Lab 3

Object Oriented Programming

Instructions:

- **Attempt the following tasks exactly in the given order.**
- Indent your code properly.
- Use meaningful variable and function names. Follow the naming conventions.
- Use meaningful prompt lines and labels for all input/output.
- Make sure that there are **NO dangling pointers** or **memory leaks** in your program.

Task # 1

Personal Information Class

Design a class that holds the following personal data: name, address, age, and phone number. Write appropriate accessor and mutator functions. Demonstrate the class by writing a program that creates three instances of it. One instance should hold your information, and the other two should hold your friends' or family members' information.

Task # 2

Write a Circle class that has the following member variables:

- radius: a double
- pi: a double initialized with the value 3.14159

The class should have the following member functions:

- **setRadius**. A mutator function for the radius variable.
- **getRadius**. An accessor function for the radius variable.
- **getArea**. Returns the area of the circle, which is calculated as

  area = pi * radius * radius
- **getDiameter**. Returns the diameter of the circle, which is calculated as

  diameter = radius * 2
- **getCircumference**. Returns the circumference of the circle, which is calculated as

  circumference = 2 * pi * radius

Write a program that demonstrates the Circle class by asking the user for the circle's radius, creating a Circle object, and then reporting the circle's area, diameter, and circumference.

Task # 3

Create a class **Date** that has 3 private member variables to store following three attributes:

- Day (an integer)
- Month (an integer)
- Year (an integer)

Note: *All member functions of the* **Date** *class, which are not supposed to change data stored in the calling object, should be declared as* **const** *member functions.*

a) Implement the getter and setter functions for each of the above three member variables.

Names of these functions should be: **setDay**, **setMonth**, **setYear**, **getDay**, **getMonth**, **getYear**.

b) Implement a public member function **displayDate** of the **Date** class. This function should display the date in the following format. For example, the date 15th March 2016 should be displayed on screen like **15/03/2016**

c) Write a main function (driver program) to test the working of **Date** class and its member functions.

d) In the main function, allocate an array of 5 **Date** objects. Store these five dates in these Date objects: 25th December 1876, 9th November 1877, 21st April 1938, 14th August 1947, and 11th September 1948. Now, print these five dates by calling the appropriate member function of the **Date** class.

Task # 4

Create a class **SaleItem** that a grocery store might use to represent a particular item being sold at the store. The class **SaleItem** should have private member variables to store each of the following:

- ID (an integer to uniquely identify each item)
- Name (a char array of size 41)
- Quantity (an integer to represent how many such items are present in the store)
- Price (a double value indicating the price of the item in Rupees)

Note: *All member functions of the* **SaleItem** *class, which are not supposed to change data stored in the calling object, should be declared as* **const** *member functions.*

a) Implement the **getter and setter functions** for each data member of the **SaleItem** class.
b) Implement a public member function **incrementQuantity** which will take an integer value as argument and increments the quantity of the calling **SaleItem** object by that much.
c) Implement a public member function **decrementQuantity** which will take an integer value as argument and decrements the quantity of the calling **SaleItem** object by that much.

d) Implement a public member function **displayInformation** of the **SaleItem** class. This function should display all the information about the calling **SaleItem** object on the screen, in a neat and readable way.

e) Write a main function (driver program) to test the working of **SaleItem** class and its member functions.

f) Write a menu based program using the **SaleItem** class. Your program should ask the user that how many **SaleItem**s he/she wants to create. Then, it should dynamically allocate an array of **SaleItem**s. Then, the program should ask the user to enter the values of all attributes for each **SaleItem** object (see the input format in the following sample run).

After that a menu should be displayed to the user, showing the ID, Name, Quantity, and Price of each item. The user should be able to select (buy) any **one** item by specifying its **ID** and the **quantity** that he/she needs to buy. After that, the program should display the bill of the user (see the output format in the following sample run). Make sure that your program **decrements** the quantity of the **SaleItem** which was bought by the user. After displaying the bill the user should be asked to press any key to return back to the menu. The program should eventually terminate when the user enters **-1** at the menu. At the end, make sure that all the dynamically allocated memory is properly deallocated.

Note: *In the following sample run, text that is shown in **red** is entered by the user.*

```
How many SaleItems you want to create: 3

Enter details of SaleItem # 1: 901,Tuc Biscuits,150,15
Enter details of SaleItem # 2: 808,Lays Chips,100,20
Enter details of SaleItem # 3: 850,Dairy Milk Choc,200,10

          Menu
901   Tuc Biscuits          150   15
808   Lays Chips            100   20
850   Dairy Milk Choc       200   10

Enter the ID of item that you want to buy (-1 to exit): 901
Enter the Quantity you want to buy: 6

Your Bill is as follows:
ID    Name            Quantity        Unit Price
-----------------------------------------------
901   Tuc Biscuits    6               15

                      Total Amount:   Rs.90
Press any key to return to menu: ↵

          Menu
901   Tuc Biscuits          144   15
808   Lays Chips            100   20
850   Dairy Milk Choc       200   10

Enter the ID of item that you want to buy (-1 to exit): -1

Bye Bye!!
```