

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

### Objectives

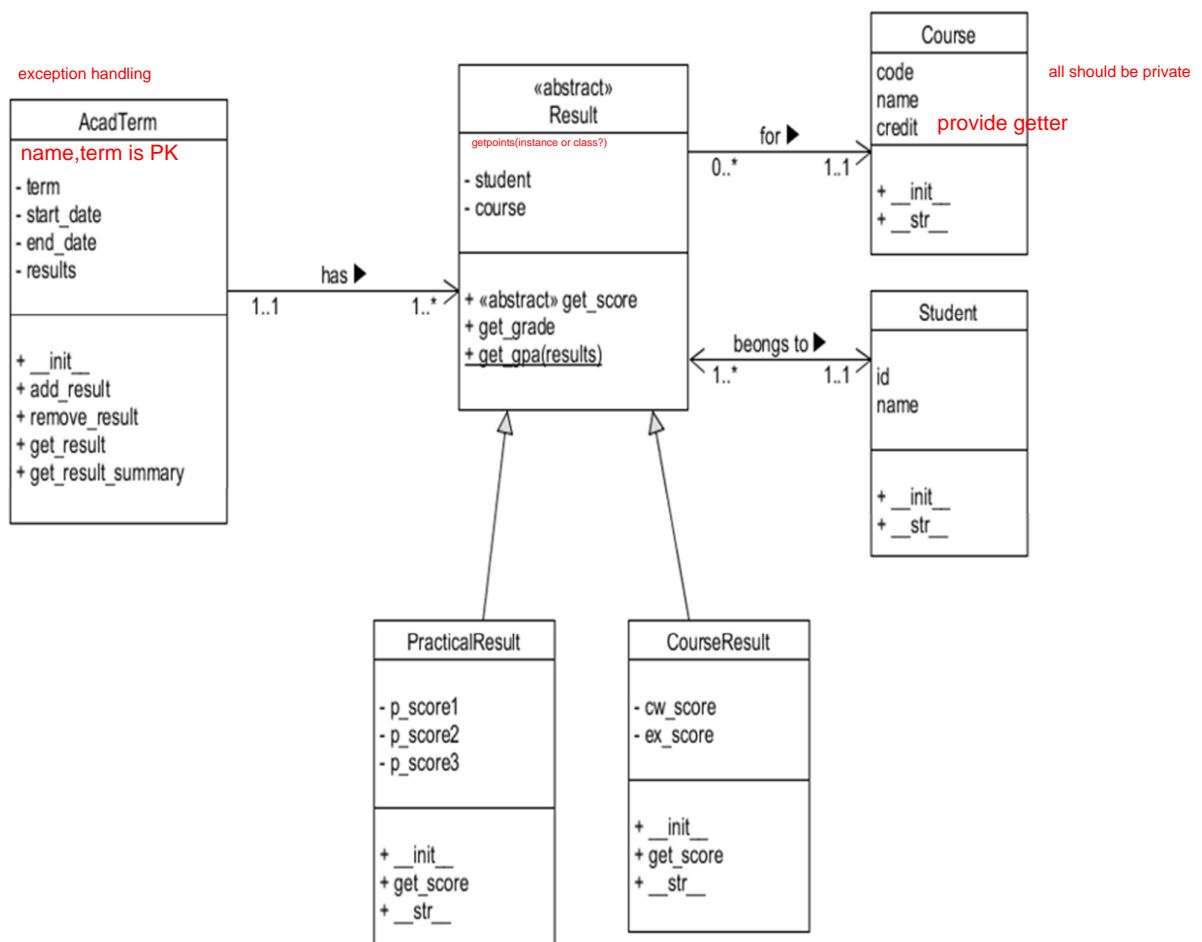
- To apply Object Oriented Design (OOD).
- To apply Object Oriented Programming (OOP) using Python.

### 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: <ul style="list-style-type: none"> <li>• student id and name</li> <li>• course code</li> <li>• overall score</li> <li>• grade</li> </ul>
get_gpa	This is a class method that compute and return the GPA for a collection of <b>results</b> .

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.
<b>Method</b>	
__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: <ul style="list-style-type: none"> <li>• student id and name</li> <li>• course code</li> <li>• p_score1</li> <li>• p_score2</li> <li>• p_score3</li> <li>• overall score</li> <li>• grade</li> </ul>

Class: CourseResult

Attribute	Description
cw_score	The score of the course work.
ex_score	The score of the exam.
<b>Method</b>	
__init__	Constructor.
get_score	This method will compute and return the overall score as follows: $cw\_score \times 0.3 + ex\_score \times 0.7$
__str__	This method will return a string containing the following: <ul style="list-style-type: none"> <li>• student id and name</li> <li>• course code</li> <li>• cw_score</li> <li>• ex_score</li> <li>• overall score</li> <li>• grade</li> </ul>

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 <b>results</b> of the academic term.
<b>Method</b>	

<code>__init__</code>	Constructor.
<code>add_result</code>	This method will add a result (object of Result subclasses) to the academic term. The method must ensure that the <b>same result</b> cannot be added multiple times to the same academic term.
<code>remove_result</code>	Given the student id and course code, this method will remove the matching result from the academic term.
<code>get_result</code>	This method will return all the results for a student (id) or will return the result of a student for a course (if given).
<code>get_result_summary</code>	This method will return the summary result academic term. The summary result include: <ul style="list-style-type: none"> <li>• term</li> <li>• number of passes</li> <li>• number of failures</li> </ul>
<code>__str__</code>	The method will return a string containing the following: <ul style="list-style-type: none"> <li>• term</li> <li>• start_date</li> <li>• end_date</li> </ul>

#### Grading system

Grade	Score	Points awarded
A	$\geq 95$	4.0
A-	$\geq 90$	3.7
B+	$\geq 85$	3.3
B	$\geq 80$	3.0
B-	$\geq 75$	2.7
C+	$\geq 70$	2.3
C	$\geq 65$	2.0
C-	$\geq 60$	1.7
D+	$\geq 55$	1.3
D	$\geq 50$	1.0
D-	$\geq 45$	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	$4.0 \times 6 = 24$
B	6	$3.0 \times 6 = 18$
B	6	$3.0 \times 6 = 18$
	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.