Object Oriented Programming

Spring 2020

Lab Manual 6

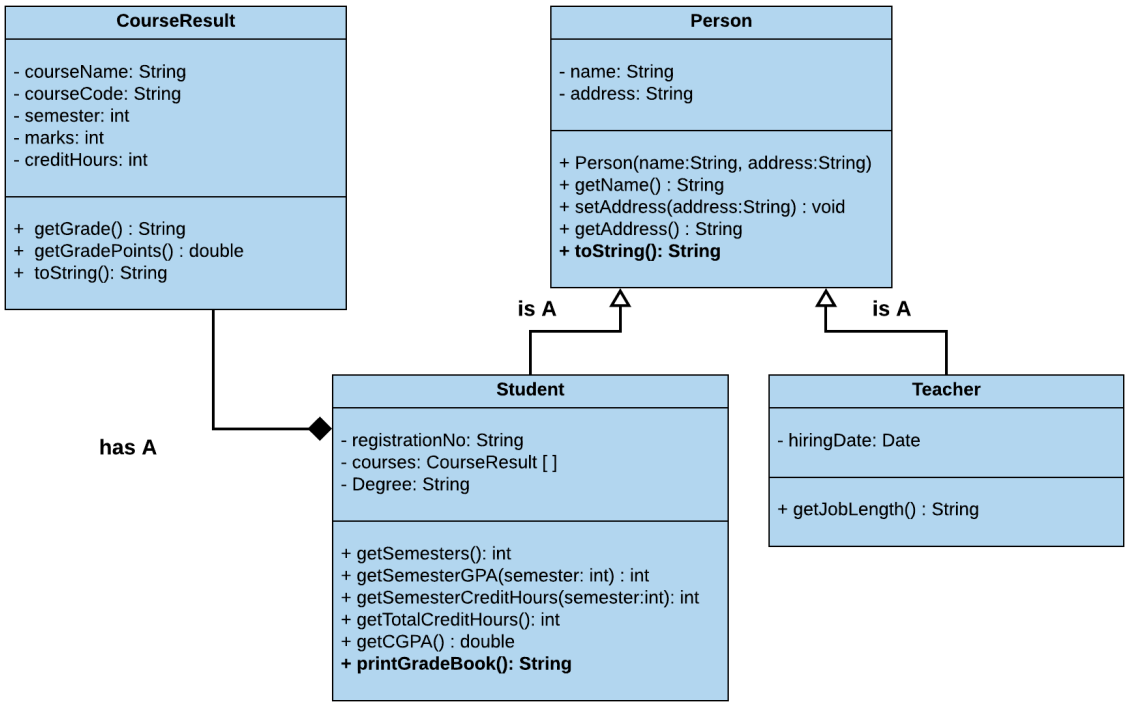
Instructor: Samyan Qayyum Wahla

Deadline:

- July 26th, 2020

Uploading instructions

You should follow the instructions provided in course outline document to get the full credit of assignment.



.

Constraints for each attribute are given below.

- **StudentName** //should be alphabetic, special characters and numbers are not allowed
- **RegistrationNumber** //Format should be like this: 2015-CS-888, any other format should be handled in setter function
- **Degree** //it should be MS, BS or BE
- **CourseID** // Format should be valid according to your course codes given in your LMS. For instance, OOP lab has course ID of CS241L. Length of course code should be from 2 to 8 characters.

- **CourseTitle** // should be alphabetic. Length of course code should be from 10 to 35 characters.
- **CreditHours** // values from 1 to 3 are allowed
- **Marks //** values from 0 to 100 are allowed
- **Semester //** valid range is from 1 to 8

1. Your Program should define three constructors for class of **CourseResult**
   - a constructor with **no parameter**
   - a constructor with **parameters**
   - **copy** constructor
2. Your Program should define three constructors for class of **Teacher**
   - a constructor with **no parameter**
   - a constructor with **parameters**
   - **copy** constructor
3. Your Program should define three constructors for class of **Person**
   - a constructor with **no parameter**
   - a constructor with **parameters**
   - **copy** constructor
4. All classes should have getter, setters and destructor
5. **Student** will have only one constructor without parameter
6. Define getter setter for each data member in classes
7. Apart from getter, setter and constructor, define the following functions in respective classes according to class diagram
   i **getGrade()** – it should calculate grade based on marks using the following criteria.
      a. IF marks are less than 40 – Grade is F
      b. IF marks are between 40 and 50(exclusive) - Grade is D
      c. IF marks are between 50 and 55(exclusive) - Grade is C
      d. IF marks are between 55 and 60(exclusive) - Grade is C+
      e. IF marks are between 60 and 65(exclusive) - Grade is B-
      f. IF marks are between 65 and 70(exclusive) - Grade is B+
      g. IF marks are between 70 and 80(exclusive) - Grade is A-
      h. IF marks are above 80 - Grade is A

   ii **getGradePoints()** – function should return grade points using the following criteria

Grade	CoursePoints
A	4.0
A-	3.7
B+	3.3
B-	3.0
C+	2.7
C	2.3

D	1.0
F	0

iii. **getSemesters()** – it should return number of semesters based on course list iv. **getSemesterGPA(semester: int)** – calculate semester GPA according to following formula

$$SemesterGPA = \frac{\sum SemesterCourseGradePoints}{SemesterCreditHours}$$

v. **getCGPA():** calculate GPA using the following formula

$$CGPA = \frac{\sum CourseGradePoints}{TotalCreditHours}$$

vi. **getTotalCreditHours()** – it should return number of credit hours based on course list

vii. **getSemesterCreditHours(semester: int)** – it should return number of credit hours for a given semester based on course list

viii. **getSession()** – extract session from RegistrationNumber

ix. **getDiscipline()**: extract session from RegistrationNumber

x. **toString()** – Purpose of this function is to write all attributes of a class in desired format and return as a string

xi. **printGradeBook()** – Purpose of this function is to print DMC in desired format and return as a string as shown in output section

8. Further, you must make a new file named **Driver** in which you have to define main function.
9. Declare a new object of **Each Class** in main function

Output

```
Choose the following option:
Choose 1 to set basic information of student
Choose 2 to add new course for the student
Choose 3 to edit a course
Choose 4 to delete a course
Choose 5 to view all course
Choose 6 to view CGPA
Choose 7 to view detailed marks sheet
```

a. On pressing 1, basic information input format is as follow:

```
Please enter the basic information in the following format
Name, Registration Number, Degree, Address
Enter Input: Samyan Qayyum, 2009-CS-01, BS, UET Lahore
```

b. On pressing 2, enter course information in following format

```
Please enter the course information in the following format
Course ID, Course Title, CreditHours, Semester, Marks
Enter Input: CS241L, OOP, 1, 2, 85
```

c. On pressing 3, update course information as follow:

```
Enter Course Id to Update: CS241L
```

```
Please enter the course information in the following format
Course ID, Course Title, CreditHours, Semester, Marks
Enter Input: CS241L, OOP, 1, 2, 85
```

d. On pressing 4, input dialog should ask for course ID to delete a course
e. On pressing 5, course should be shown on output dialog using toString() method in the following format

ID	Name	CH	Marks	Grade
CS381	Software Engineering	3	90	A
CS141	Computing Fundamentals	2	79	A-

f. On pressing 6, CGPA should be shown on output dialog.
g. On Pressing 7, DMC will be shown on output dialog in the following format
   Name: Samyan Qayyum Degree: BS CS
   Registration Number: 2009-CS-01
   Session: 2009

Semester 1:

ID	Name	CH	Marks	Grade
MTH134	Calculus	3	90	A
CS141	Computing Fundamentals	2	79	A-
PHY101	Physics	3	75	A-
			SGPA: 3.8125	

Semester 2:

ID	Name	CH	Marks	Grade
MTH111	Linear Algebra	1	80	A
CS141	Programming Fundamentals	3	65	B+
			SGPA: 3.475	
			CGPA: 3.7	

What to submit

You are simply required to submit a source file (**Student.cpp, Person.cpp, Teahcer.cpp, CourseResult.cpp** and **Driver.cpp**(in which main function is defined)) that includes the implementation of the above mentioned program. No extra file should be submitted.