▼ **Different time zone detected (javascript:void(0))**

This course is delivered in the Australia/Melbourne time zone. We've noticed that your user profile is currently set to the Asia/Hong_Kong time zone. Please keep in mind that Canvas will display course dates and times to you as per your current user profile time zone. If you'd like to review or change these settings please do so in **your profile settings. (https://rmit.instructure.com/profile/settings)**

Assignment 2

Start Assignment

Due May 29 by 9:59pm **Points** 30 **Submitting** a file upload **File Types** py
Available Apr 17 at 10pm - Jun 11 at 9:59pm about 2 months

Assignment 2

Key assessment information

Weighting: 30%

Due date: End of Week 12, 29 May 11:59 PM (Melbourne time)

Assessment type: Code submission

Late submission penalty: Deduct 10% of total marks (e.g., 3 marks) per day late unless special consideration has been granted.

The specification of this assignment can be found **here** ↓ **(https://rmit.instructure.com/courses/90501/files/24131595/download?download_frd=1)** **.** Please make sure you read every page of the assignment specification. You must upload your submission for this assessment in the .py format (one file). Note you are required to submit weekly (from Week 9 to 11) in order to get some marks, please refer to the assignment specification for more details. Some sampled txt files used in the assignment can be downloaded from **here** ↓ **(https://rmit.instructure.com/courses/90501/files/24131596/download?download_frd=1)** .

Assessment policies

1. Referencing guidelines

You must acknowledge all the sources of information you have used in your assessments. See details in the assignment specification (Section 5).

2. Academic integrity and plagiarism

Academic integrity is about the honest presentation of your academic work. It means acknowledging the work of others whilst developing your own insights, knowledge, and ideas. If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own. RMIT University treats plagiarism as a very serious offense constituting misconduct. See details in the assignment specification (Section 6).

3. Assessment Declaration

When you submit work electronically, you agree to the assessment declaration **here** **(https://www.rmit.edu.au/students/student-essentials/assessment-and-results/how-to-submit-your-assessments)** .

4. Penalties for Late Submission

All assignments will be marked as if submitted on time. Late submissions of assignments without special consideration or extension will be automatically penalised at a rate of 10% of the total marks available per day (or part of a day) late. For example, if an assignment is worth 30 marks and it is submitted 1 day late, a penalty of 10% or 3 marks will apply. This will be deducted from the assessed mark. Assignments will not be accepted if more than five days late, unless special consideration or an extension of time has been approved.

5. Special Consideration

If you are applying for extensions for your assessment within five working days after the original assessment date or due date has passed, or if you are seeking an extension for more than seven days, you will have to apply for Special Consideration, unless there are special instructions on your Equitable Learning Plan.
In most cases, you can apply for special consideration online **here** **(https://specon.rmit.edu.au/specon/)** . For more information on special consideration, visit the university website on special consideration **here** **(https://www.rmit.edu.au/students/student-essentials/assessment-and-results/special-consideration)** .

Assignment_2_COSC2531

Criteria	Ratings			Pts
PASS Level - The program correctly shows inheritance through Customer, Member, and VIPMember. These classes are defined accurately per the assignment specification.	**2 pts** **Full Marks**	**2 to >1.0 pts** **Good implementation with 1 to 2 minor issues.**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**	2 pts
PASS Level - Correctly define the class Product and class Order, including their variables and methods.	**1 pts** **Full marks**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**		1 pts
PASS Level - Correctly define the class Records, including its variables and methods.	**3 pts** **Full Marks**	**3 to >1.5 pts** **Good implementation with 1 to 2 minor issues.**	**1.5 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**	3 pts
PASS Level - The program can correctly initialize (e.g., load the data from the files, initialize classes, etc.) and shows the menu. It will go back to the menu after completing a task on the menu. The program can exit.	**2 pts** **Full Marks**	**2 to >1.0 pts** **Good implementation with 1 to 2 minor issues.**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**	2 pts
PASS Level - The program can correctly display all customers and products through the menu. This includes adding new customers or updating existing customer information.	**2 pts** **Full Marks**	**2 to >1.0 pts** **Good implementation with 1 to 2 minor issues.**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**	2 pts
PASS Level - Orders can be entered through the menu and the total price of the order can be displayed accordingly (including handling VIP membership).	**2 pts** **Full Marks**	**2 to >1.0 pts** **Good implementation with 1 to 2 minor issues.**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**	2 pts
CREDIT Level ======= No marks for this level if the PASS Level requirements are not completed ======== Class Bundle is correctly defined, including its variables and methods. Class Records are updated properly. Your program is updated accordingly to support Bundle.	**1 pts** **Full Marks**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**		1 pts
CREDIT Level - Appropriate exceptions are defined. The corresponding exception handling mechanism is in place.	**1 pts** **Full Marks**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**		1 pts
CREDIT Level - Required operations are supported by the program, including placing an order, updated products + prices display, handling both name and ID, etc.	**1 pts** **Full Marks**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**		1 pts

Criteria	Ratings		Pts
DI Level ======= No marks for this level if requirements of early levels are not completed ======== Class Order is updated.	**0.5 pts** **Full Marks**	**0 pts** **No Marks**	0.5 pts
DI Level - Orders can be read from a CSV file and correctly populated. The menu program can display all orders or all orders of a customer and handle both name and ID.	**1 pts** **Full Marks**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**	1 pts
DI Level - The discount rate and the threshold for VIP members can be adjusted. Corresponding exceptions are handled properly. Other required exceptions can also be handled.	**1 pts** **Full Marks**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**	1 pts
DI Level - Orders can be entered through the menu and the total price of the order can be displayed accordingly.	**0.5 pts** **Full Marks**	**0 pts** **No Marks**	0.5 pts
HD Level ======= No marks for this level if requirements of early levels are not completed - ======== The class structure is analyzed and optimized.	**0.5 pts** **Full Marks**	**0 pts** **No Marks**	0.5 pts
HD Level – The order file can be specified by the user through the menu. Re-try is permitted.	**0.5 pts** **Full Marks**	**0 pts** **No Marks**	0.5 pts
HD Level - Command-line arguments can be correctly handled as specified in the requirements.	**1 pts** **Full Marks**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**	1 pts
HD Level – The program allows ordering multiple items within an order.	**1 pts** **Full Marks**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**	1 pts
HD Level - Display the most valuable customer and most popular product according to the requirements	**1 pts** **Full Marks**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**	1 pts
HD Level - Display the summary of the orders in full as required.	**1 pts** **Full Marks**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**	1 pts
HD Level - The customer file, product file, and order file used by the program can be correctly updated when the program terminates.	**1 pts** **Full Marks**	**1 to >0 pts** **Poor implementation (e.g. missing components, poor coding, major mistakes)**	1 pts

Criteria	Ratings			Pts
Others - Code quality and style	**1.5 pts** **Full Marks**	**1.5 to >1.0 pts** **Good coding style** With one or two minor issues	**1 to >0 pts** **Need improvement** E.g., insistent style, spaghetti code, poor choice of variable/method names, non-OO style	1.5 pts
Others - Comments, analysis and reflection	**1.5 pts** **Full Marks**	**1.5 to >1.0 pts** **Good coding style** With one or two minor issues	**1 to >0 pts** **Need improvement** E.g., lack of comments, overuse of comments, poor explanation of the OO design, code design.	1.5 pts
Others - Weekly submissions from Week 9 to Week 11, 1 x number of submissions (max is 3, excluding the final submission).	**3 pts** **Full Marks**	**3 to >0 pts** **Need improvement** Missed one or a few submissions.		3 pts
				Total Points: 30