Problem [70 Marks]

Consider the following diagram in Figure 1 which shows the data model of a course registration application. There are five (5) courses offered as shown in Table 1, but a student can **ONLY register up to three (3) courses**. Table 1 is a list of the courses offered. Based on the class diagram, write a complete C++ program that does the following tasks:

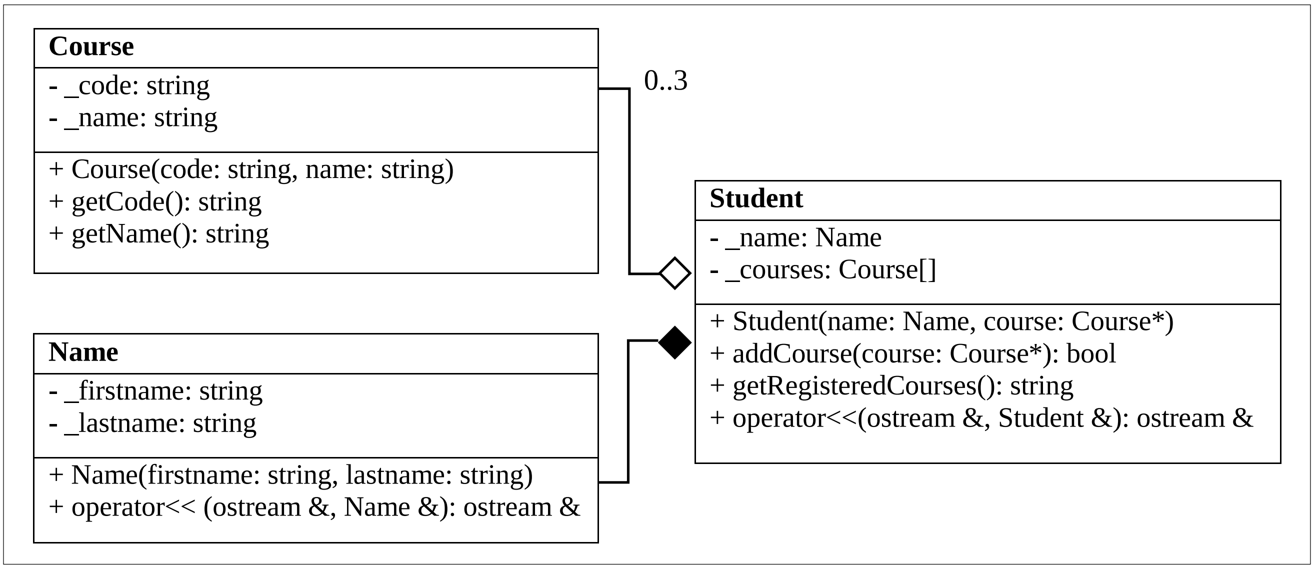
Course
- _code: string
- _name: string
+ Course(code: string, name: string)
+ getCode(): string
+ getName(): string
0..3
Student
- _name: Name
- _courses: Course[]
+ Student(name: Name, course: Course*)
+ addCourse(course: Course*): bool
+ getRegisteredCourses(): string
+ operator<<(ostream &, Student &): ostream &
Name
- _firstname: string
- _lastname: string
+ Name(firstname: string, lastname: string)
+ operator<< (ostream &, Name &): ostream &

Figure 1: Class diagram for a course registration application

Table 1: Courses offered

Code	Name
0101	Cryptography
0234	Programming
1234	Calculus
2356	History
3658	Geography

1. Implement all classes with the given attributes and operations as in Figure 1. Note that, the purpose of each operation is as its name implies. [52 Marks]
   a. Default constructors may be needed for some classes. *(3 marks)*
   b. Class **Name** (13 marks)
      - The overloaded output **operator<<** function in class **Name** can be used to get the full name, which is the first name followed by the last name. *(5 marks)*
   c. Class **Student** (24 marks)
      - The overloaded output **operator<<** function class **Student** can be used to get the full name and all registered courses. *(5 marks)*
      - The function **getRegisteredCourses** in class **Student** will only display the name of the courses registered to the student (space delimited), or “None” if no course is registered. *(5 marks)*

  - The function **addCourse** in class **Student** will return true if successful or false if failed (e.g. maximum number of registered courses reached, or registering an already registered subject). *(6 marks)*

d. Class **Course** (15 marks)

2. Create a **main** function that does ALL of the following tasks. [18 Marks]
   a. Create a list of **Course** objects and use the data given in Table 1. (5 marks)
   b. Create a student with any name and no subjects registered, and print the student's detail (via **operator<<** of class **Student**). (4 marks)
   c. **Registers any two (2)** subjects, and print the student's details (via **operator<<** of class **Student**) (3 marks)
   d. **Register one more** subject that **<u>has already been registered</u>**. Based on the **addCourse** function's return value, display "Registration successful" if successful or "Registration failed" if otherwise. (4 marks)
   e. **Register one more** subject that **<u>has NOT been registered</u>** and print the student's details (via **operator<<** of class **Student**). (2 marks)

Figure 2 shows sample program output.

```
Arif Ariffin registered courses: None

Arif Ariffin registered courses: Cryptography Programming

Registration failed

Arif Ariffin registered courses: Cryptography Programming Geography
```

Figure 2: Sample program output