# SCHOOL OF COMPUTING
## Faculty of Engineering

**FINAL EXAMINATION**
**SEMESTER 2, SESSION 2019 / 2020**
**PART B (PRACTICAL)**

| | | |
|---|---|---|
| **SUBJECT CODE** | : | **SCSJ 2154** |
| **SUBJECT NAME** | : | **OBJECT ORIENTED PROGRAMMING** |
| **SECTION** | : | **2 (SCSJ / SCSV / SCSB / SCSR)** |
| **DATE/DAY** | : | **11th JULY 2020 (SATURDAY)** |
| **TIME** | : | **10.00 AM - 12.45 PM** |

_____

## INSTRUCTIONS:

- You are given 2 HOURS 45 MINUTES to complete the exam inclusive the submission of your answers.
  - ✓ Download the question: 10.00 – 10.15 am (15 minutes)
  - ✓ Answer the question & Interim submission: 10.15 – 12.30 pm (2 hours 15 minutes)
  - ✓ Final answer submission: 12.30 – 12.45 pm (15 minutes)
- A candidate who is suspected of cheating in examinations is liable to disciplinary action including (but not limited to) suspension or expulsion from the University. All materials and or devices which are found in violation of any examination rules and regulation will be confiscated.

## IMPORTANT NOTES:

- All the **COMMENT STATEMENTS** in the submitted program **WILL NOT BE EVALUATED**.

## SUBMISSION PROCEDURE:

- Compress all files using **\*.zip format**. Only the compress file is required for the submission and the file shall be named as follows: ***InterimSCSJ2154_Name_ matricsNo_section.zip*** (for interim submission) and ***FinalAnsSCSJ2154_Name_ matricsNo_section.zip*** (for final submission).

| Name | |
|---|---|
| **I/C No.** | |
| **Year / Course** | |
| **Section** | |
| **Lecturer Name** | |

*This question booklet consists of 9 pages inclusive of the cover page.*

# PROBLEM SOLVING                                    (100 Marks)

Write a complete Java program based on the UML class diagram given in Figure 1. Your program should be able to produce the output shown in Figure 2.
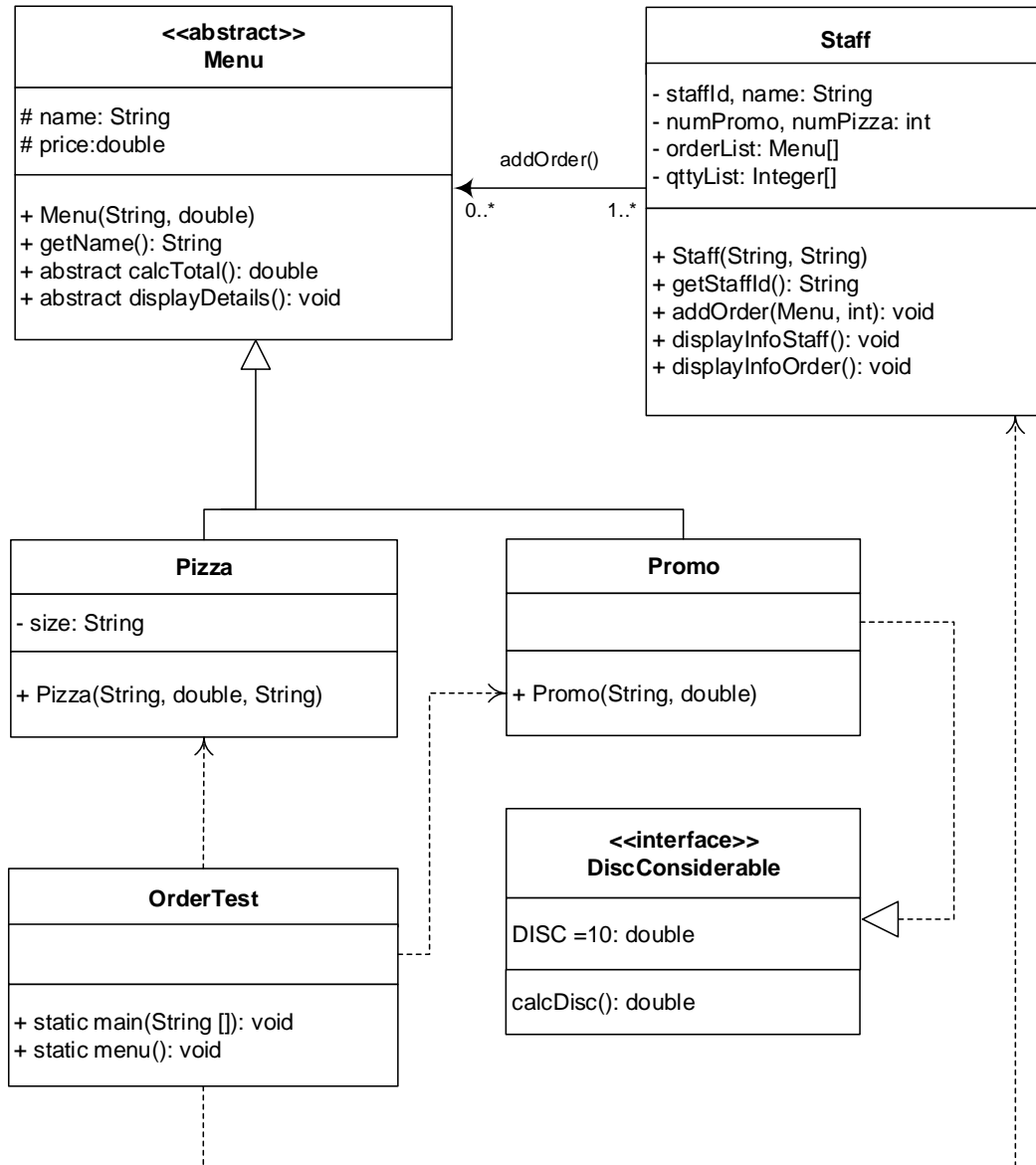


**Figure 1:** UML class diagram

You must write six complete classes, **Menu**, **Staff**, **Promo**, **Pizza**, **DiscConsiderable** and **OrderTest** based on the instruction given below:

(a) Write a class named **Staff**. The class provides the following instance/ member variables and methods:                                    (22.5 Marks)

(i) The class should have six (6) **private** member variables (attributes) (refer to Figure 1). Two of the member variables are the array of **Menu** objects named **orderList** and the array of **Integer** objects named **qttyList**. *Note:* You are allowed to use dynamic array (**ArrayList** or **Vector**) or static array for **orderList** and **qttyList**. (2 marks)

(ii) Write the codes for a constructor with arguments that accepts the staff's id and name. This value should be assigned to the staff's **staffId** and **name** member variables. The constructor also enables to assign zero (0) to the **numPromo** and **numPizza** member variables, and creates **orderList** (the array of **Menu** to hold objects of **Menu**) and **qttyList** (the array of integer). (2.5 marks)

(iii) Write the codes for **getStaffId** method that returns the value of **staffId** member variable. (1 mark)

(iv) Write the codes for **addOrder** method that will receive two arguments of **Menu** instance and an integer value, update the array named **orderList** and **qttyList** to include the new instance of **Menu** and the new integer value in the arrays and update the number of promo and pizza menus. (4.5 marks)

(v) Write the codes for **displayInfoStaff** method to display the staff's id and name of **Staff**'s object in the following format (please refer to the sample output given in **Figure 2**): (1.5 marks)

```
Staff Id  : ?????
Staff Name: ????? ?????
```

(vi) Write the codes for **displayInfoOrder** method to display the number of promo and pizza menus of **Menu**'s object, the number of all menus, the list of menus and total of sales (if any) in the following format (please refer to the sample output given in **Figure 2**): (10 marks)

```
Number of All Orders  : ##
Number of Promo Orders: ##
Number of Pizza Orders: ##

No  Description      Unit Price(RM)  Discount(RM)  Quantity
Total(RM)
1.  ????? ?????          ##.##          #.##          #         ##.##
2.  ????? ?????          ##.##          #.##          #         ##.##
```

```
3.   :    :              :              :            :            :
Total sales = RM##.##
```

Please display an appropriate message if there is no order, i.e. "No order!". *Note:* You may need to invoke an appropriate method(s) defined in this class or other class to display the list of orders.

(b) Write an abstract superclass named **Menu**. The class provides a constructor with arguments that will initialize all the member attributes to the values received as arguments. It also provides **getName** method that returns the value of **name** member variable. (5 Marks)

(c) Write a subclass **Promo** and implements interface with the following codes: (8 Marks)

(i) Write the codes for a constructor with arguments that will initialize all the superclass's attributes. (1.5 marks)

(ii) Write the codes for **calcDisc** method that will return discounted price. The discount of promo menu is 10% (please refer to the interface). (1.5 marks)

(iii) Write the codes for **calcTotal** method that will return the price after discount. (1.5 marks)

(iv) Write the codes for **displayDetails** method that will display the menu's name, price, and discounted price in the following format (please refer to the sample output given in **Figure 2**): (2 marks)
```
????? ?????              ##.##          #.##
```

(d) Write a subclass **Pizza** with the following codes: (6 Marks)

(i) Write the codes for a constructor with arguments that will initialize all the member attributes for the class, including the superclass's attributes. (2 marks)

(ii) Write the codes for **calcTotal** method that will return the menu's price. (1 marks)

(iii) Write the codes for **displayDetails** method that will display the menu's name, prece and whitespace in the following format (please refer to the sample output given in **Figure 2**): (1.5 marks)

```
????? ?????              ##.##            (whitespace)
```

(e) Write an interface named **DiscConsiderable** with the following codes: (1.5 Marks)

(i) Declaration of one final static variable.

(ii) Declaration of one abstract method.

(f) Write a driver class named **OrderTest**. The class provides the following methods:

(49.5 Marks)

(i) Write the codes for **menu** method that will provide the user a menu-driven interaction as follows (please refer to the sample output given in **Figure 2**): (2 marks)

```
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
```

(ii) Write the codes for **main** method that will produce the output like the sample output given in **Figure 2** according to the following descriptions: (47 marks)

- Identify the suitable variables and/ or instances where appropriate.

- Create a static or dynamic array of **Staff** and **Menu** objects.

- Create a **Scanner** object for input purpose.

- Ask the user to enter the chosen task. Note that, all the interactions shown in **Figure 2** are continuous in a single run. Note also that, the **bold** texts indicate input entered by the user.

*[1] Add Staff*                                                    *[4 marks]*

- Ask the user to enter the staff's id and name.

- Create a new **Staff** object.

- Insert a new staff object into the **Staff** array.

**Note:** You are allowed to use dynamic array (**ArrayList** or **Vector**) or static array for the **Staff** array.

*[2] Add Menu* *[9.5 marks]*

- Ask the user to enter the menu's type, name and price. If it is a pizza menu, ask the user to enter the pizza size and update the name of pizza by including the pizza size.
- Create a new menu (a promo or pizza) object. *Hint:* Use a polymorphic concept.
- Insert a new menu object into the **Menu** array.

**Note:** You are allowed to use dynamic array (**ArrayList** or **Vector**) or static array for the **Menu** array.

*[3] Add Order* *[15 marks]*

- Ask the user to enter the selected staff.
- Ask the user to enter the selected menu and quantity required.
- Insert a new order object into the **Menu** array for the selected staff.
- The process of a new order insertion will be continued until the user enters invalid input.
- If there is no staff or menu, display an appropriate message, i.e. "Sorry!! No staff or menu, please add staff or menu first...".

*[4] Display Orders* *[6.5 marks]*

- Display the information of all the orders based on staff if any.
- If there is no staff, display an appropriate message, i.e. "Sorry!! No staff, please add staff first...".

*[5] Exit* *[0.5 mark]*

- End the program by displaying an appropriate message, i.e. " Thank you! :)".

(g) Using an appropriate structure for the program such as using a proper output formatting, indentation and Java statements. (3 Marks)

(h) The program is able to run, work, and display the output as required. (4 Marks)

```
========== Menu ==========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
==========================

Select task: 3
Sorry!! No staff or menu, please add staff or menu first...

========== Menu ==========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
==========================

Select task: 4
Sorry!! No staff, please add staff first...

========== Menu ==========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
==========================

Select task: 1

<<< Add Staff >>>
Enter Staff Id   : S1234
Enter Staff Name : Aishah Aiman

========== Menu ==========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
==========================

Select task: 2

<<< Add Menu >>>
Type (Promo/ Pizza) : Promo
Name : Large Double Box
Price: RM82.70

========== Menu ==========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
==========================

Select task: 2

<<< Add Menu >>>
Type (Promo/ Pizza) : Promo
Name : Raikan Detik Lengkap Sekeluarga
Price: RM105.30

========== Menu ==========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
==========================

Select task: 2

<<< Add Menu >>>
Type (Promo/ Pizza) : Promo
Name : Peneman Selera Jauh di Sana
Price: RM24.15
```

```
========== Menu ==========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
==========================

Select task: 2

<<< Add Menu >>>
Type (Promo/ Pizza) : Pizza
Name : Aloha Chicken
Price: RM37.90
Size [S/M/L]: L

========== Menu ==========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
==========================

Select task: 2

<<< Add Menu >>>
Type (Promo/ Pizza) : Pizza
Name : Beef Pepperoni
Price: RM27.90
Size [S/M/L]: M

========== Menu ==========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
==========================

Select task: 3

<<< Add Order(s) >>>

Staff List
1) S1234

Select staff: 1

--- Enter Order ---
1) Large Double Box
2) Raikan Detik Lengkap Sekeluarga
3) Peneman Selera Jauh di Sana
4) Aloha Chicken (L)
5) Beef Pepperoni (M)

Select menu: 1
Quantity: 2

Press 'Y' to continue >> n

========== Menu ==========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
==========================

Select task: 4

<<< Staff(s) and Order(s) Info >>>

Number of Staffs: 1

Staff #1
Staff Id   : S1234
Staff Name: Aishah Aiman

Number of All Orders   : 1
Number of Promo Orders: 1
Number of Pizza Orders: 0
```

```
No   Description                              Unit Price(RM)  Discount(RM)  Quantity  Total(RM)
1.   Large Double Box                             82.70          8.27          2       148.86

Total sales = RM148.86

========= Menu =========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
=========================

Select task: 1

<<< Add Staff >>>
Enter Staff Id   : S1111
Enter Staff Name : Imran Rashid

========= Menu =========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
=========================

Select task: 3

<<< Add Order(s) >>>

Staff List
1) S1234
2) S1111

Select staff: 2

--- Enter Order ---
1) Large Double Box
2) Raikan Detik Lengkap Sekeluarga
3) Peneman Selera Jauh di Sana
4) Aloha Chicken (L)
5) Beef Pepperoni (M)

Select menu: 5
Quantity: 3

Press 'Y' to continue >> y

--- Enter Order ---
1) Large Double Box
2) Raikan Detik Lengkap Sekeluarga
3) Peneman Selera Jauh di Sana
4) Aloha Chicken (L)
5) Beef Pepperoni (M)

Select menu: 2
Quantity: 1

Press 'Y' to continue >> y


--- Enter Order ---
1) Large Double Box
2) Raikan Detik Lengkap Sekeluarga
3) Peneman Selera Jauh di Sana
4) Aloha Chicken (L)
5) Beef Pepperoni (M)

Select menu: 1
Quantity: 1

Press 'Y' to continue >> n

========= Menu =========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
=========================
```

```
Select task: 3

<<< Add Order(s) >>>

Staff List
1) S1234
2) S1111

Select staff: 1

--- Enter Order ---
1) Large Double Box
2) Raikan Detik Lengkap Sekeluarga
3) Peneman Selera Jauh di Sana
4) Aloha Chicken (L)
5) Beef Pepperoni (M)

Select menu: 5
Quantity: 4

Press 'Y' to continue >> n

========== Menu ==========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
==========================

Select task: 4

<<< Staff(s) and Order(s) Info >>>

Number of Staffs: 2

Staff #1
Staff Id  : S1234
Staff Name: Aishah Aiman

Number of All Orders  : 3
Number of Promo Orders: 2
Number of Pizza Orders: 1

No  Description                       Unit Price(RM)  Discount(RM)  Quantity  Total(RM)
1.  Large Double Box                       82.70          8.27         2        148.86
2.  Beef Pepperoni (M)                     27.90                       4        111.60

Total sales = RM260.46

Staff #2
Staff Id  : S1111
Staff Name: Imran Rashid

Number of All Orders  : 3
Number of Promo Orders: 2
Number of Pizza Orders: 1

No  Description                       Unit Price(RM)  Discount(RM)  Quantity  Total(RM)
1.  Beef Pepperoni (M)                     27.90                       3         83.70
2.  Raikan Detik Lengkap Sekeluarga       105.30         10.53         1         94.77
3.  Large Double Box                       82.70          8.27         1         74.43

Total sales = RM252.90

========== Menu ==========
[1] Add Staff
[2] Add Menu
[3] Add Order
[4] Display Orders
[5] Exit
==========================

Select task: 5
Thank you! :)
```

**Figure 2:** Expected output of the program