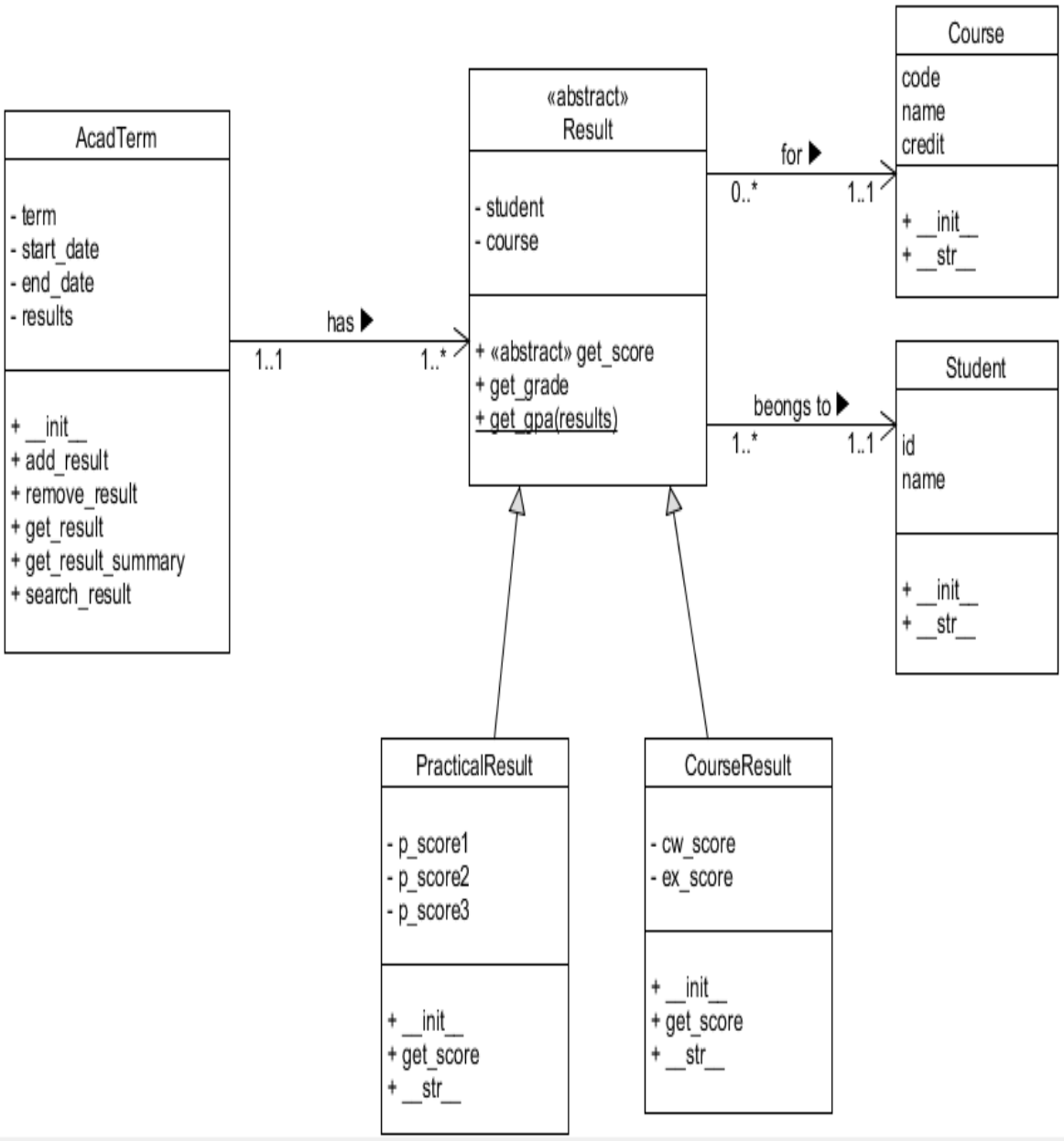
University of Wollongong
School of Computing and Information Technology
CSIT121 Object Oriented Design and Programming
Assignment 2

Objectives

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

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 following: • code. • name. • credit.

Class: Student

Attribute	Description
id	The student id.
name	The student name
Method	
__init__	Constructor.
__str__	This method will return a string containing the following: • id. • name.

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 in the next section.

__str__	This method will return a string containing the following: • student id. • course code. • score. • grade.
get_gpa	This class method will computer 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. • course. • p_score1. • p_score2. • p_score3. • overall score.

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 follow: cw_score x 0.4 + ex_score * 0.6
__str__	This method will return a string containing the following: • student. • course. • cw_score. • ex_score. • overall score.

Class: AcadTerm

Attribute	Description
term	Term name such as "2023 Q1", "2024 Q1", 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 to the academic term. The method must ensure that the ***same result*** cannot be added multiple times to the same academic term.
remove_result	This method will remove a result from the academic term.
get_result	This method will return the result of a student for a course.
get_result_summary	This method will return the summary result academic term. The summary result include: • term. • number of passes. • number of failures.
search_result	This method will return the results of a course (code) or a student (id).
__str__	The method will return a string containing the following: • term. • start_date. • end_date.

Grading system

Grade	Score	Points awarded
A	>= 80	4
B	70 – 79	3
C	60 – 69	2
D	50 – 59	1
F	< 50	0

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

Grade	Credit	Points awarded
A	6	4 x 6 = 24
B	6	3 x 6 = 18
B	6	3 x 6 = 18
	Total points awarded	24 + 18 + 18 = 60
	GPA	60 / 18 = 3.33

You will carry out OOD and OOP as follow:

- You must ***not*** use global variable.
- You must choose an appropriate data type (class) for each attribute.
- You must include appropriate properties and setters (or get and set methods).
- You must decide the 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.

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: TX_Y.txt where X is your tutorial group and Y is your full name.
- Late submission will be penalized 25% per day late. Please refer to UOW Moodle for the assignment due date (in Singapore time).