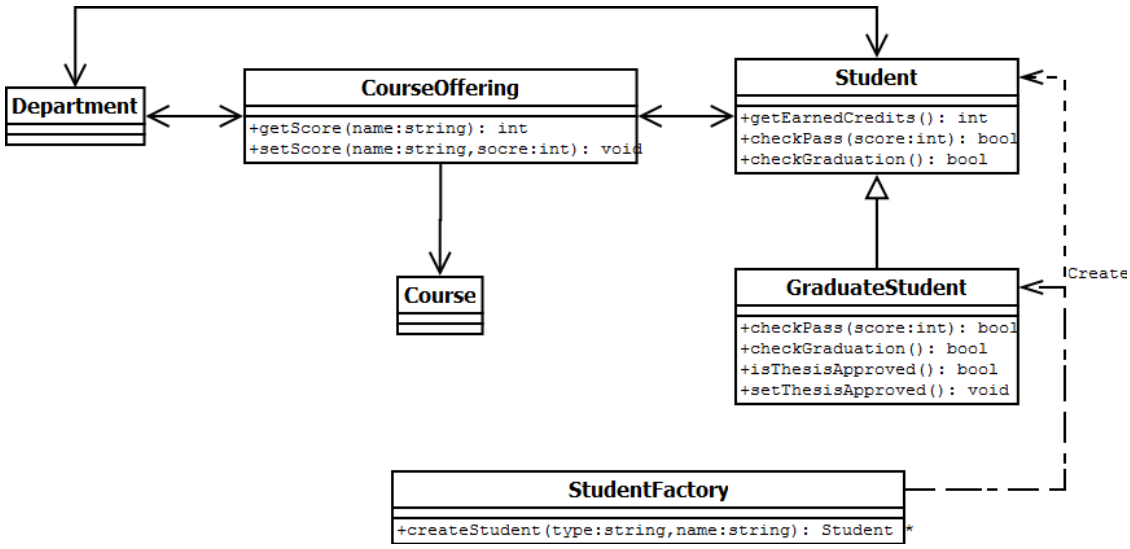
National Taipei University of Technology

Object-Oriented Programming (fall, 2009)

Homework # 9

(Due: 6pm, 8 January 2010)

本次作業主要為練習物件的繼承及多型，主要架構來自 Homework #8 以及上課所講解的範例。



在本次作業中主要修改的部分：

- 新增一種學生的型態 GraduateStudent，並且讓 GraduateStudent 繼承 Student
- 修改原本學生資料檔，新增一個欄位表示學生身分(Student, GraduateStudent)
- 新增一個類別 StudentFactory，主要功能為根據類型建立相對應的物件
- 新增課程分數資料檔，並且透過讀檔的方式將學生修課成績讀入程式中

在本次作業中主要需要撰寫的部分如下：

CourseOffering:

```
// 根據修課學生姓名取得該學生的成績
int getScore(string name);
// 根據修課學生姓名設定該學生的成績
void setScore(string name, int score);
```

Student:

```
// 根據該學生所修習的課程計算實得學分
int getEarnedCredits();
// 判斷分數是否達合格標準 (修課成績大於 60 分)
virtual bool checkPass(int score);
// 判斷該學生是否達畢業標準 (實得學分大於等於畢業學分)
virtaul bool checkGraduation();
// 取得修課總平均
double getScoreAverage();
// 取得系所排名
// 若為 Student 則取得在此系所的 Student 中的排名
// 若為 GraduateStudent 則取得在此系所的 GraduateStudent 中的排名
int getDepRank();
```

GraduateStudent:

```
// 判斷分數是否達合格標準 (修課成績大於 70 分)
virtual bool checkPass(int score);
// 判斷該學生是否達畢業標準 (實得學分大於等於畢業學分並且完成論文)
virtaul bool checkGraduation();
```

StudentFactory:

```
// 根據學生類型建立相對應的物件，並回傳父類別的指標
// 若 Type 為 Student 則建立 Student 物件並回傳
// 若 Type 為 GraduateStudent 則建立 GraduateStudent 物件並回傳
Student * createStudent(string type, string name);
```

Other:

```
// 讀取課程分數資料檔
void loadScoreFromFile(const char *fn, vector<CourseOffering *> courses);
// 讀取學生資料
vector<Student *> createStudentsFromFile(const char * fn, vector<Department *>
depts)
```

注意事項：

- 取得實得學分必須判斷所修習的課程成績是否達到合格標準，並且將達合格標準之課程學分數累加
- 必須通過程式中所有測試
- createStudentsFromFile 方法中，必須使用 StudentFactory 建立學生物件