

# [lab12 보고서]

자율전공학과

2019111677

김지연

## 1. 소스코드

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

class Student {
private:
    string name;
    string snum;
    string major;
    int grade;
    int credit;

public:
    Student(string n, string s, string m, int g, int c) :name(n), snum(s), major(m),
grade(g), credit(c) {};
    void setName(string name) {};
    void setSnum(string snum) {};
    void setMajor(string major) {};
    void setGrade(int grade) {};
    void setCredit(int credit) {};
    string getName() { return name; }
    string getSnum() { return snum; }
    string getMajor() { return major; }
    int getGrade() { return grade; }
    int getCredit() { return credit; }
    void print();
};

void Student::print() {
    cout << "이름 : " << this->getName() << endl;
    cout << "학번 : " << this->getSnum() << endl;
    cout << "소속학과 : " << this->getMajor() << endl;
    cout << "학년 : " << this->getGrade() << endl;
    cout << "이수학점 : " << this->getCredit() << endl;
}

class UnderGraduate : public Student {
private:
    string club;
public:
    // 학부생의 생성자 및 멤버함수
    UnderGraduate(string name, string snum, string major, int grade, int credit, string
c) :Student(name, snum, major, grade, credit), club(c) {}
    void setClub(string club) {};
    string getClub() { return club; }
    void print();
};

void UnderGraduate::print() {
    Student::print();
    cout << "동아리명 : " << this->getClub() << endl;
}

class Graduate : public Student {
```

```

private:
    string assistant;
    double srate;
public:
    // 대학원생의 생성자 및 멤버함수
    Graduate(string name, string snum, string major, int grade, int credit, string a,
double s) :Student(name, snum, major, grade, credit), assistant(a), srate(s) {}
    void setAssistant(string assistant) {};
    string getAssistant() { return assistant; };
    void setSrate(double srate) {};
    double getSrate() { return srate; }
    void print();
};

void Graduate::print() {
    Student::print();
    cout << "조교유형 : " << this->getAssistant() << endl;
    cout << "장학비율 : " << this->getSrate() << endl;
}

int main() {
    Graduate g("홍길동", "20141111", "컴퓨터학과", 2, 140, "연구조교", 0.5);
    UnderGraduate u("홍길서", "20191112", "디지털미디어학과", 1, 80, "게임프로그래밍");

    g.print();
    cout << "-----" << endl;
    u.print();
    return 0;
}

```

## 2. 실행결과화면

```

Microsoft Visual Studio 디버그 콘솔

이름 : 홍길동
학번 : 20141111
소속학과 : 컴퓨터학과
학년 : 2
이수학점 : 140
조교유형 : 연구조교
장학비율 : 0.5
-----
이름 : 홍길서
학번 : 20191112
소속학과 : 디지털미디어학과
학년 : 1
이수학점 : 80
동아리명 : 게임프로그래밍

C:\Users\wyjysh\source\repos\Project20\
이 창을 닫으려면 아무 키나 누르세요.

```

### 3. 문제 정의 및 분석

- 학생클래스의 이름/학번/학과/학년/이수학점을 상속받아 대학원과 학부 클래스 구현한다.
- 상속이 의미 있게 사용하기 위하여 중복되는 코드인 이름/학번/학과/학년/이수학점은 부모클래스에서 정의하고 자식클래스(학부/대학원)에서 명시적으로 호출하여 사용한다.
- 학부생은 동아리명 멤버변수, 대학원생은 조교유형/장학금비율 멤버변수를 추가한다.
- 새로 추가한 멤버함수에 getter와 setter를 지정하여 접근을 가능하게 한다.
- print함수에 중복되는 코드는 부모클래스에서 바로 호출하고, 새로 정의한 멤버변수만 cout으로 출력할 내용을 작성한다.
- 각각의 생성자는 초기화리스트를 사용하여 초기화 시켜준다.

상속- 자식클래스 뒤에 ':'를 쓰고 부모클래스를 쓴다.

상속의 가장 큰 의미는 코드의 재사용성 (공통되는 코드들을 부모클래스로 만든다)