

COMP642 Advanced Programming

Software Project Part 2 – Implementing the Design

Worth:	50%
Due:	Fri 29 October 2021 5:00pm.
Late Penalty:	Work not received by the due time attracts an immediate penalty of 25% of the marks available. No work will be accepted after Tue 2 November 2021 5:00pm.
Submission:	Zip your completed files and submit .zip through the link on the COMP642 LEARN page.

Overview

Create a Python application which uses the model and controller classes designed in Project Part 1 to manage information for a supermarket. Some extensions to the design are required.

Design

The outline of a design for **Customer**, **Item**, **UnitItem**, **WeightItem** and **ShoppingCart** classes and for the **Supermarket** class (which acts as a controller) can be downloaded from COMP642 LEARN Page. You may base your application on this design or on the one you developed yourself in Project Part 1. The provided design does not include the extra functionality required to extend the application.

Requirements

Create a Python application to manage the Supermarket. A file called **customers.text** contains information about customers and can be downloaded from COMP642 LEARN page.

1. Adapt as necessary your design (or the provided design) from Project Part 1 so that:
 - When a customer starts shopping, the cart is empty (currentCart).
 - A customer can place multiple items in a shopping cart.
 - When a customer completes the checkout process with their cart:
 - i. the total amount of the shopping cart will be displayed,
 - ii. the currentCart should be added to the list of shopping carts for the customer to keep track of previous purchases,
 - iii. the club points earned for the current transaction should be calculated and added to the club point balance.
 - It provides the following information for a customer:
 - i. Name, card number, total club points, and grand total of all purchases

- ii. A list of all transactions (shopping carts)
- iii. A list of all items purchased in each transaction

```
John Gray 10001 19 $195.61
29/09/2021 $30.84
Milk $5.99 1 $5.99
Sugar $4.99 2 $9.98
Chicken $15.99 0.74 $11.88
Carrot $2.99 1 $2.99
29/09/2021 $164.77
Beef $16.99 3.36 $57.00
Salmon $32.99 2.93 $96.79
Bread $5.49 2 $10.98
```

- It provides the following information for the supermarket
 - i. A list of all customers showing, name, card number and total spent
 - ii. A list of all transactions for each customer

```
John Gray 10001 $175.48
29/09/2021 $24.95
29/09/2021 $42.74
29/09/2021 $107.79

Louise Smith 10002 $99.25
29/09/2021 $30.97
29/09/2021 $68.27
```

- It provides the total sales to date for the supermarket, the average cart value and the customer with the highest total value of purchases.
2. Write and test the code for all the classes in your design using **pytest**.
 3. Build an appropriately designed interface (an example is given below) with appropriate controls, useful feedback and exception handling.
 4. Your code must be clear and easy to maintain and appropriately commented.
 5. Your application **does not** have to provide facilities for the user to add new customers, delete existing objects and to amend details of customers or items.

Possible Interface Design

The screenshot shows a window titled "Supermarket Application" with a subtitle "Lincoln Supermarket". It is divided into three main sections: "Customer Detail", "Transaction", and "Summary".

- Customer Detail:** Contains a "Customer Name" dropdown menu with "John Gray" selected, a "Card Number" text field with "10001", and four buttons: "Start Shopping" (highlighted with a blue border), "Next Customer", "Exit", and "Customer Information".
- Transaction:** Contains a "Product Name" text field, an "Item Type" section with two radio buttons ("Unit Item" and "Weight Item"), a large empty rectangular area for item details, and three buttons: "Add to Cart", "New Item", and "Checkout". Below these is a "Total Cost" label and an empty text field.
- Summary:** Contains four buttons: "Sales by Customer", "Total Sales", "Top Customer", and "Average Cart".

When a customer is selected and the button Start Shopping is clicked

This screenshot shows the same application after a unit item has been added. The "Transaction" section now contains more data:

- Product Name:** The text field now contains "Cereal".
- Item Type:** The "Unit Item" radio button is now selected.
- Unit Item Section:** A new section appears with "Number of Units" set to "1" and "Price per Unit" set to "10.99".
- Item Details Area:** The previously empty area now displays "Bread \$10.98".
- Buttons:** The "Add to Cart", "New Item", and "Checkout" buttons remain.
- Total Cost:** The text field remains empty.

The "Customer Detail" and "Summary" sections remain unchanged from the previous state.

Adding a Unit Item to the customer's cart

Supermarket Application

Lincoln Supermarket

Customer Detail

Customer Name: Card Number:

Transaction

Product Name:

Item Type

☐ Unit Item

☒ Weight Item

Weight Item

Price per Kilo:

Weight:

Total Cost:

Summary

Adding Weight Item to the customer's cart

Supermarket Application

Lincoln Supermarket

Customer Detail

Customer Name: Card Number:

Transaction

Product Name:

Item Type

☐ Unit Item

☒ Weight Item

Weight Item

Price per Kilo:

Weight:

Total Cost:

Summary

When Checkout button is clicked

Extending the application

Extend your application to allow for reporting of sales for a particular month of the current year. You will need to add functionality to your Supermarket class controller and to adapt your view (form). You should **not** need to change your model classes.

- The user must be allowed to choose or specify one particular month. To minimise errors, you should not use a text box. You could use ListBox (with options for Jan, Feb, Mar,... or 01, 02, 03, ...) or some radio buttons or other suitable interface component for this selection.
- Your application should allow the user to view the details of the sales (customer name, card number and total spend) for the chosen month. Include the total sales for the supermarket that month.

```
Sales for Month: 02

John Gray 10001 $151.90
Louise Smith 10002 $69.93
Henry Waite 10003 $28.96
Harry Liu 10004 $123.47
Joe Blogg 10005 $130.05
Jaime Watson 10008 $86.05

Total: $590.35
```

Marking

Criteria	Marks (out of 100)
Application functionalities <ul style="list-style-type: none">• Runs with no errors• Selecting customer• Adding items to cart• Checkout• Reporting• Input validation and feedback	50
Code <ul style="list-style-type: none">• Model classes• Controller• View• Error handling and preventions• Testing for all model classes	50