

# Programming and Data Structures – Section 15

Pre-Lab Practice Exercises for the Lab class on 19-01-2023

## (Topic: Structures)

- Write programs for the following problems yourself before coming to Lab on 19-01-2023
- No need to submit the solutions. These are purely for your practice.

1. Define a structure named **customer** to specify data of the customers in a bank. The data to be stored is: Account number (integer), Name (character string having at most 50 characters), and Balance in account (integer).

Assume data for 10 customers of the bank are stored in the structure array :

```
struct customer bank[10];
```

A function named, **transaction**, is used to serve customer requests for withdrawal or deposit into the customer's account. Every such request is represented by the following three parameters: Account number of the customer, request type (0 for deposit and 1 for withdrawal) and amount. The function prototype is as follows:

```
int transaction (int account_number, int request_type, int amount, struct customer bank [ 10 ])
```

The array **bank** (defined above) is an input to the transaction function and is suitably updated after every request. In case of a failed transaction, no change is made in the bank array.

The function **transaction** returns 0, if the transaction fails and 1 otherwise. A transaction fails only when the account balance is less than the withdrawal amount requested.

Write the **main()** function which will read the names of five customers and populate their account number serially, starting from 100. The balance for a customer should be a random number generated in the range [10000, 100000]. It should then display the details of all the customers properly formatted.

The **main()** function should in an infinite loop ask the user to enter either 0, 1, or 2. If the user enters any other value, it will keep on asking the user until he/she enters a valid value. If the user enters 0, it should exit the program. If the user enters either 1 or 2, it should ask the account number of the customer and the transaction amount, and then invoke the function **transaction** for a deposit or withdrawal (depending on whether the value is 1 or 2). For a withdrawal transaction, if the total balance is less than the withdrawal, it would be a failed transaction and should display an appropriate message. Other wise, it should change the account balance appropriately. It should display the account balance before and after the

withdrawal.

2. A restaurant needs to develop the following software to automate its order processing. For this, write the following functions to develop a simple software:
  - a) In the main program define an array (called **order\_list**) of 50 structures named **order** to handle orders placed in the restaurant. The structure **order** has the following members: name: string of 10 characters, up to 5 item codes, quantity of each item ordered (integer). The items served by the restaurant are coded in integers from 1 to 20. In an infinite loop display a menu and prompt the user to enter choice (a character). Based on the user choice (a, or, b, or c), call the following functions appropriately, and exit the program if the user enters 'e'.
  - b) **create\_order**: This function should ask the name of the customer and upto five item names. Based on this, create and insert the **order** at the end of the **order\_list** and print all the **orders** (customer name and items ordered and corresponding quantities) till now.
  - c) **serve\_order**: This function should display and delete the oldest order. It should also display all pending orders till now.
  - d) **display\_statistics**: Display the statistics regarding the items sold and the corresponding quantities.

Test your programs by giving different inputs and checking if you are getting correct outputs.

Appropriately document your code.

—The end —