

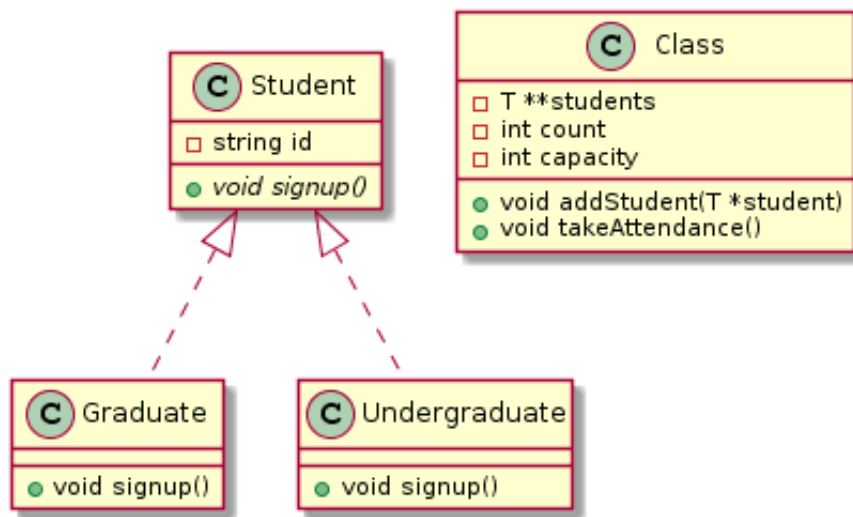
Lab1

学院某助教设计并实现了一个不好用的课程点名系统。

学院目前有两类学生：本科生(undergraduate)和研究生(graduate)。每个学生有一个字符串id表示其学号。此外，每个学生都有一个签到函数(signup)，其被调用时会输出一个字符串表示签到。本科生在签到的时候需要输出一行"Undergraduate student XXX"，其中"XXX"为该学生的学号。研究生在签到的时候需要输出一行"Graduate student XXX"，其中"XXX"为该学生的学号。

现在学院开设三种课程(Class)：本科生课程、研究生课程和通识课程。本科生课程只允许本科生选修；研究生课程只允许研究生选修；而通识课程两类学生均可选修。

在系统设计中，类的关系如下图所示。



基类 Student 是一个抽象类，其中声明了一个字符串类型的成员变量 id 表示学生的学号。同时，基类 Student 中声明了纯虚函数 signup 用于签到。派生类 Graduate 和 Undergraduate 均继承自 Student，并分别实现了 signup 函数。

在 Class 类中 T **student 用于保存选修了该课程的所有学生；capacity 为该课程所允许的最大选课人数；count 为该课程当前的选课人数；addStudent 函数用于向课程中添加的学生，takeAttendance 为课程点名。在 takeAttendance 被调用时，会依次调用所有已选课的学生的 signup 函数。

任务

点名系统的部分代码已经在 lab1.cc 文件中给出，现在需要你将其补全。

代码补全：

1. 实现学生类 Student、Undergraduate 以及 Graduate。
2. 补全 takeAttendance 函数；
3. 将本科生加入到基础课程 basic_class 中；

4. 所有的学生均加入到通识课 `common_class` 中；
5. 完成三门课程的点名。

请回答下述问题：

1. 请解释下述 `addStudent` 实现中第2~3行的作用，并简述如何触发第3行代码。

```
1 void addStudent(T *student) {  
2     if (count >= capacity)  
3         throw out_of_range("Too many students in the class");  
4     students[count++] = student;  
5 }
```

2. 请查阅资料，简述在生成学生信息时所涉及到的 `stringstream` 的各种用法（即解释2~4行）。

```
1 for (int i = 0; i < 12; ++i) {  
2     stringstream ss;  
3     ss << "1180379" << std::setfill('0') << setw(2) << i;  
4     graduate_students[i] = new Graduate(ss.str());  
5 }
```

3. 在按照前述要求完成代码后，请指出最终代码中所存在的一个问题。（不一定会导致程序错误，也可指出不良风格、不良习惯等，答案不唯一）

提交

提交时，请将

- 你完成的点名系统源代码 `lab1-XXX.cc`；
- 你对上述问题的回答 `lab1-XXX.pdf`。

上传到 <ftp://dmkaplony:public@public.sjtu.edu.cn:/upload/c++2019/lab1/> 中。

（其中XXX为学号，如 `lab1-518037910001.cc` 和 `lab1-518037910001.pdf`）