03

클래스와 객체

연습 - 이론문제

5. 다음 코드는 Circle 클래스의 선언부이다. 틀린 부분을 수정하라.

```
class Circle {
   int radius;
   double getArea();
}
```

6. 다음 코드는 Tower 클래스를 작성한 사례이다. 틀린 부분을 수정하라.

```
class Tower {
   int height = 20;
public:
   Tower() { height = 10; return; }
};
```

7. 다음 코드에서 틀린 부분을 수정하라.

```
class Building {
private:
   int floor;
public:
   Building(int s) { floor = s; }
};
int main() {
   Building twin, star;
   Building BlueHouse(5), JangMi(14);
}
```

8. 다음 코드는 Calendar 클래스의 선언부이다. year를 10으로 초기화하는 생성자와 year 값을 리턴하는 getYear()를 구현하라.

```
class Calendar {
private:
   int year;
public:
   Calendar();
   int getYear();
};
```

9. 생성자에 대한 설명 중 틀린 것은?

① 생성자의 이름은 클래스 이름과 같다.

② 생성자는 오직 하나만 작성 가능하다.

③ 생성자는 리턴 타입을 가지지 않는다.

④ 생성자가 선언되어 있지 않으면 컴파일러에 의해 기본 생성자가 삽입된다.

10. 소멸자에 대한 설명 중에 틀린 부분을 지적하라.

> 소멸자는 ① 객체가 소멸되는 시점에 자동으로 호출되는 멤버 함수로서 ② 클래스의 이름 앞에 ~를 붙인 이름으로 선언되어야 한다. ③ 매개 변수 있는 소멸자를 작성하여 소멸 시에 의미 있는 값을 전달할 수 있으며, 소멸자가 선언되어 있지 않으면 ④ 기본 소멸자가 자동으로 생성된다.

11. 다음 프로그램에 대해 답하여라.

```
class House {
   int numOfRooms;
   int size;
public:
   House(int n, int s); // n과 s로 numOfRooms, size를 각각 초기화
};
void f() {
   House a(2,20);
}
House b(3,30), c(4,40);
int main() {
   f();
   House d(5,50);
}
```

(1) n과 s로 numOfRooms, size를 각각 초기화하고, 이들을 출력하는 생성자를 구현하라.

(2) size와 numOfRooms 값을 출력하는 House 클래스의 소멸자를 작성하라.

(3) 객체 a,b,c,d가 생성되는 순서와 소멸되는 순서는 무엇인가?

12. 다음 프로그램에서 객체 a,b,c가 생성되고 소멸되는 순서는 무엇인가?

```
class House {
   int numOfRooms;
   int size;
public:
   House(int n, int s) { numOfRooms = n; size = s; }
   void test() {
      House a(1,10);
   }
};
void f() {
   House b(2,20);
   b.test();
}
House c(3,30);
int main() {
   f();
}
```

13. 다음 프로그램의 오류를 지적하고 수정하라.

```
class TV {
   TV() { channels = 256; }
public:
   int channels;
   TV(int a) { channels = a;}
};
int main() {
   TV LG;
   LG.channels = 200;
   TV Samsung(100);
}
```

14. 다음 프로그램의 오류를 지적하고 수정하라.

```
class TV {
   int channels;
public:
   int colors;
   TV() { channels = 256; }
   TV(int a, int b) { channels = a; colors = b; }
};
int main() {
   TV LG;
   LG.channels = 200;
   LG.colors = 60000;
   TV Samsung(100, 50000);
}
```

15. 다음 코드에서 자동 인라인 함수를 찾아라.

```
class TV {
   int channels;
public:
   TV() { channels = 256; }
   TV(int a) { channels = a; }
   int getChannels();
};
inline int TV::getChannels() { return channels; }
```

16. 인라인 함수의 장단점을 설명한 것 중 옳은 것은?

① 인라인 함수를 사용하면 컴파일 속도가 향상된다.

② 인라인 함수를 이용하면 프로그램의 실행 속도가 향상된다.

③ 인라인 함수를 사용하면 프로그램 작성 시간이 향상된다.

④ 인라인 함수를 사용하면 프로그램의 크기가 작아져서 효과적이다.

17. 인라인 함수에 대해 잘못 설명한 것은?

① 인라인 선언은 크기가 큰 함수의 경우 효과적이다.

② C++ 프로그램에는 크기가 작은 멤버 함수가 많기 때문에 이들을 인라인으로 선언하면 효과적이다.

③ 컴파일러는 먼저 인라인 함수를 호출하는 곳에 코드를 확장시킨 후 컴파일한다.

④ 인라인 함수는 함수 호출에 따른 오버헤드를 줄이기 위한 방법이다.

18. inline 선언은 강제 사항이 아니다. 다음 함수 중에서 컴파일러가 인라인으로 처리하기에 가장 바람직한 것은?

①
```
inline int big(int a, int b) {
   return a>b?a:b;
}
```

②
```
inline int sum(int a, int b) {
   if(a>=b) return a;
   else return a + sum(a+1, b);
}
```

③
```
inline void add(int a, int b) {
   int sum=0;
   for(int n=a; n<b; n++)
      sum += n;
}
```

④
```
inline int add(int a) {
   static int x = 0;
   x += a;
   return x;
}
```

연습 - 실습문제

1. main()의 실행 결과가 다음과 같도록 Tower 클래스를 작성하라. 난이도 3

```
#include <iostream>
using namespace std;

int main() {
   Tower myTower; // 1 미터
   Tower seoulTower(100); // 100 미터
   cout << "높이는 " << myTower.getHeight() << "미터" << endl;
   cout << "높이는 " << seoulTower.getHeight() << "미터" << endl;
}
```

```
높이는 1미터
높이는 100미터
```

```
#include <iostream>
using namespace std;

int main() {
    Tower myTower; // 1 미터
    Tower seoulTower(100); // 100 미터
    cout << "높이는 " << myTower.getHeight() << "미터" << endl;
    cout << "높이는 " << seoulTower.getHeight() << "미터" << endl;
}
```

2. 날짜를 다루는 Date 클래스를 작성하고자 한다. Date를 이용하는 main()과 실행 결과는 다음과 같다. 클래스 Date를 작성하여 아래 프로그램에 추가하라. 난이도 6

```
#include <iostream>
using namespace std;

int main() {
   Date birth(2014, 3, 20); // 2014년 3월 20일
   Date independenceDay("1945/8/15"); // 1945년 8월 15일
   independenceDay.show();
   cout << birth.getYear() << ',' << birth.getMonth() << ',' << birth.getDay() << endl;
}
```

```
1945년8월15일
2014,3,20
```

힌트 Hint

<string> 헤더 파일의 stoi() 함수를 이용하면 string의 문자열을 숫자로 변환할 수 있다. stoi()는 C++11 표준부터 삽입되었다.

```
string s = "1945";
int n = stoi(s); // n은 정수 1945. VS 2008에는 int n = atoi(s.c_str());
```

```cpp
#include <iostream>
using namespace std;

int main() {
   Date birth(2014, 3, 20); // 2014년 3월 20일
   Date independenceDay("1945/8/15"); // 1945년 8월 15일
   independenceDay.show();
   cout << birth.getYear() << ',' << birth.getMonth() << ',' << birth.getDay() << endl;
}
```

Date 클래스

- 멤버변수 year
- 멤버변수 month
- 멤버변수 day

- 생성자 (매개변수 int, int, int)
- 생성자 (매개변수 string)
- 멤버함수 show()
- 멤버함수 int getYear()
- 멤버함수 int getMonth()
- 멤버함수 int getDay()

2. 날짜를 다루는 Date 클래스를 작성하고자 한다. Date를 이용하는 main()과 실행 결과는 다음과 같다. 클래스 Date를 작성하여 아래 프로그램에 추가하라. 난이도 6

```
#include <iostream>
using namespace std;

int main() {
   Date birth(2014, 3, 20); // 2014년 3월 20일
   Date independenceDay("1945/8/15"); // 1945년 8월 15일
   independenceDay.show();
   cout << birth.getYear() << ',' << birth.getMonth() << ',' << birth.getDay() << endl;
}
```

Date birth(2014, 3, 30);

Date independenceDay("1945/8/15");

(참고) string 클래스의 멤버함수

string str; // 로 선언되어 있다고 가정한다

▶ string의 특정 원소 접근

str.at(index)	index 위치의 문자 반환. 유효한 범위인지 체크 O
str[index]	index 위치의 문자 반환. 유효한 범위인지 체크 X. 따라서 at 함수보다 접근이 빠름
str.front()	문자열의 가장 앞 문자 반환
str.back()	문자열의 가장 뒤 문자 반환

출처: https://rebro.kr/53

(참고) string 클래스의 멤버함수

string str; // 로 선언되어 있다고 가정한다

▶ string의 크기

str.length()	문자열 길이 반환
str.size()	문자열 길이 반환 (length와 동일)
str.capacity()	문자열이 사용중인 메모리 크기 반환
str.resize(n)	string을 n의 크기로 만듦. 기존의 문자열 길이보다 n이 작다면 남은 부분은 삭제하고, n이 크다면 빈공간으로 채움
str.resize(n, 'a')	n이 string의 길이보다 더 크다면, 빈 공간을 'a'로 채움
str.shrink_to_fit()	string의 capacity가 실제 사용하는 메모리보다 큰 경우 낭비되는 메모리가 없도록 메모리를 줄여줌
str.reserve(n)	size = n만큼의 메모리를 미리 할당해줌
str.empty()	str이 빈 문자열인지 확인

출처: https://rebro.kr/53

(참고) string 클래스의 멤버함수

string str; // 로 선언되어 있다고 가정한다

▶ string에 삽입, 추가, 삭제

str.append(str2)	str 뒤에 str2 문자열을 이어 붙여줌 ('+' 와 같은 역할)
str.append(str2, n, m)	str 뒤에 'str2의 n index부터 m개의 문자'를 이어 붙여줌
str.append(n, 'a')	str 뒤에 n개의 'a'를 이어 붙여줌
str.insert(n, str2)	n번째 index 앞에 str2 문자열을 삽입함.
str.replace(n, k, str2)	n번째 index부터 k개의 문자를 str2로 대체함
str.clear()	저장된 문자열을 모두 지움
str.erase(n, m)	n번째 index부터 m개의 문자를 지움
str.erase(n, m) (iterator)	n~m index의 문자열을 지움 (n과 m은 iterator)
str.erase()	clear와 같은 동작
str.push_back(c)	str의 맨 뒤에 c 문자를 붙여줌
str.pop_back()	str의 맨 뒤의 문자를 제거
str.assign(str2)	str에 str2 문자열을 할당. (변수 정의와 동일)

출처: https://rebro.kr/53

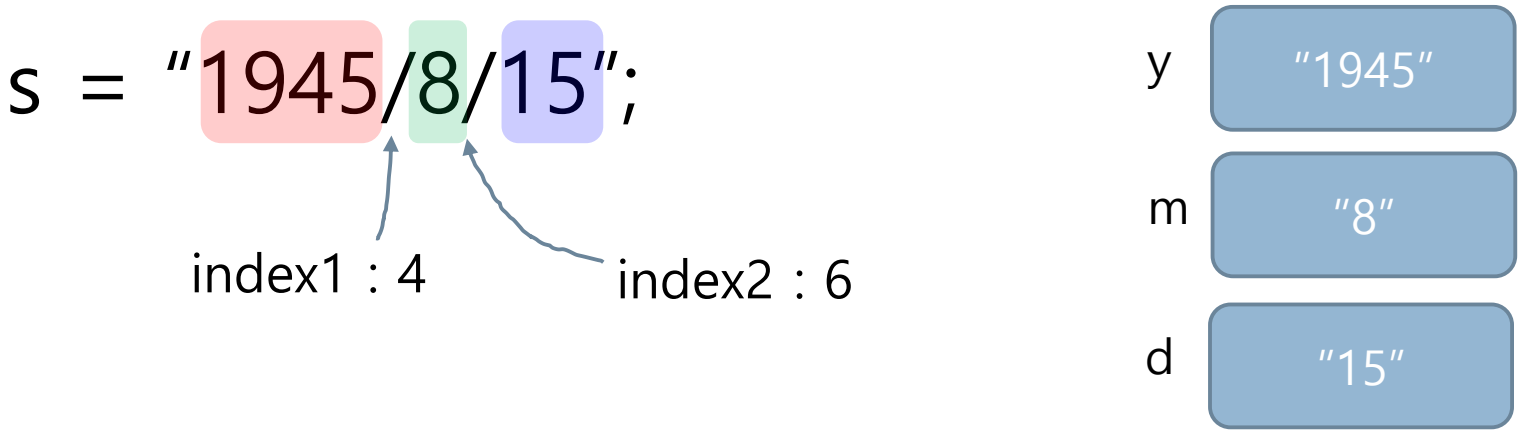
(참고) string 클래스의 멤버함수

string str; // 로 선언되어 있다고 가정한다

▶ 기타 유용한 string 멤버 함수

str.find("abcd")	"abcd"가 str에 포함되어있는지를 확인. 찾으면 해당 부분의 첫번째 index를 반환
str.find("abcd", n)	n번째 index부터 "abcd"를 find
str.substr()	str 전체를 반환
str.substr(n)	str의 n번째 index부터 끝까지의 문자를 부분문자열로 반환
str.substr(n, k)	str의 n번째 index부터 k개의 문자를 부분문자열로 반환
str.compare(str2)	str과 str2가 같은지를 비교. 같다면 0, str<str2 인 경우 음수, str>str2 인 경우 양수를 반환
swap(str1, str2)	str1과 str2를 바꿔줌. reference를 교환하는 방식
isdigit(c)	c 문자가 숫자이면 true, 아니면 false를 반환
isalpha(c)	c 문자가 영어이면 true, 아니면 false를 반환
toupper(c)	c 문자를 대문자로 변환
tolower(c)	c 문자를 소문자로 변환

출처: https://rebro.kr/53




```
int index1 = s.find("/");
string y = s.substr(0, index1); // 인덱스가 0부터 index1 개수만큼 추출
int index2 = s.find("/", index1+1); // 인덱스가 index1 +1 부터 찾음
string m = s.substr(index1+1,                    );
string d = s.substr(index2+1, s.length());
```

3.★ 은행에서 사용하는 프로그램을 작성하기 위해, 은행 계좌 하나를 표현하는 클래스 Account가 필요하다. 계좌 정보는 계좌의 주인, 계좌 번호, 잔액을 나타내는 3 멤버 변수로 이루어진다. main() 함수의 실행 결과가 다음과 같도록 Account 클래스를 작성하라. 난이도 4

```
int main() {
   Account a("kitae", 1, 5000);    // id 1번, 잔액 5000원, 이름이 kitae인 계좌 생성
   a.deposit(50000);               // 50000원 저금
   cout << a.getOwner() << "의 잔액은 " << a.inquiry() << endl;
   int money = a.withdraw(20000); // 20000원 출금. withdraw()는 출금한 실제 금액 리턴
   cout << a.getOwner() << "의 잔액은 " << a.inquiry() << endl;
}
```

```
kitae의 잔액은 55000
kitae의 잔액은 35000
```

힌트 Hint

Account는 name, id, balance(잔액)의 3 멤버 변수와 생성자, getOwner(), deposit(), withdraw(), inquiry()의 3 멤버 함수를 가지는 클래스로 만들면 된다.

```cpp
#include <iostream>
using namespace std;

int main() {
   Date birth(2014, 3, 20); // 2014년 3월 20일
   Date independenceDay("1945/8/15"); // 1945년 8월 15일
   independenceDay.show();
   cout << birth.getYear() << ',' << birth.getMonth() << ',' << birth.getDay() << endl;
}
```

Date 클래스

- 멤버변수 year
- 멤버변수 month
- 멤버변수 day

- 생성자 (매개변수 int, int, int)
- 생성자 (매개변수 string)
- 멤버함수 show()
- 멤버함수 int getYear()
- 멤버함수 int getMonth()
- 멤버함수 int getDay()

9.★ Oval 클래스는 주어진 사각형에 내접하는 타원을 추상화한 클래스이다. Oval 클래스의 멤버는 모두 다음과 같다. Oval 클래스를 선언부와 구현부로 나누어 작성하라. 난이도 6

- 정수값의 사각형 너비와 높이를 가지는 width, height 변수 멤버
- 너비와 높이 값을 매개 변수로 받는 생성자
- 너비와 높이를 1로 초기화하는 매개 변수 없는 생성자
- width와 height를 출력하는 소멸자
- 타원의 너비를 리턴하는 getWidth() 함수 멤버
- 타원의 높이를 리턴하는 getHeight() 함수 멤버
- 타원의 너비와 높이를 변경하는 set(int w, int h) 함수 멤버
- 타원의 너비와 높이를 화면에 출력하는 show() 함수 멤버

Oval 클래스를 활용하는 코드의 사례와 실행 결과는 다음과 같다.

```
#include <iostream>
using namespace std;

int main() {
   Oval a, b(3, 4);
   a.set(10, 20);
   a.show();
   cout << b.getWidth() << "," << b.getHeight() << endl;
}
```

```
width = 10, height = 20
3, 4
Oval 소멸 : width = 3, height = 4
Oval 소멸 : width = 10, height = 20
```