

## COSC1410: Spring 2016

### Assignment 8

[1] **Objective:** This assignment will provide practice on the following concepts:

- Classes,
- Array of structures, and
- Binary search.

[2] **Description:** We are going to process customer accounts in this assignment. The program should be able to generate unique account id and balance randomly according to the number of customer in a bank. Each *customer* has following two attributes:

- **ID**: A unique integer number created by a random number generator. The ID should be at least 6 digits.
- **Balance**: a positive integer value range from **0** to **1,000,000**.

After customers have been generated, print out ID and balance of all the customers. Your program should allow the user to interactively query the accounts for balances based on the account ID. If the ID is not in the customer array, display a message. You have to use binary search to find the ID in the array. Also you have to print how many comparisons were made to find that ID in binary search. Repeat the process until user enters 0 as ID.

[2] **Criteria:**

- Create a *customer* structure with two members: int *ID* (storing a unique customerId), and int *balance* (storing the customer's balance).
- Create a *bankAccount* class that will store an array of customers as defined above plus the current number of accounts.
- Max number of customers is **100**
- The balance should be a double. The range of balance is **0** to **1000000**.
- Implement a binary search function. The function will return the index of the **ID** in the array (if available) or -1 if the ID is not available. The function also keeps track of the number of comparison attempts for each binary search.
- You are required to use a C-structure, a C++ class.
- Some members of the class must be private.

[4] **Input:** Since we are unable to use a file yet, we will generate the accounts using random number generators again. In order for the binary search to work, we must have all accounts arranged in increasing order of the IDs. We can achieve that by generating accounts with IDs higher than the last one. Your program should generate 40 (this may change) customers randomly. Then user will enter ID to get the corresponding balance. When user enters 0 as ID, the program quits. To get the next ID number, you can randomly generate a number between 1 and 100 and add that to the last ID. To generate a random double, just generate a random number and multiply it with a suitable double.

[5] **Output:** The program will show generated IDs and balances. After that it will show the balance of the ID entered by the user. The program will also show the number of comparison attempts made for each binary search.

## Sample output

```
Generated IDs and balance :
Row  ID      Balance
  1  100097    30504.42
  2  100177    180243.27
  3  100242    305192.64
  4  100292    300492.04
  5  100356    338010.25
  6  100359    350800.83
  7  100391    311043.65
  8  100478    378954.95
  9  100556    117985.06
 10  100596    275937.59
 11  100629    324984.64
 12  100648     11887.57
 13  100712    342203.68
 14  100778    306677.04
 15  100812    101916.43
 16  100867    403831.02
 17  100937    391411.54
 18  100985    357666.18
 19  100994     29539.56
 20  101081     43777.43
 21  101165     32953.68
 22  101177    276123.14
 23  101237    395233.87
 24  101313    267389.92
 25  101388     30318.87
 26  101433     58708.02
 27  101489     95657.21
 28  101525    128598.52
 29  101575    168788.65
 30  101629    321867.40
 31  101676     58386.40
 32  101709     36528.61
 33  101784    371570.06
 34  101840    187974.52
 35  101866     26075.96
 36  101872    376406.73
 37  101958     42849.68
 38  101966    288975.57
 39  102065     52795.16
 40  102119    383791.62
Enter an ID (0 to quit): 101313
6 comparisons made in searching for searching 101313. The balance is 267389.92.
Enter an ID (0 to quit): 101709
4 comparisons made in searching for searching 101709. The balance is 36528.61.
Enter an ID (0 to quit): 102118
6 comparisons made in searching for searching 102118.
Enter an ID (0 to quit): 102119
6 comparisons made in searching for searching 102119. The balance is 383791.62.
Enter an ID (0 to quit): 0
Press any key to continue . . .
```