키워드를 써야 한다.)

형식

friend 키워드를 클래스 (or 함수) 앞에 사용
friend class (클래스 이름); // friend class Date;
friend +(함수 원형);// friend void OutToday(Date &d, Time &t);

friend 멤버함수

- 클래스의 특정 멤버함수만 friend로 선언하여 접근 허용 범위를 줄인다.
- friend 멤버함수는 순서가 중요하다.
 - (1) friend 로 선언될 멤버함수를 포함한 클래스
 - (2) friend 멤버함수를 선언할 클래스
 - (3) friend 멤버함수 외부 정의

```
/*
  friend 클래스
  - friend로 선언된 클래스는 friend를 선언한 클래스의 private멤버까지도 접근 가능하다.
#include <iostream>
using namespace std;
class Time {
private:
  int hour, min, sec;
public:
  Time(int h, int m, int s): hour(h), min(m), sec(s) {}
  friend class Date; //friend 클래스(Date클래스는 Time클래스의 private멤버에 접근 가능)
class Date {
private:
  int year, month, day;
public:
  Date(int y, int m, int d): year(y), month(m), day(d) {}
  void OutToday(Time& t);
};
void Date::OutToday(Time&t) {
  cout << "오늘 날짜는" << year << "/" << month << "/" << day << "\n";
  cout << "지금시간은" << t.hour << ":" << t.min << ":" << t.sec << endl;
int main() {
  Date D(2023,2,14);
  Time T(10, 40, 15);
  D.OutToday(T);
  return 0;
```

```
/* friend 멤버함수
 - 클래스의 특정 멤버함수만 friend로 선언하여 접근 허용 범위를 줄인다. (class -> method)
 - friend 멤버함수는 순서가 중요하다.
  (1) friend 선언될 멤버함수를 포함한 클래스
  (2) friend 멤버함수를 선언할 클래스
  (3) friend 멤버함수외부정의*/
#include <iostream>
using std∷cout;
class Time; // ## OutToday함수에서 Time을 클래스를 사용하기 때문에 선언
class Date { //(1)friend 선언될 멤버함수를 포함한 클래스
private:
  int year, month, day;
public:
  Date(int y, int m, int d): year(y), month(m), day(d){}
  void OutToday(Time&t); // 함수의 원형 ##Time을 클래스로 인식하기 위해 선언
};
class Time { //(2) friend 멤버함수를 선언할 클래스
private:
  int hour, min, sec;
  friend void Date::OutToday(Time& t);//Date클래스의 OutToday함수만 Time클래스의 private멤버에 접근할수 있다.
public:
  Time(int h, int m, int s): hour(h), min(m), sec(s) {}
};
void Date::OutToday(Time&t) { //(3) friend 멤버함수 외부 정의
  //t.hour = 15; t.min = 24; t.sec = 45;
  cout << "오늘은 " << year << "년 " << month << "월 " << day << "일이며\n";
  cout << "지금 시간은 " << t.hour << ":" << t.min << ":" << t.sec << "입니다.\n";
int main() {
  Date D(2023, 2, 14);
  Time T(10, 35, 27);
  D.OutTodav(T);
  return 0;
```

```
/* friend 전역함수
전역함수는 friend를 선언한 클래스의 private멤버까지도 접근 가능
객체지향에 위배되기 때문에 가급적 사용하지 않는 것이 원칙이다.
특별한 경우에 전역함수를 자신의 멤버함수처럼 필요할 때 선언한다.
*/
#include <iostream>
using namespace std;
class Date; //Time 클래스에서 OutToday함수의 매개변수로 Date클래스를 사용하기 때문에 Date클래스 선언
class Time {
private:
 int hour, min, sec;
friend void outToday(Date&d, Time&t); // friend 멤버 함수 원형 선언, (Data를 클래스사용)
public:
 Time(int hour = 0, int min = 0, int sec = 0): hour(hour), min(min), sec(sec) {}
};
class Date {
private:
 int year, month, day;
friend void outToday(Date&d, Time&t); // friend 멤버 함수 원형 선언
public:
 Date(int year, int month, int day): year(year), month(month), day(day) {}
void outToday(Date&d, Time&t)
{ //전역함수, Time class private : int hour, min, sec; & Date class private: int year, month, day; 접근 가능
 cout << "오늘은 " << d.vear << "/" << d.month << "/" << d.day << ". " << t.hour << ":" << t.min << ":" << t.sec << endl;
int main() {
 Date d(2023,02,14); // Date 객체 생성
 Time t(10,30,28); // Time 객체 생성
 outToday(d, t); // 전역 함수 호출
return 0;
```

Day15 - 1. 연산자 오버로딩 (Operator Overloading)

```
연산자 오버로딩 (객체 멤버함수의 연산자 적용)
객체를 다양한 연산자로 직관적인 기능 구현 [자료형(int, double, char…)]
형식: operator + "연산자[+,-,*,/,+=,++,--,…]" = 함수의 이름 을 정의
호출 형식
                                  class Point {
- 함수 이름 호출
                                  private:
   Point p1(1,1), p2(2,2);
                                    int x, y;
   p1.operator+(p2);
                                  public:
                                    Point(int x = 0, int y = 0) : x(x), y(y) {}
- 연산자 이용 호출
                                    Point operator+ (const Point& r) {
   Point p1(1,1), p2(2,2);
                                      Point temp;
                                      temp.x = this -> x + r.x;
   Point p3 = p1 + p2;
                                      temp.y = this -> y + r.y;
                                      return temp;
p1. operator+ (p2)
                                    Point operator - (const Point& r) {
                                      Point temp;
멤버함수 오버로딩
                                      temp.x = this -> x - r.x;
p1.operator+(p2);
                                      temp.y = this->y - r.y;
전역함수 오버로딩
                                      return temp;
operator+(p1, p2); 교환법칙
멤버함수가 전역함수보다 우선시 호출
```

Day15 - 1. 연산자 오버로딩 (Operator Overloading)

```
단항 연산자 오버로딩
- 전치(전위) 연산자
해당 줄에서 먼저 연산처리 후
변경된 값으로 다음을 진행.
Point p1(1,1);
++p1; // p1.operator++();
++++++++++p1;
//(++(++(++(++(++p1))))); 같다
[[[p1.operator++()].operator++()].operator++())].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operator++()].operat
```

후치(후위) 연산자
(int: 후치 연산자 구분 키워드 > 해당 줄 연산 모두 종료 후 증감. 임시 객체의 값으로 해당 줄 연산 호출한 객체의 값은 증감 처리. Point p1(1,1), p2;
p1++; // p1.operator++();
p2 = p1++;
//p1(2,2), p2(1,1)

```
후치 연산자 오버로딩
Point operator++(int) {
    Point temp;
    temp.x = x;
    temp.y = y;
    x += 1;
    y += 1;
    return temp;
}
```

```
//전역함수에 의한 오버로딩
#include <iostream>
using namespace std;
class Point {
private:
 int x, y;
public:
 Point(int x = 0, int y = 0) : x(x), y(y) {}
 Point operator+ (const Point&r) { // Point + Point
  Point temp;
  temp.x = this -> x + r.x;
  temp.y = this -> y + r.y;
  return temp;
 Point operator - (const Point&r) { // Point - Point
  Point temp;
  temp.x = this -> x - r.x;
  temp.y = this -> y - r.y;
  return temp;
 Point& operator++() {
  x += 1;
  v += 1;
  return *this;
 }//반환값이 호출 객체자기 자신
 Point operator++(int) {
  Point temp;
  temp.x = x;
  temp.v = v;
  x += 1;
  y += 1;
  return temp;
```

```
inline int GetX() { return x; }
 inline int GetY() { return y; }
 friend Point operator+(int value, const Point&r); //전역함수를
자신의 멤버함수처럼 쓸 수 있다.
friend Point operator-(int value, const Point&r); //전역함수를
자신의 멤버함수처럼 쓸 수 있다.
};
Point operator+(int value, const Point&r) { return Point(r.x+
value, r.y + value); } // Value + Point
Point operator-(int value, const Point&r) { return Point(r.x -
value, r.y - value); } // Value + Point
int main() {
 int x. v;
 cout << "첫 번째 x, y좌표 입력:";
 cin >> x >> y;
 Point p1(x, y);
 Point p2 = 10 + p1; //10.operator+(p1) //불가능 -> operator+(10,
p1) //전역함수 오버로딩으로 해결
 cout << "p1(" << p1.GetX() << ", " << p1.GetY() << ")\n";
 cout << "p2(" << p2.GetX() << ", " << p2.GetY() << ")\n";
 +++++p1;
 cout << p1.GetX() << ", " << p1.GetY() << "\n";
 p1++;
 cout << p1.GetX() << ". " << p1.GetY() << "\n";
 return 0:
```

Day15 - 1. 연산자 오버로딩 (Operator Overloading)

https://learn.microsoft.com/ko-kr/cpp/cpp/operator-overloading?view=msvc-170

구분	연산자	기능
오버로딩 불가능한 연산자	•	멤버 접근
	.*	멤버 포인터
	::	범위 지정자
	?:	3항 연산자
	sizeof	Byte 크기 계산
	typeid	RTTI 연산자
	static_cast	type casting
	dynamic_cast	type casting
	const_cast	type casting
	retinterpret_cast	type casting
멤버함수로만 오버로딩 가능한 연산자	=	대입 연산자
	()	함수 호출 연산자
	[]	배열 접근 연산자
	->	멤버 접근 포인터 연산자

Day15 - 1. 연산자 오버로딩 (Operator Overloading)

연산자 오버로딩 주의사항

- 함수의 기능을 그 어떤 사람이 보더라도 같은 의미로 통할 수 있도록 작성 직관적이고 일반적으로! 객체에 따라 같은 연산자를 다르게 처리하면......
- 연산자의 우선순위와 결합성은 바뀌지 않는다. 결합성: 연산자 처리 순서 왼쪽 -> 오른쪽 or 왼쪽 <- 오른쪽
- 매개변수의 디폴트 값 설정이 불가능하다
- 연산자의 기본 기능을 오버로딩 할 수는 없다.

```
//stream cout 클래스 디자인
#include <stdio.h>
namespace psk {// namespace std; == namespace psk;
  class o_str {
  public:
     o_str& operator<<(const char* const str) { printf("%s", str); return *this; }
     o_str& operator<< (int i) { printf("%d", i); return *this; }
     o_str& operator<< (double d) { printf("%f", d); return *this; }
     o_str& operator<< (char c) { printf("%c", c); return *this; }
  };
  o_str print; ///객체 생성 ( cout == print )
  const char* const endline = "\n"; // endl == endline
using namespace psk; // namespace std; == namespace psk;
//using psk::endline; // endl == endline
//using psk::print; // cout == print
int main() {
  print << "Hello World" << ", " << 3.14 << ", " << endline;
  print << 100 << ", " << 'Q' << endline;
  return 0;
```

```
/* cin으로 Point 객체 입력 / cout으로 Point 객체 출력
 - cin은 istream 클래스의 객체, cout은 ostream 클래스의 객
체.
 - istream이나 ostream 클래스의 변경이 불가능 (멤버함수에
의한 오버로딩은 불가능)
 전역함수의 의한 오버로딩으로 처리
*/
#include <iostream>
using namespace std;
class Point {
private:
 int x, y;
public:
 Point(int x = 0, int y = 0) : x(x), y(y) {}
 inline void SetX(int x) \{ this -> x = x; \}
 inline void SetY(int y) { this->y = y; }
 inline int GetX() const { return x; }
 inline int GetY() const { return y; }
 friend ostream& operator << (ostream& out, Point& p);
//friend함수
 friend istream& operator>>(istream& in, Point& p);
};
ostream& operator << (ostream& out, Point& p) { out << p.x
<< ", " << p.y; return out; } //전역함수
istream& operator>>(istream& in, Point& p) { in >> p.x >>
p.y; return in; } //전역함수
int main() {
 int x, y;
 cout << "\n x, y좌표 입력 (공백구분): ";
```

```
cin >> x >> y;
Point p1(x, y); //객체 생성
 cout << "p1(" << p1 << ")\n"; // cout으로 point 객체 출력
//cout.opearator<<(p1); //불가능 - 표준 클래스에 멤버 함수
정의 불가능
 operator<<(cout, p1); //전역함수에 의한 오버로딩으로 해결
Point p2; //객체생성
 cout << "\n x, y좌표 입력 (공백구분): ";
 cin >> p2; //전역함수에 의한 오버로딩으로 해결
 cout << "p2(" << p2 << ")\n"; // cout으로 point 객체 출력
 operator<<(cout, p2); //전역함수에 의한 오버로딩으로 해결
return 0;
```

friend, 연산자 오버로딩(Operator Overloading)을 활용한 Time 클래스 작성.

```
#include <iostream>
#include <time.h> // time(): 1초 단위 반환
using namespace std;
class Time
private:
 int hours; // 시
 int minutes; // 분
 int sec; // 초
public:
 Time(int h = 0, int m = 0, int s = 0) : hours(h), minutes(m), sec(s) {}
 void Show() const {}
 Time& operator+(const int s) {} // 객체 + 정수
Time operator+(const Time& t) const {} // 객체 + 객체
Time operator-(const Time& t) const {} // 객체 - 객체
 Time operator*(double multiple) const {} // 객체 * 실수 friend Time operator*(double m, const Time& t){} // 실수 * 객체
 friend ostream& operator<<(ostream& os, const Time& t) {} // cout 출력
```

```
sum.minutes %= 60:
#include <iostream>
                                                                                                                       int main() {
                                                              sum.sec %= 60;
#include <cstdlib>
                                                                                                                        Time WakeUp(6, 30);
                                                                                                                        Time attendance(10,00);
#include <time.h>
                                                              return sum;
using namespace std;
                                                                                                                         Time closing(19, 00);
class Time
                                                             Time operator-(const Time&t) const {
                                                                                                                         Time study(22, 30);
                                                              Time diff;
                                                                                                                        Time sleep (23, 45);
                                                              long long tot1, tot2, tot3;
                                                                                                                         int sec1 = 0, sec2 = 0;
private:
int hours;
                                                              tot1 = sec + 60 * t.minutes + 3600 * t.hours;
                                                                                                                         int hours = 20. min = 40. sec = 45;
                                                              tot2 = sec + 60 * minutes + 3600 * hours;
                                                                                                                         cout << "현재 시간을 입력 하세요." << endl;
int minutes;
                                                              tot3 = ((tot2 - tot1) > 0) ? (tot2 - tot1) : (tot1 - tot2);
                                                                                                                         cout << "시(hours):";
int sec;
public:
                                                              diff sec = 20:
                                                                                                                        cin >> hours:
Time(int h = 0, int m = 0, int s = 0): hours(h), minutes(m),
                                                              diff.sec = (tot3 \% 60);
                                                                                                                        cout << "분(min): ";
sec(s)
                                                              tot3 /= 60;
                                                                                                                         cin >> min;
void AddSec(int s) {
                                                              diff.minutes = tot3 % 60;
                                                                                                                         cout << "초(sec):";
  sec += s;
                                                              tot3 /= 60;
                                                                                                                         cin >> sec;
                                                              diff.hours = tot3;
                                                                                                                         Time realtime(hours, min, sec);
  minutes += sec / 60;
  hours += minutes / 60;
                                                              return diff:
                                                                                                                         Time temp;
  hours %= 24;
                                                                                                                         while (1) {
  minutes %= 60;
                                                             Time operator*(double multiple) const {
                                                                                                                          sec1 = time(NULL);
  sec %= 60;
                                                              Time result;
                                                                                                                          if (sec1! = sec2) {
                                                                                                                           system("cls");
                                                              long long tot3 = multiple * (sec + 60 * minutes + 3600 *
 void AddMin(int m) {
                                                           hours);
                                                                                                                           sec2 = sec1;
  minutes += m;
                                                              result.sec = tot3 % 60;
                                                                                                                           realtime = realtime + 1;
                                                                                                                           cout << "현재 시간: " << realtime << endl << endl;
  hours += minutes / 60;
                                                              tot3 /= 60;
  hours %= 24;
                                                              result.minutes = tot3 % 60;
                                                                                                                           temp = attendance - closing;
                                                                                                                           cout << "기상: " << WakeUp << endl;
cout << "출근: " << attendance << " 근무 시간: " << temp
  minutes %= 60;
                                                              tot3 /= 60:
                                                              result.hours = tot3;
void AddHr(int h) { hours += h; hours %= 24; }
                                                              return result;
void Reset(int h = 0, int m = 0) { hours = h; minutes = m; } }
                                                                                                                           cout << "퇴근: " << closing << " 2.5일 근무: " << 2.5 *
 Time& operator+(const int s) {
                                                             friend Time operator*(double m, const Time&t)
                                                                                                                       temp << endl:
                                                                                                                           cout << "공부 : " << study << endl;
  sec += s;
  minutes += sec / 60;
                                                              return t * m;
                                                                                                                           cout << "꿀잠: " << sleep << endl;
  hours += minutes / 60;
  hours %= 24;
                                                             void Show() const {
                                                              cout << hours << "시 " << minutes << "분 " << sec << "초 return 0;
  minutes %= 60;
                                                           " << endl;
  sec %= 60;
  return *this;
                                                            friend std::ostream& operator<<(std::ostream& os, const
 Time operator+(const Time&t) const {
                                                              cout << t.hours << "시" << t.minutes << "분" << t.sec
  Time sum:
                                                           << "초 ";
  sum.sec = sec + t.sec;
  sum.minutes = minutes + t.minutes + sum.sec / 60;
                                                              return os;
  sum.hours = hours + t.hours + sum.minutes / 60;
  sum.hours %= 24;
                                                           };
```

Day16 - Contents

- 1. 템플릿 (Template)
 - 개념
 - 함수 템플릿
 - 클래스 템플릿
- 2. 예제 코드

데이터의 추상화, 코드의 재활용

객체지향 프로그래밍 (Objective-Oriented Programming) 추상적인 대상을 고유 속성과 기능으로 정의, 다양한 객체를 공통 기준으로 표준화(속성, 기능)하는 방법론 (데이터 중점)

일반화 프로그래밍 (Generic Programming) 다양한 자료형을 모두 포괄적으로 처리하는 방법론 (알고리즘 중점)

[자료형에 얽매이지 않는 방법]

함수 오버로딩 (Function Overloading) 함수의 이름, 매개변수의 자료형, 개수에 따라 함수를 개별적으로 구분

void swap (char a, char b);
void swap (int a, int b);
void swap (double a, double b);

…..?..?..? 필요할 때마다 작성..?.. 귀찮..!

템플릿 (Template) 개념

함수 or 클래스를 자료 타입에 따라 개별적으로 다시 작성하지 않아도, [오버로딩 (overloading)]

자료형에 관계없이 범용적인 함수 or 클래스를 제작하는 틀 (여러 자료 형으로 함수 or 클래스를 만들 수 있는 틀)

매개변수의 자료형을 확인하고, 컴파일 단계에서 매개변수의 자료형에 맞는 함수 or 클래스 제작

- 함수 템플릿(Function Template)
- 클래스 템플릿(Class Template)

템플릿을 많이 쓰면 컴파일 속도가 느려진다. (프로그램 처리 속도 X)

소스코드의 유지보수성이 뛰어나게 완성도를 높이고, STL(Standard Template Library)처럼 검증된 라이브러리를 사용한다면 코드의 완성도와 생산성을 모두 갖출 수 있다.

함수 템플릿(Function Template)

- 함수 템플릿 : 함수를 만드는 틀
- 다양한 자료 타입을 적용할 수 있는 범용적인 함수를 제작 (Generic Programming)
- 자료 타입을 템플릿 기호로 표기, 일반적으로 T를 사용
- 기능(논리)은 명확하게 작성
- 컴파일 단계에서 만들어지기 때문에, 컴파일 시간이 늘어난다.
- 함수의 호출 구문으로 매개변수의 자료 타입을 확인하고 자료타입 별 함수제작 (오버로딩)
- int, double, char, class, ...

형식

- template < typename Type > // template < class Type >
 data type == Type

[오버로딩(overloading): 함수의 이름이 같고 자료 타입을 다르게 선언]

```
double d1 = 12.34, d2 = 56.78;
#include <iostream>
                                                     cout << "변경 전: " << d1 << ", " << d2 << " / ";
#include <cstring>
                                                     f_swap(d1, d2);
using namespace std;
template <typename T1>// == template <class T1> cout << "변경 후: " << d1 << ", " << d2 << endl
void f_swap(T1\& val1, T1\& val2) {
                                                    << endl;
                                                     char c1 = 'A'. c2 = 'B':
 T1 temp = val1;
                                                     cout << "변경 전: " << c1 << ". " << c2 << " / ";
 val1 = val2;
                                                     f_swap(c1, c2);
 val2 = temp;
 cout << typeid(val1).name() << " swap call\n"; //
                                                     cout << "변경 후: " << c1 << ", " << c2 << endl
typeid(val1).name() == val1의 자료형
                                                    << endl:
                                                     char str1[255] = "C++ language", str2[255] = "Well
                                                    done!";
template <class T1>
                                                     cout << "변경 전: " << str1 << ". " << str2 << " / ";
void str_swap(T1& p1, T1& p2) {
 T1 temp;
                                                     str_swap(str1, str2);
                                                     cout << "변경 후: " << str1 << ", " << str2 << endl
 strcpy_s(temp, strlen(p1) + 1, p1);
 strcpy_s(p1, strlen(p2) + 1, p2);
                                                    << endl:
 strcpy_s(p2, strlen(temp)+1, temp);
                                                     return 0;
 cout << typeid(p1).name() << " swap call\n";</pre>
int main() {
 int i1 = 123. i2 = 456;
 cout << "변경 전: " << i1 << ", " << i2 << " / ";
 f_swap(i1, i2);
 cout << "변경 후: " << i1 << ", " << i2 << endl
<< endl;
```

```
#include <iostream>
using namespace std:
template <typename T1, typename T2, typename T3>
void showData(T1 a, T2 b, T3 c) {
   cout << endl:
   cout << a << "\t" << b << "\t" << endl:
   cout << typeid(a).name() << "\t" << typeid(b).name() << "\t" << typeid(c).name() << endl:
}
int main() {
   showData(10, 20, 30);
   showData('A', 'B', 'C');
   showData(10, 3.14, "C++");
   return 0;
}</pre>
```

```
#include <iostream>
using namespace std;
template <typename T>
void print(T n) { // 템플릿 기본형 함수
 cout << "마지막 인수 1개 => ";
 cout << n << endl;
template <typename T1, typename... T2> //템플릿 파라미터 팩
void print(T1 arg, T2... args) { //함수 파리미터 팩
 cout << "가변인수 => ";
 cout << arg << endl;
 print(args...);
int main() {
 print(1.2345);
 print(1, 2.5, "C++");
 print(1, 2, "Hello", 'a', 32.14, 78, 56, "apple");
 return 0;
```

특수화 (Specialization)

```
함수의 기능을 특정 자료형에서 다른 알고리즘으로 처리 할 경우 특수화
```

- 함수의 이름 아래 모든 경우 오버로딩이 가능하다.
 템플릿이 아닌 함수
 템플릿 함수
 명시적 특수화 템플릿 함수
- 명시적 특수화는 반드시 작성 template<> 반환형 함수 이름 <typename> (매개변수 리스트) { 함수 정의 }
- 우선순위: 일반함수 > 템플릿 특수화 함수 > 템플릿 함수

구체화 (Instantiation) 컴파일 과정에서 특정 자료형에 맞는 함수를 생성.

void swap(T a, T b);

int a, b;

- swap(a,b); // void swap(int a, int b) 함수 생성, 암시적 구체화
- template void swap<int>(int a, int b); // swap(int, int) 함수 생성, 명시적 구체화

```
#include <iostream>
#include <cstring>
using namespace std;
template <typename T>
T Max(const T a, const T b) {
  return a > b? a : b;
template <> // char* 형에 대해서 특수화
char* Max<char*>(char* a, char* b) {
  cout << "char* Max <char*> (char* a, char* b)" << endl;
  return strcmp(a, b) > 0 ? a : b; // 문자 a <-> z 비교
template <> // const char* 형에 대해서 특수화
const char* Max<const char*>(const char* a, const char* b) {
  cout << "const char* Max <const char*> (const char* a, const char* b)" << endl;</pre>
  return strlen(a) > strlen(b)? a:b;// 문자열길이 비교
//특수화하는 자료형 정보를 생략하건 생략하지 않건 그 의미하는 바에 차이는 없으나,(컴파일 과정에서 처리)
//가급적이면 자료형 정보를 명시하는 것이 뜻을 명확히 하는 방법
int main() {
  cout << Max(11, 15) << endl; //Max<int>(11, 15) <int> 생략
  cout << Max('T', 'Q') << endl; //Max<char>('T', 'Q') <char> 생략
  cout << Max(3.5, 7.5) << endl; //Max<double>(3.5, 7.5) <double> 생략
  char str1[] = "Simple", str2[] = "Balance";
  cout << Max(str1, str2) << endl << endl; //Max<char*>(str1, str2) <char*> 생략
  cout << Max<char*>(str1, str2) << endl;</pre>
  cout << Max<const char*>(str1, str2) << endl;</pre>
  return 0:
```

```
#include <iostream>
using namespace std;
template <typename T>
int GetSize(T data) {
 cout << typeid(data).name() << " 타입 함수 호출\n";
 return sizeof(data); //data의 바이트(byte)를 리턴
template <>
int GetSize <char*>(char* data) { //템플릿의 특수화 : 특정 자료형의 기능을 다르게 할 경우 사용!!
 cout << typeid(data).name() << " 타입 함수 템플릿 특수화1\n";
 return strlen(data)+1; //data 문자열의 길이를 리턴
template <>
int GetSize <const char*>(const char* data) { //템플릿의 특수화
 cout << typeid(data).name() << " 타입 함수 템플릿 특수화2\n";
 return strlen(data)+1; //data 문자열의 길이를 리턴
int main() {
 cout << GetSize(5) << "바이트\n";
 cout << GetSize(3.14) << "바이트\n";
 cout << GetSize('A') << "바이트\n";
 cout << GetSize((char*)"Hello World") << "바이트\n";
 cout << GetSize("Hello World") << "바이트\n";
 return 0;
```

- 부분 특수화(Partial specialization)
- 템플릿에서 인자의 수는 메인 템플릿 인자 수와 다를 수 있다.
- 함수 (템플릿 매개변수 + 일반 매개변수)

특수화를 많이 할 경우 소스코드가 늘어날 수는 있지만 결국 목적코드로 생성되는 기계어 코드의 양은 같음

템플릿 특수화(specialization)/부분 특수화(Partial specialization) 주의사항 부분 특수화에서 T의 타입이 결정되는 방식을 주의 (다른 템플릿에 중복되는 경우 발생) 부분 특수화에서 기본 파라미터는 표시하지 않음 (메인 템플릿에 정의 값을 그대로 받음) 클래스의 특정 멤버 함수만 특수화 할 수 있으나 부분 특수화는 불가능

```
#include <iostream>
using namespace std;
template <typename T, int count> // 부분 특수화: 템플릿의 매개변수가 2개 이상인 경우 매개변수 중 일부만 특수화
void display(T value) {
  for (int i = 0; i < count; i++) cout << value << ' ';
    cout << endl;
}
int main() {
  display<int, 5>(0); //<int, 5> 반드시 작성
  display<char, 10>('A'); //<char, 10> 반드시 작성
  display<double, 3>(3.4); // <double, 3> 반드시 작성
  return 0;
}
```

클래스 템플릿(Class Template)

typedef 사용의 한계

- 자료형에 따라 코드 수정 필요
- 프로그램에서 원소 자료형 1가지 스택만 사용 가능 (data type = int, struct,… 중 1개만)

형식

- template < typename Type > // template < class Type >
 data type == Type

```
#include <iostream>//클래스템플릿: 클래스가 아닌 클래스를 만드는 틀!!!
using namespace std;
template <typename T>
class Dummy {
private:
  Ta;
public:
  Dummy(Ta);
  inline void SetA(T a);
  inline T GetA() const;
};
template <typename T>
Dummy < T > :: Dummy (T a) : a(a) {}
template <typename T>
inline void Dummy<T>::SetA(T a) { this->a = a; }
template <typename T>
inline T Dummy<T>::GetA() const { return a; }
int main() {
  Dummy<int> d1(10);
  cout << d1.GetA() << endl; // 10
  Dummy<int> d2('A');
  cout << d2.GetA() << endl; // 'A' = 65
  Dummy<char>d3('A');
  cout << d3.GetA() << endl; // 'A' = 65
  Dummy<double>d4(3.14);
  cout << d4.GetA() << endl; // 3.14
  Dummy<const char*> d5("C++");
  cout << d5.GetA() << endl; // "C++"
  return 0:
```

```
#include <iostream>//클래스 템플릿: 클래스가 아닌 클래스를 만드는 틀!!!
using namespace std;
template <typename T1, typename T2>
class dosomething
private:
 T1 a;
 T2 b;
public:
 dosomething();
 dosomething(T1 a, T2 b);
 inline T1 getA() const { return a; }
 inline T2 getB() const { return b; }
 inline void setA(T1 a) { this->a = a; }
 inline void setB(T2 b) { this->b = b; }
};
template <typename T1, typename T2>
dosomething\langle T1, T2 \rangle :: dosomething() : a(0), b(0) {}
template <typename T1, typename T2>
dosomething\langle T1, T2 \rangle :: dosomething(T1 a, T2 b) : a(a), b(b) {}
int main()
 dosomething<int, int>d1(1, 10);
 cout << "d1 = " << d1.getA() << ", " << d1.getB() << endl;
 dosomething<int. char>d2('A'. 'B');
 cout << "d2 = " << d2.getA() << ", " << d2.getB() << endl;
 dosomething<char, char>d3('A', 'B');
 cout << "d3 = " << d3.getA() << ", " << d3.getB() << endl;
 dosomething<double, char> d4(3.14, 'Q');
 cout << "d4 = " << d4.getA() << ", " << d4.getB() << endl;
 dosomething<const char*, const char*> d5("C++", "Well done");
 cout << "d5 = " << d5.getA() << ", " << d5.getB() << endl;
 return 0;
```

템플릿 (Template)을 활용한 stack 클래스 작성, main()에서 다양한 자료형 사용

```
template <typename T, int size> //타입(자료형)이 아닌 인수 전달
class stack { //stack클래스 템플릿
private:
 T* arr; //동적 메모리의 주소를 저장 int top; //배열의 저장 위치
public:
 stack() {}
 ~stack() {}
 void push(T value) {}
 inline void pop() {}
 inline void clear() {}//스택 초기화 inline T getTop() const {}//마지막 저장 값 리턴
 void print() const;
 bool isEmpty();
```

```
#include <iostream>
                                           typedef char element; // 스택 자료형 변경
                                                                                            cout << "\n\t\t[" << stk.getTop()
                                                                                      << "] pop!!!\n";
                                           #define count 100 // 스택 수량 변경
using namespace std;
template <typename T, int size> //타입(자 int main(){
                                                                                            stk.pop();
료형)이 아닌 인수전달
                                            stack<element, count> stk;
class stack { //stack클래스 템플릿
                                            while (true){
                                                                                          break;
                                                                                         case 3:
private:
                                              system("cls");
           //동적 메모리의 주소를 저장
                                              cout << "\n\n\t\t *** stack *** \n\n";
 T* arr;
                                                                                          stk.clear();
                                              cout << "1. push 2. pop 3. clear 4.
 int top;
           //배열의 저장 위치
                                                                                          break;
                                           print 0. terminate\n";
                                                                                         case 4:
public:
 stack() \{ arr = new T[size]; top = 0; \}
                                              cout << "choice: ";
                                                                                          stk.print();
 ~stack() { delete[] arr; }
                                              int choice;
                                                                                          break;
 void push(T value) { if (top < size)</pre>
                                             cin >> choice;
                                                                                         case 0:
{ arr[top] = value; top++; } }
                                                                                          return 0; //프로그램 종료
                                             if (cin.fail()) {
                                               cout << "Wrong Input. Retry" << endl;</pre>
 inline void pop() { if (top > 0)top--; }
 inline void clear() { top = 0; }//스택 초기화
                                               cin.clear();
                                                                                         cout << "\n\n\t\t";
 inline T getTop() const { return arr[top -
                                               cin.ignore(256, '\n');
                                                                                         system("pause");
1]; }//마지막 저장 값 리턴
 void print() const;
                                              element data=0;
                                                                                       return 0;
 boolisEmpty();
                                              switch (choice){
                                              case 1:
template <typename T, int size>
                                               cout << "value push: ";
void stack<T, size>::print() const{
                                               cin >> data;
 cout << "stack display (LIFO): ";
                                               if (cin.fail()) {
 for (int i = top - 1; i >= 0; i--) cout << arr[i]
                                                cout << "Wrong Input. Retry" << endl;
<< ' ';
                                                cin.clear();
                                                cin.ignore(256, '\n');
 cout << endl;
template <typename T, int size>
                                               else stk.push(data);
bool stack<T, size>∷isEmpty(){
                                               break;
 if (top == 0) return true;
                                             case 2:
 else return false;
                                               if (!stk.isEmpty())
```

표준 템플릿 라이브러리 (STL, Standard Template Library) 템플릿들의 집합 컨테이너(Container): 자료구조 이터레이터(Iterator): 데이터 접근 객체 함수 객체(Function Object): 함수 역할의 객체 알고리즘(Algorithm): 알고리즘 함수(정렬, 검색, 비교, … 등)

이터레이터(Iterator: 반복자, 포인터) 일반화 프로그램에서 데이터를 접근하는 객체 배열, 연결리스트, 큐 등 다양함(컨테이너 or 알고리즘)에 얽매이지 않고 저 장된 data를 접근할 수 있는 일반화된 표현

Container<data>∷iterator itr = Container.begin(), itr1, itr2;

- 데이터 값을 접근 할 수 있다. [내용 참조 기능 : (*itr)]
- 다른 이터레이터에 대입 할 수 있다. [대입 가능 : itr1=itr]
- 다른 이터레이터와 비교 할 수 있다. [비교 가능 : itr==itr1, itr != itr2]
- 컨테이너의 모든 원소를 접근 할 수 있다. [증감 연산 : ++itr, itr++]

_Fn for_each(_InIt _First, _InIt _Last, _Fn _Func) 함수포인터 for_each(시작 ITR, 종단 ITR, 함수 포인터)