Lab 8

Due on: November 15^{th}, Friday 11:59 pm

Task: Driver Programs

Sometimes when writing larger modular programs, individual functions can have significant complexity. In those situation, programmers will sometimes write a separate program just to test or exercise an individual function. In this kind of scenario, the main function contains a call to the function being tested, and some test data that properly exercises the tested function.
For the assignment this week, you are asked to create a program that manages a bank database by changing the data depending on users' input values. For this lab exercise, write separate driver programs that will test the following functions:

1. menu
2. newAccount
3. deposit
4. withdraw
5. gamblingIdx
6. swapValues

Go ahead and read the Assignment 10 write-up at this point so that you clearly understand the requirements of each function.

Submit a separate .cpp file for each driver program (**6 total**). Use fairly short arrays for your driver programs (something that can be easily viewed in the command prompt, say 10 to 20 elements long). Print the contents of your test arrays before-and-after inside of each of your driver programs (you can have an additional function for printing array contents). For now, implement function stubs (dummy code, don't need implementation yet). Once you have the lab written, you will be able to use the driver programs to help in developing and testing the function implementation for the assignment. Take a look at the example to understand what we mean by driver program.

Instructions to submit your Lab work

Zip all the .cpp files together and submit the resulting .zip file through Moodle as Lab 8 by due date. You do not need to submit any executable files.

Keep in mind the Honor code and ensure that you do not violate any of the rules it entails.

Example:

```
#include <iostream>
#include <ctime>
using namespace std;

void dispArray(int arr[], int length);  // auxiliary function
void foo(int arr[], int length);        // function being tested

int main() {
    const int N = 10;

    int arrayA[] = {1, 3, 2, 0, 2, 2, 7, 0, 2, 0};

    dispArray(arrayA, N);
    foo(arrayA, N, 0);
    dispArray(arrayA, N);

    return 0;
}

void dispArray(int arr[], int length) {
    // This is an auxiliary function. Not the function being tested.
    for (int i=0; i<length; i++){
        cout << arr[i] << ",";
    }
    cout << endl;
}

void foo(int arr[], int length) {
    // Function stub for now. To be completed later.
    // Let's just fill the array with -1s as dummy data.
    for (int 9=0; i<length; i++) {
        arr[i] = -1;
    }
}
```