Software Engineering Design – Advanced Programming Techniques

LAB ASSESSMENT 2 (30%) – Semester 2022C

Test Duration: 150 mins (+ 15 mins for submission)

NOTE: only do and submit **three files for three questions 1-3, and don't zip them together.**
Try doing easy questions first, and should try all three questions (score will be always given for attempt).

Question 1 (8 pts)

Define a class namely **Vehicle** with two attributes are ***name*** (string) and manufacturing ***year*** (int). Define subclass namely **Car** (inherits from the Vehicle class) which has an additional attribute is ***plateNumber*** (int).

For each class, provide a **constructor** to initialize all attributes. Create an **array *of 03 car objects*** using *dynamic memory allocation*. Find and print out the information of *oldest car (with smallest **year**)*.

Question 2 (12 pts)

A company implements a referal program that each customer will get ***bonus*** 5% of *expense* of the directly referred customer and 2% of *all expenses* of indirectly referred people. The list of connected customers and ***expense*** of each customer is as below

```
Peter (1000) -> John (200) -> Harry (1200) -> Luna (400)
```

a) Use **singly linked list** concepts to record those information. Write a function to print out the referral connections in *forward direction* exactly as above.

   Hint: Define a class, e.g. namely **Customer**, with attributes are ***name***, ***expense, bonus***, **next*.

b) Write a function to find and print out the name of Customer with *largest expense*.

c) Write a function namely `updateBonus(Customer *head)` to update and show the ***bonus*** values for all customers in the list.

Sample Run:

```
Peter (1000) -> John (200) -> Harry (1200) -> Luna (400)
Customer with largest expense: Harry
All bonuses: Peter: 42  John: 68  Harry: 20
```

Question 3 (10 pts)

Write a simple C++ program to help manage a retail shop as below:

The **shop** has name, list of selling products, and need to record its total revenue (sum of all customers' bill values). Each **product** has *name* (string), *id* (integer) and *price* (double). The shop can add a new product to its selling list (*given name, id and price of the product*).

The shop manage each **customer** by *name, id*, *last bill* and *total spent amount* (sum of all bills' values including the last one). The shop can record a new customer with *given name and id*, and it has a function to calculate and print out the **bill for each customer** with the given *customer's id* and list of *product id*. Each bill should have value and list of the bought products.

Implement classes with suitable attributes and methods to satisfy the above requirement. Test them in main() with appropriate output messages. Note: to save time, attributes can be declared as public.