University of Wollongong
School of Computing and Information Technology
CSIT121 Object Oriented Design and Programming
Assignment 1

Objectives

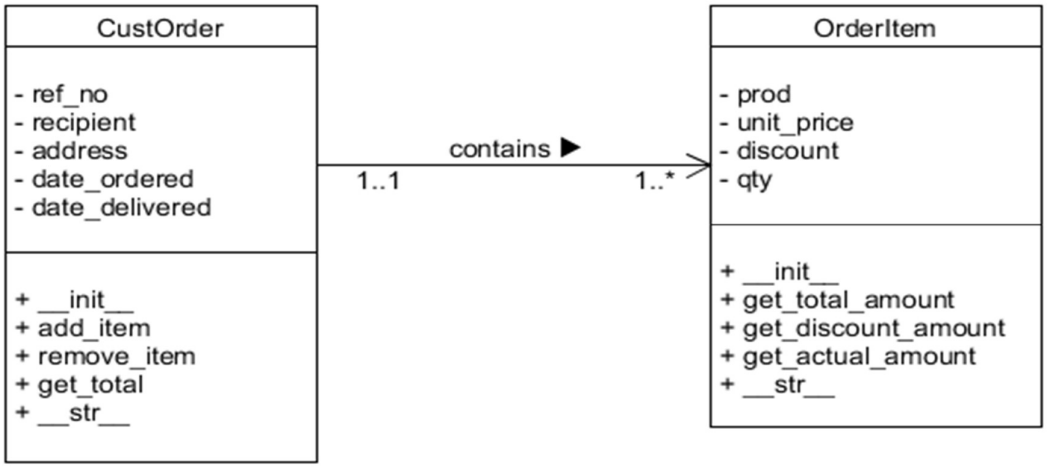
- To apply Object Oriented Design (OOD).
- To apply Object Oriented Programming (OOP) using Python.

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 Order Management Application depicted in the following, ***draft*** class diagram.



Class: CustOrder

Attribute	**Description**
ref_no	A unique, identifying code of each order.
recipient	The name of the recipient of the order.
address	The address to deliver the order.
date_ordered	The date when the customer places the order.
date_delivered	The date when the order is delivered to the recipient
Method	

__init__	The constructor to initialise the attributes.
add_item	This method will add an item to the order. If the item is already contained in the order, it adds to existing item with increase in QTY.
remove_item	The method will remove an item from the order. If the item is not in the order, the method will return False. Otherwise, the item will be removed, and the method will return True.
get_total	The method will compute and return the total amount due which is the amount of all the items contained in the order.
__str__	This method will return a string containing the following: • ref_no. • recipient. • total amount due. • the detail of each OrderItem (see OrderItem class).

Sample __str__() of CustOrder
==========
Order ref: 112233
Name: Ong Siew Teng
Address: 1, Oxley drive, Singapore 545322
Date ordered/delivered: 4-Oct-2024 10:00am / 5-Oct-2024 10:34am

S/N	Product	Price	Qty	SubTotal
1.	Mars	2.00	2	$ 4.00
2.	Maggie Mee	2.80*	1	$ 2.80
			Total	$ 6.80

* - discounted price
==========

Class: OrderItem

Attribute	**Description**
prod	A unique, identifying code of the ordered product.
unit_price	The unit price of the ordered product.
discount	The discount given in percentage. Default is 0.
qty	The quantity ordered.
Method	
__init__	The constructor to initialise the attributes.
get_total_amount	The method will compute and return the total amount of an ordered item: unit_price x qty.

get_discount_amount	The method will compute and return the discount of an ordered item: unit_price x qty x discount.
get_actual_amount	The method will compute and return: total amount – discount amount
__str__	The method will return a string containing the following: • prod. • qty. • actual amount.

You will carry out OOD and OOP as follow:

- You must choose an appropriate data type (class) for each attribute.
- You must include appropriate properties and setters.
- 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 which includes sufficient test cases to demonstrate the functionalities of the program. For instance, in the main function, you may
    - create a CustOrder object;
    - create several OrderItem objects and add them to the CustOrder object;
    - print the OrderItem object to show its content;
    - add "duplicate" OrderItem object to the CustOrder object;
    - print the OrderItem object to verify that duplicate OrderItem QTY has increased;
    - remove OrderItem objects from the CustOrder object;
    - remove non-existing OrderItem objects from the CustOrder object;
    - print the CustOrder object to verify that the removal operations are carried out correctly.