University of Wollongong
School of Computing and Information Technology
CSIT121 Object Oriented Design and Programming
Assignment 2

Objectives

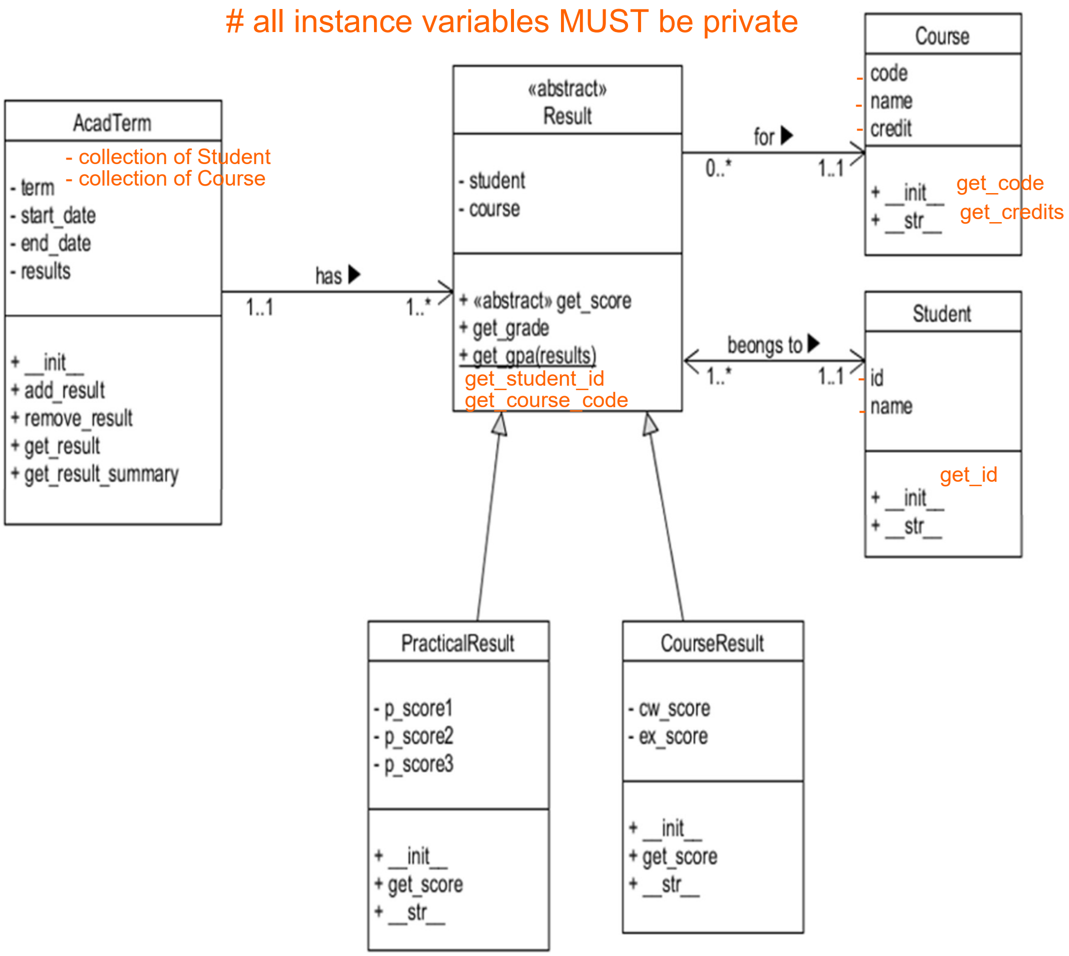
- To apply Object Oriented Design (OOD).
- To apply Object Oriented Programming (OOP) using Python.

Submission

- Please submit **one text** file containing the Python code (with comments) and the execution results (in text form) to UOW Moodle.
- File name must be in the form of: **TXX_NAME_UOWID.txt** where XX is your tutorial group, NAME is your full name and UOWID is your 7-digit UOW ID number. For example, **T02F_JeffreyTan_8080426.txt**
- Late submission will be penalized 25% per day late. Please refer to UOW Moodle for the assignment due date (in Singapore time).

Tasks

Write the Python classes to implement the prototype of an Academic Result Management Application depicted in the ***draft*** class diagram.



Class: Course

Attribute	**Description**
code	The course code.
name	The course name
credit	The number of credits of this course
Method	
__init__	Constructor.
__str__	This method will return a string containing the 3 attributes.

Class: Student

Attribute	**Description**
id	The student id.
name	The student's name
Method	
__init__	Constructor.
__str__	This method will return a string containing the 2 attributes.

Class: Result

Attribute	**Description**
student	The student enrolled in a course.
course	The course taken by a student.
Method	
__init__	Constructor.
get_score	This method will compute and return the overall score. It will be overridden.
get_grade	This method will compute and return the final grade based on the overall score. Please refer to the grading system described below.
__str__	This method will return a string containing the following: • student id and name • course code • overall score • grade
get_gpa	This is a class method that compute and return the GPA for a collection of ***results***.

Class: PracticalResult

Attribute	**Description**
p_score1	The score of practical work 1.
p_score2	The score of practical work 2.
p_score3	The score of practical work 3.
Method	
__init__	Constructor.
get_score	This method will compute and return the overall score as follow: (p_score1 + p_score2 + p_score3) / 3
__str__	This method will return a string containing the following: • student id and name • course code • p_score1 • p_score2 • p_score3 • overall score • grade

Class: CourseResult

Attribute	**Description**
cw_score	The score of the course work.
ex_score	The score of the exam.
Method	
__init__	Constructor.
get_score	This method will compute and return the overall score as follows: cw_score x 0.3 + ex_score * 0.7
__str__	This method will return a string containing the following: • student id and name • course code • cw_score • ex_score • overall score • grade

Class: AcadTerm

Attribute	**Description**
term	Term names such as "2024 Q1", "2024 Q2", etc.
start_date	First day of the academic term.
end_date	Last day of the academic term.
results	A collection of ***results*** of the academic term.
Method	

__init__	Constructor.
add_result	This method will add a result (object of Result subclasses) to the academic term. The method must ensure that the ***same result*** cannot be added multiple times to the same academic term. # also ensure student and course are valid (i.e. found in collections of AcadTerm)
remove_result	Given the student id and course code, this method will remove the matching result from the academic term.
get_result	This method will return all the results for a student (id) or will return the result of a student for a course (if given).
get_result_summary	This method will return the summary result academic term. The summary result include: • term • number of passes • number of failures
__str__	The method will return a string containing the following: • term • start_date • end_date

Grading system

Grade	Score	Points awarded
A	>= 95	4.0
A-	>= 90	3.7
B+	>= 85	3.3
B	>= 80	3.0
B-	>= 75	2.7
C+	>= 70	2.3
C	>= 65	2.0
C-	>= 60	1.7
D+	>= 55	1.3
D	>= 50	1.0
D-	>= 45	0.7
F	< 45	0.0

Grade point average (GPA)
Assuming a student enrolled in three courses and obtained the following results:

Grade	Credit	Points awarded
A	6	4.0 x 6 = 24
B	6	3.0 x 6 = 18
B	6	3.0 x 6 = 18
	Total Credits: 18	Total Points: 24 + 18 + 18 = 60
		GPA: 60 / 18 = 3.33

You will carry out OOD and OOP as follows:

- You must ***not*** use global variables.
- You must choose an appropriate data type (class) for each attribute.
- You must include appropriate properties and setters (or get and set methods).
- You may decide the appropriate parameter(s) for each method.
- You may include additional attributes and methods for each class.
- You must define a main function with helper functions to thoroughly test the functionalities of the program.
- You must include comments in the program.