20191128 실습 10주차

박 건 호

KITRI BoB 8th

5422 RTDCS

devgunho.github.io

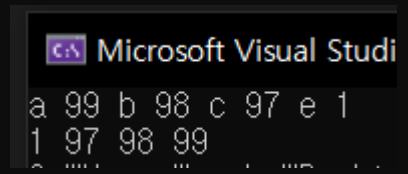
1. Sort ABCD

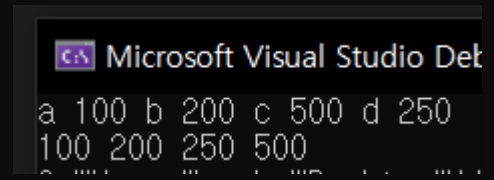
ABCD 값을 문자열 한줄로 입력받고 (공백으로 구분) 크기순으로 정렬된 값을 출력하기

- 값의 개수는 A~Z 까지
- 값의 범위는 0<N<=1,000,000
- 출력은 오름차순으로 '값'만을 출력
- 출력값은 공백으로 구분

string, vector, algorithm 사용 권장

atoi(str.c_str()) 사용 권장





friend (C++)

프렌드 함수나 프렌드 클래스는 객체 지향의 중요한 원칙인 **정보 은닉**을 무너뜨리는 것이 된다.
따라서 반드시 필요한 경우에만 선별적으로 사용하여야 한다.

반드시 필요한 경우?

- 함수가 수행하는 작업이 오직 하나의 객체에만 관련된다면 멤버 함수로 정의한다.
- 함수가 수행하는 작업이 두개 이상의 객체에 관련된다면 프렌드 함수로 정의한다.

물론 함수가 수행하는 작업이 두개 이상의 객체에 관련되더라도 접근자와 설정자를 사용하면 똑같이 구현할 수 있다.

하지만 효율성을 생각하면 프렌드 함수로도 정의할 수 있다.

2. friend (C++)

```cpp
#include <iostream>

using namespace std;
class YourClass {
    friend class YourOtherClass;  // Declare a friend class
public:
    YourClass() : topSecret(0) {}
    void printMember() { cout << topSecret << endl; }
private:
    int topSecret;
};

class YourOtherClass {
public:
    void change(YourClass& YC, int x)
    {
```

```cpp
int main() {
    YourClass yc1;
    YourOtherClass yoc1;
    int input;
    cin >> input;

    yc1.printMember();
    yoc1.change(yc1, input);
    yc1.printMember();
```

```
Micr
123
0
123
```

YourClass 로 생성된 객체 yc1의 멤버 변수인 topSecret의 값을 사용자로부터 입력받은 input값으로 변경하려 한다.

위와 같은 기능이 정상동작하도록 구현하기

3. Complex Number Operation

$$a + bi$$

Real part — Imaginary part

$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

$$(a + bi) - (c + di) = (a - c) + (b - d)i$$

$$(a + bi) * (c + di) = (ac - bd) + (ad + bc)i$$

$$(a + bi) / (c + di) = \frac{ac + bd}{c^2 + d^2} + \frac{bc - ad}{c^2 + d^2}i$$

```
int main()
{
    double a, b, c, d;
    cin >> a >> b >> c >> d;
    Complex c1(a, b), c2(c, d);
    Complex c3 = add(c1, c2);   // call friend Complex add
    c3.print();
    Complex c4 = c1 + c2;   // plus operation overloading
    c4.print();
    Complex c5 = c1 - c2;   // minus operation overloading
    c5.print();

    return 0;
}
```

Class Inheritance

C++에서 클래스 상속이란 기존에 정의되어 있는 클래스의 모든 멤버 변수와 멤버 함수를 물려받아, 새로운 클래스를 작성하는 것을 의미합니다.

이때 기존에 정의되어 있던 클래스를 기초 클래스(base class) 또는 부모 클래스(parent class), 상위 클래스(super class)라고도 합니다.

그리고 상속을 통해 새롭게 작성되는 클래스를 파생 클래스(derived class) 또는 자식 클래스(child class), 하위 클래스(sub class)라고도 합니다.

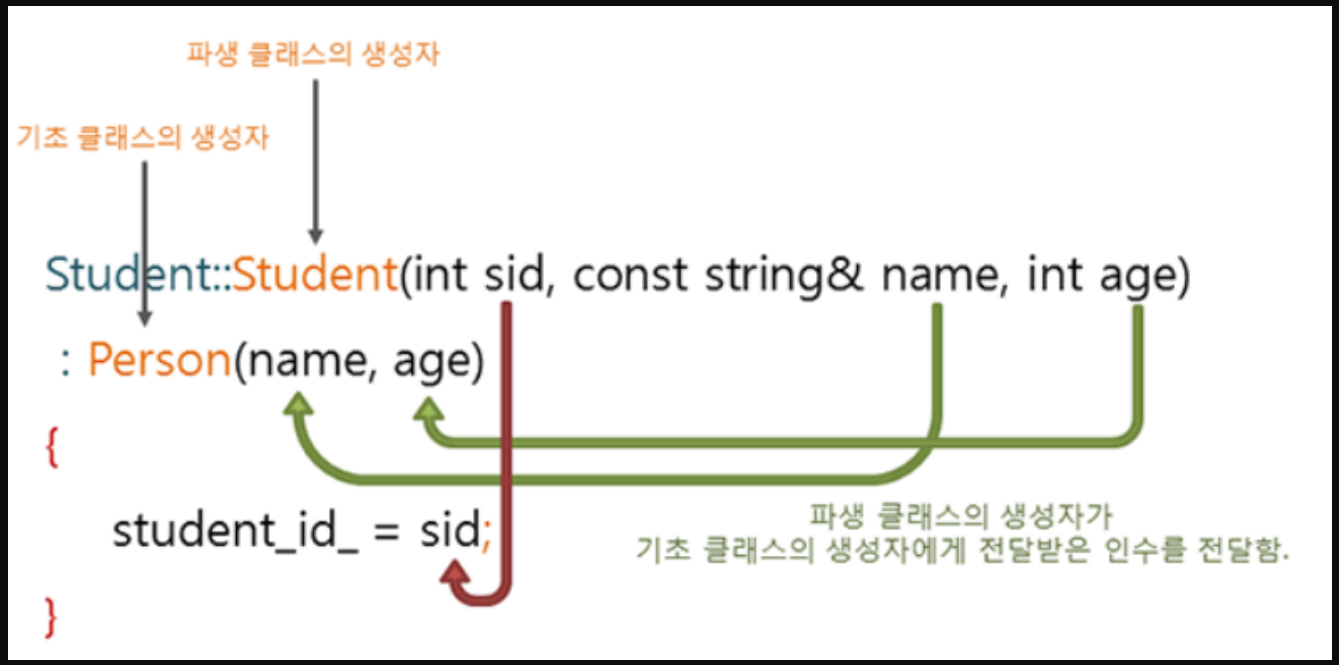
이와 같은 클래스의 상속은 다음과 같은 장점을 가집니다.

1. 기존에 작성된 클래스를 재활용할 수 있습니다.
2. 공통적인 부분은 기초 클래스에 미리 작성하여, 파생 클래스에서 중복되는 부분을 제거할 수 있습니다.

4. Person Class

```cpp
class Person
{
private:
    string name_;
    int age_;
public:
    Person(const string& name, int age);    // 기초 클래스 생성자의 선언
    void ShowPersonInfo();
};

class Student : public Person
{
private:
    int student_id_;
public:
    Student(int sid, const string& name, int age);  // 파생 클래스 생성자의 선언
};
```



4. Person Class

Person Class를 상속받아 작성된 Student Class를 보고,
같은 방식으로 main 함수를 참고하여 정상 동작할 수 있는
Professor Class를 구현하기

```
int main(void)
{
    Student hong(123456789, "홍길동", 20);
    hong.ShowPersonInfo();
    Professor kim(123456777, "컴퓨터전자시스템공학부", "김길동", 50);
    kim.ShowPersonInfo();

    return 0;
}
```

```
Microsoft Visual Studio Debug Console
이름은 홍길동이고, 나이는 20살입니다.
이름은 김길동이고, 나이는 50살입니다.
```

Overloading vs Overriding

Overloading : "과적하다, 과부하"

같은 클래스내에서

같은 이름의 메서드를 사용하는 것

```
class OverloadingTest{
    //이름이 cat인 메서드
    void cat(){
        System.out.println("매개변수 없음");
    }

    //매개변수 int형이 2개인 cat 메서드
    void cat(int a, int b){
        System.out.println("매개변수 :"+a+", "+b);
    }

    //매개변수 String형이 한 개인 cat 메서드
    void cat(String c){
        System.out.println("매개변수 : "+ c);
    }

}
```

Overloading vs Overriding

Override : "무시하다, 무효하다, 기각하다"

Overriding : "가장 우선되는, 최우선되는, 다른것보다 우선인"

부모 Class에서 정의한 메서드를 자식 Class에서 변경하는 것

오버로딩(Overloading)과 오버라이딩(Overriding) 성립조건

구분	오버로딩	오버라이딩
메서드 이름	동일	동일
매개변수, 타입	다름	동일
리턴 타입	상관없음	동일

파생 클래스에서의 오버라이딩

C++에서는 파생 클래스에서 상속받은 기초 클래스의 멤버 함수를 직접 재정의할 수 있습니다.

```
void Person::ShowPersonInfo()
{
    cout << name_ << "의 나이는 " << age_ << "살입니다." << endl;
}

void Student::ShowPersonInfo()
{
    cout << "이 학생의 학번은 " << student_id_ << "입니다." << endl;
}
```

5. Overriding

파생 클래스(Student, Professor)에서 상속받은 기초 클래스(Person)의 멤버함수(ShowPersonInfo)를 재정의 하여 다음과 같은 출력이 나오도록 코드를 구현하기

문제를 해결하기 위해서는 특정 부분의 멤버 변수를 protected로 바꾸어 주어야 한다.

```
int main(void)
{
    Person lee("이순신", 25);
    lee.ShowPersonInfo();
    Student hong(123456789, "홍길동", 20);
    hong.ShowPersonInfo();
    Professor kim(123456777, "컴퓨터전자시스템공학부", "김길동", 50);
    kim.ShowPersonInfo();

    return 0;
}
```

```
Microsoft Visual Studio Debug Console
이름은 이순신이고, 나이는 25살입니다.
홍길동 학생의 학번은 123456789입니다.
교수님의 이름은 김길동, 전공은 컴퓨터전자시스템공학부입니다.

D:₩_Main_ver.19.11.13₩'19 2nd Seme. CP Lecture₩2019_CP_Fall_Lect
 with code 0.
Press any key to close this window . . .
```