CONFIDENTIAL

SCHOOL OF COMPUTING
Faculty of Engineering

UNIVERSITI TEKNOLOGI MALAYSIA

TEST 2

PROBLEM SOLVING

SEMESTER II 2020/2021

SUBJECT CODE	**: SECJ1023**
SUBJECT NAME	**: PROGRAMMING TECHNIQUE II**
YEAR/COURSE	**: 1 (SECBH/ SECJH/ SECPH/ SECRH/ SECVH)**
TIME	**: 17:00 – 18:30 MYT (UTC +8, 90 Minutes)**
DATE/ DAY	**: 25th MAY 2021 (TUESDAY)**

INSTRUCTIONS TO THE STUDENTS:

- Read the problem and instructions carefully.
- You are given **ONE HOUR 30 MINUTES** to complete the test inclusive the submission of your program.

IMPORTANT NOTES:

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

SUBMISSION PROCEDURE:

- Only the source code is required for the submission and the source code's file shall be named as follows: ***Name_ matricsNo_section.cpp*** (i.e. *AinaAli_A19EC018_01.cpp*).
- You do not need to compress the file.
- Submit the source code file via the **UTM's e-learning system**.

PROBLEM SOLVING (70 Marks)

Write a complete C++ program based on the UML class diagram given in **Figure 1**. Your program should be able to produce the output shown in **Figure 3**.

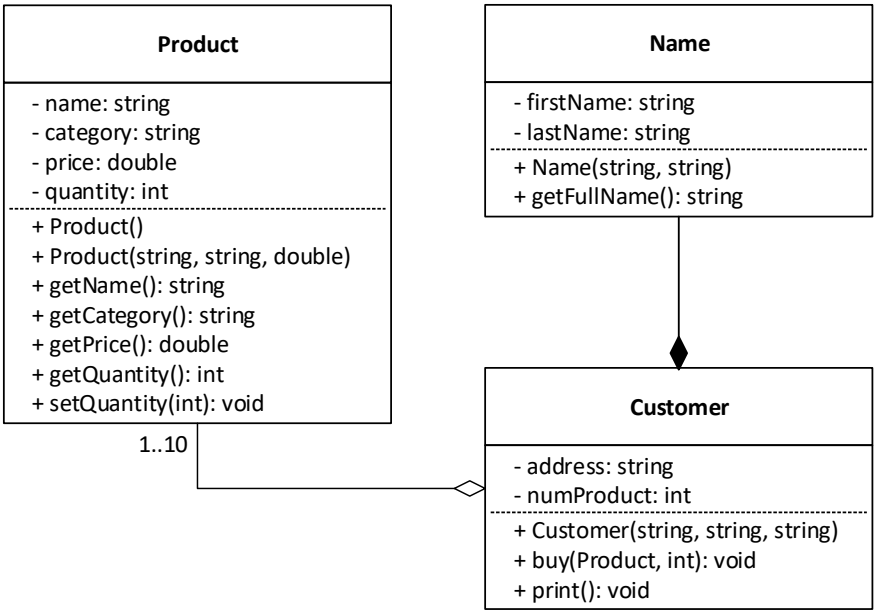

Figure 1: UML class diagram

Implement all the classes with the member variables (attributes) and member functions (methods) specified in the diagram. The purpose of each function is as the name implies, and some of them are further explained below. Note that the definition for the **`main`** function is fully given (refer to **Figure 2**). ***IMPORTANT NOTE:*** **Do not modify the `main` function** given. Write the program based on the following tasks:

```
1   int main () {
2         Customer cust("Amir", "Jalil", "Masai, Johor");
3         Product p1("Jacob", "Biscuit", 14.8);
4         Product p2("Twister", "Drink", 7.5);
5         Product p3("Ayamas", "Nugget", 18.4);
6         Product p4("Oreo", "Biscuit", 3.8);
7
8         cust.buy(p4, 5);
9         cust.buy(p2, 4);
10        cust.buy(p3, 2);
11        cust.buy(p1, 3);
12        cust.print();
13
14        return 0;
15  }
```

Figure 2: `main` function

Task 1: (9 Marks)

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

a) Define all the member variables of the class. (1 mark)

b) Define constructor with arguments. Initialize all the member variables for the class with the passed arguments. (3.5 marks)

c) Define **`getFullName`** function. The function returns the full name. (3.5 marks)

Task 2: (18.5 Marks)

In **`Product`** class, do the following tasks:

a) Define all the member variables of the class. (2 marks)

b) Define default constructor. Initialize all the string member variables with an empty string and numerical member variables with zero. (2.5 marks)

c) Define constructor with arguments. Initialize all the member variables for the class except **`quantity`** with the passed arguments. (5 marks)

d) Define the accessor functions. (6 marks)

e) Define **`setQuantity`** function. The function assigns **`quantity`** member variable with a passed argument. (2 marks)

Task 3: (35.5 Marks)

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

a) Define all the member variables of the class. (2 marks)

b) Define constructor with arguments. Initialize all the member variables for the **`Name`** class and **`address`** member variable with passed arguments. Initialize **`numProduct`** member variable with zero. Then, assign **`Product`** pointer to an array of **`Product`** objects which dynamically allocated. (7 marks)

c) Define **`buy`** function. Suppose the number of products purchased does not exceed the maximum limit. In that case, the function assigns **`quantity`** member variable for **`Product`** object using a function defined in **`Product`** class. Then, it adds or inserts the **`Product`** object (the passed argument) into the array of Product objects defined in (b) (the array that pointed by the **`Product`** pointer in (b)). It also updates the number of products. Otherwise, this function will display an error message like *"Sorry!! You already reached the maximum number of products purchased."*. (7.5 marks)

d) Define **`print`** function. The function displays the name and address of the customer, the number and list of products purchased, and the total price of the products purchased. The list of products purchased must contain the following information: product name and category, a quantity of each product purchased, unit price, and price for the quantity purchased (amount). **Figure 3** shows a sample screen output that your program should produce. **Note:** You must use proper output formatting to generate the output, as shown in **Figure 3**. (18 marks)

```
Name   : Amir Jalil
Address: Masai, Johor
Number of products purchased: 4

No   Product Name   Category   Quantity  Unit Price(RM)  Amount(RM)
1    Oreo           Biscuit        5          3.80          19.00
2    Twister        Drink          4          7.50          30.00
3    Ayamas         Nugget         2         18.40          36.80
4    Jacob          Biscuit        3         14.80          44.40

Total price = RM 130.20
```

Figure 3: Example run of a program

Task 4: (7 Marks)

a) Use an appropriate structure for the program:
   - Use proper preprocessor directives. (1 mark)
   - Use proper output formatting. (2 marks)
   - The code is indented correctly. (1 mark)

b) The program is able to run, work, and display the output as required. (3 marks)