



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

**SCHOOL OF COMPUTING**  
Faculty of Engineering

## UNIVERSITI TEKNOLOGI MALAYSIA

### TEST 2

### PROBLEM-SOLVING

### SEMESTER II 2020/2021

**SUBJECT CODE : SECJ2154**  
**SUBJECT NAME : OBJECT ORIENTED PROGRAMMING**  
**YEAR/COURSE : 2 (SECB/ SECJ/ SECP/ SECR/ SECV)**  
**TIME : 10:00 – 12:30 MYT (UTC +8, 150 Minutes)**  
**DATE/ DAY : 5<sup>th</sup> JUNE 2021 (SATURDAY)**

---

#### INSTRUCTIONS TO THE STUDENTS:

- Read the problem and instructions carefully.
- You are given **TWO HOURS THIRTY MINUTES** to complete the test, inclusive of your program's submission (interim and final submissions).

#### IMPORTANT NOTES:

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

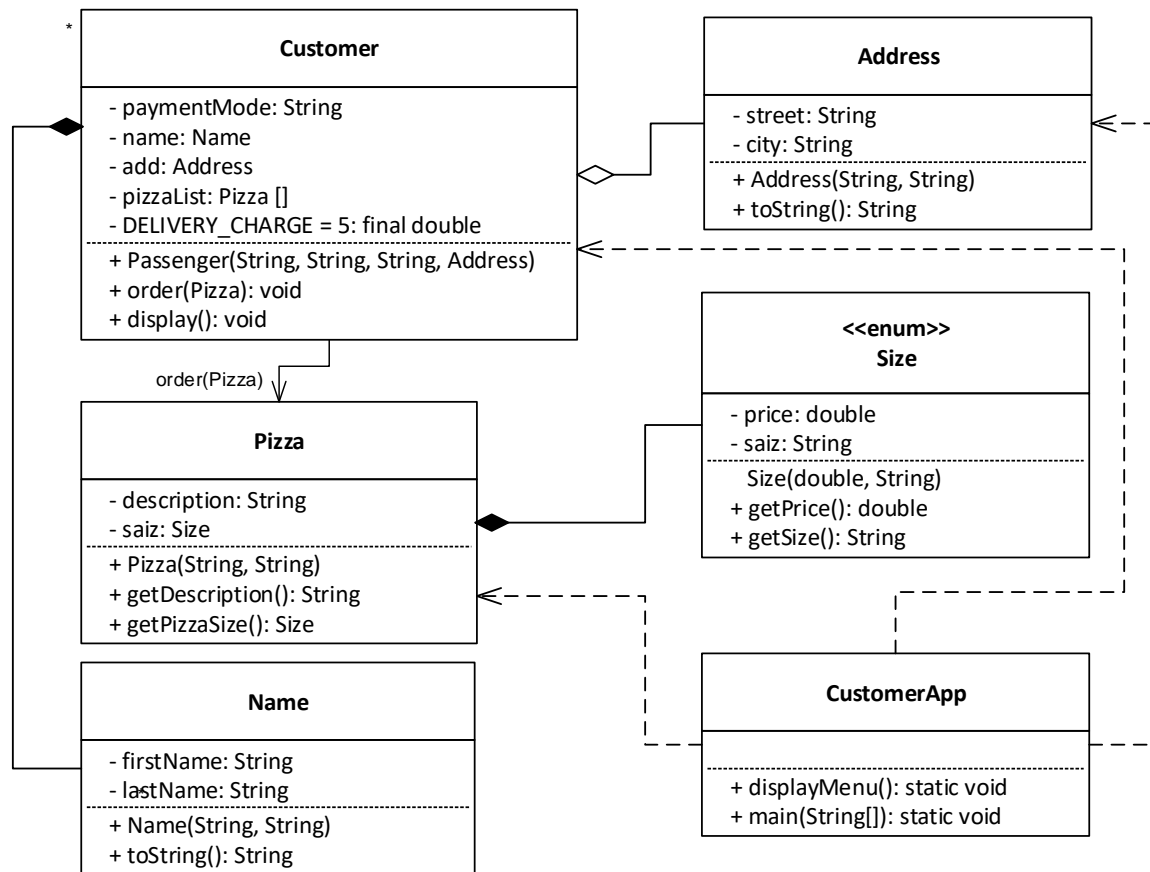
#### SUBMISSION PROCEDURE:

- Only the source codes' file (\*.java) is required for the submission.
- You do not need to compress the file.
- Submit the source code file via the **UTM's e-learning system**.

## PROBLEM-SOLVING

(100 Marks)

Write **six (6)** complete Java programs named, **Size.java**, **Pizza.java**, **Address.java**, **Name.java**, **Customer.java**, and **CustomerApp.java** 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

Implement all the classes with the instance variables (attributes) and methods specified in the diagram. The purpose of each method is as the name implies, and some of them are further explained below. Write the program based on the following tasks:

### Task 1:

(5 Marks)

In **Address** class, do the following tasks:

- Define all the attributes of the class. (1 mark)
- Define a constructor with arguments. Initialize all the instance variables for the class with the passed arguments. (2 marks)
- Define `toString` method. The method returns the address of the customer. (1.5 marks)

**Task 2:** (5 Marks)

In **Name** class, do the following tasks:

- a) Define all the attributes of the class. (1 mark)
- b) Define a constructor with arguments. Initialize all the instance variables for the class with the passed arguments. (2 marks)
- c) Define **toString** method. The method returns the full name of the customer.

(1.5 marks)

**Task 3:** (7 Marks)

In **Size** class, do the following tasks:

- a) Write the class uses **enum** data type. The **enum** class has a fixed set of constants. Define the **enum** data type based on **all constants** listed in Table 1. (1.5 marks)

**Table 1:** Set of constants and values for **Size enum** class

Constants	Size	Price (RM)
S	Small	10.00
R	Regular	15.00
L	Large	25.00

- b) Define all the attributes of the class. (1 mark)
- c) Define constructor with arguments. Initialize all the attributes for the class with the passed arguments. (2 marks)
- d) Define the accessor (getter) methods. (2 marks)

**Task 4:** (7 Marks)

In **Pizza** class, do the following tasks:

- a) Define all the attributes of the class. (1 mark)
- b) Define a constructor with arguments. Initialize **description** attribute with a passed **String** argument. Then, create **Size enum** from a passed **String** argument.

(3.5 marks)

- c) Define the accessor (getter) methods. (2 marks)

**Task 5:** (30 Marks)

In **Customer** class, do the following tasks:

- a) Define all the attributes of the class. **Note:** You need to use **Vector** (dynamic array) to declare **pizzaList** attribute. (3.5 marks)

- b) Define constructor with arguments. Initialize **add** attribute with a passed **Address** argument. Create the instance of **Name** and initialize all the attributes of the instance with passed arguments. Finally, create the object from the class **Vector** that is defined in Task 5(a). (5.5 marks)
- c) Define **order** method. The method is used to add the object from **Pizza** class to **pizzaList** array. The object added to the array refers to the pizza ordered by a customer. (1.5 marks)
- d) Define **display** method. The method displays the customer name and address, the number of ordered pizza, the list of pizzas ordered, the total price, the price after discount, and the total charge. The charge for all pizza based on the following criteria:
- If the mode of payment is **Online**, the customer will get 10% discount, while no discount being given for the Cash On Delivery (**COD**) payment.
  - The delivery charge for the pizza is RM5.00
- The list of pizzas ordered must contain the following information: pizza's size, description, and price. **Figure 2** shows the sample of the output that your program should produce. **Note:** You must use proper output formatting to generate the output. Note also that the **bold** texts indicate input entered by the user. (19 marks)

```

-----
      Pizza Delivery Ordering System
-----
[1] Add Customer
[2] Display Customers
[3] Exit
Your choice: 1

***** Add Customer *****
First name: Arshad
Last name: Muhammad Ali
Address:
    Street: 5, Jalan Kota 7, Taman Kota
    City: Kota Tinggi
Payment Mode [Online | COD]: COD
Number of pizzas: 3
    Pizza #1:
        Description: Aloha Chicken
        Size [S-Small, R-Regular, L-Large]: L
    Pizza #2:
        Description: Beef Pepperoni
        Size [S-Small, R-Regular, L-Large]: 1
    Pizza #3:
        Description: Ocean Delight
        Size [S-Small, R-Regular, L-Large]: S

-----
      Pizza Delivery Ordering System
-----

```

```

[1] Add Customer
[2] Display Customers
[3] Exit
Your choice: 1

***** Add Customer *****
First name: Ramlah
Last name: Ayob
Address:
    Street: 21, Jalan Bandar 3, Taman Bandar
    City: Johor Bahru
Payment Mode [Online | COD]: Online
Number of pizzas: 2
    Pizza #1:
        Description: BBQ Chicken
        Size [S-Small, R-Regular, L-Large]: R
    Pizza #2:
        Description: Seafood Deluxe
        Size [S-Small, R-Regular, L-Large]: L

-----
                Pizza Delivery Ordering System
-----

[1] Add Customer
[2] Display Customers
[3] Exit
Your choice: 2

***** List of Customers *****

Customer #1
Name: Arshad Muhammad Ali
Address: 5, Jalan Kota 7, Taman Kota, Kota Tinggi
Number of ordered pizza: 3
1) Aloha Chicken           Large           RM 25.00
2) Beef Pepperoni          Large           RM 25.00
3) Ocean Delight           Small           RM 10.00

Total price: RM 60.00
Price after discount: RM 60.00
Total charge: RM 65.00

Customer #2
Name: Ramlah Ayob
Address: 21, Jalan Bandar 3, Taman Bandar, Johor Bahru
Number of ordered pizza: 2
1) BBQ Chicken             Regular        RM 15.00
2) Seafood Deluxe          Large          RM 25.00

Total price: RM 40.00
Price after discount: RM 36.00
Total charge: RM 41.00

-----
                Pizza Delivery Ordering System
-----

[1] Add Customer
[2] Display Customers
[3] Exit
Your choice: 3

Thank you for using this system :)

```

**Figure 2:** Example output of a program

**Task 6:**

(40 Marks)

In **CustomerApp** class, do the following tasks:

- a) Define **displayMenu** method to provide the user a menu-driven interaction. The definition for the **displayMenu** method is fully given in **Figure 3**.

```
1 public static void displayMenu() {  
2     System.out.println("-----");  
3     System.out.println("        Pizza Delivery Ordering System");  
4     System.out.println("-----");  
5     System.out.println("[1] Add Customer");  
6     System.out.println("[2] Display Customers");  
7     System.out.println("[3] Exit");  
8     System.out.print("Your choice: ");  
9 }
```

**Figure 3: displayMenu function**

- b) Define **main** method with following tasks:

- Create a **Vector** of objects from class **Customer** to store the values for **Customer** attributes. (2 marks)
- Define objects from class **Address**, **Customer**, and **Pizza** and define any suitable variables for the program. You may also need to import a class if necessary. (4 marks)
- Implement **displayMenu** to ask the user to key-in choices. (7 marks)
  - i) If the choice is 1:
    - Display messages for choice 1 as shown in **Figure 2** and read values from the keyboard. Use the values to initialize the object from class **Address**, **Customer**, and **Pizza**.
    - Get the number of pizzas that the customer wants to order and use a loop to get the description and size for the pizza, and implement **order** method for each pizza. (20 marks)
  - ii) If the choice is 2:
    - Display messages for choice 2 and as shown in **Figure 2**. Execute **display** method for each customer in the **Vector**. (4.5 marks)
  - iii) If the choice is 3:
    - Exit from the system and display an appropriate message. (0.5 marks)

**Task 7:****(6 Marks)**

- a) Use an appropriate structure for the program:
  - Import an appropriate class. (1 mark)
  - Use proper output formatting. (1 mark)
  - The code is indented correctly. (1 mark)
- b) The program is able to run, work, and display the output as required. (3 marks)