# PROGRAMMING ASSIGNMENT #4

Write a C program that allows the user to make some simple ***banking transactions***.

The program should first prompt the user to enter the **current balance** of his/her bank account (in dollars and cents, *not less than zero*). The program should then prompt the user to enter the **number of withdrawals** to make, and then the **number of deposits** to make.

For this assignment, let's set a ***maximum of 5 deposits and 5 withdrawals,*** you'll see why as you read on.

Using a loop, the program should then prompt the user to enter the **amount of the first deposit** (a ***positive*** amount to add to the bank account balance), the **amount of the second deposit**, the third, **etc**., until the number of deposits have been processed.

Using a second loop, the program should then prompt the user to enter the **amount of the first withdrawal** (a ***positive*** amount to subtract from the bank account balance, if possible), the **amount of the second withdrawal**, the third, **etc.** until the number of withdrawals have been processed.

Once all deposits and withdrawals have been made, the program should output the **ending balance.**

The dialog with the user should look like the following:

```
Welcome to the Computer Banking System

Enter your current balance in dollars and cents: 256.40

Enter the number of deposits (0 - 5): 3

Enter the number of withdrawals (0 - 5): 2

Enter the amount of deposit #1: 10.50
Enter the amount of deposit #2: 12.25
Enter the amount of deposit #3: 125.30

Enter the amount of withdrawal #1: 120.35
Enter the amount of withdrawal #2: 35.60

*** The closing balance is $248.50 ***
```

The program should also output one of the following messages based on the closing balance. That is:

- If the closing balance is greater than or equal to 50000.00, output:
  "*** It is time to invest some money! ***"
- If the closing balance is between 15000.00 and 49999.99, output:
  "*** Maybe you should consider a CD. ***"
- If the closing balance is between 1000.00 and 14999.99, output:
  "*** Keep up the good work! ***"
- If the closing balance is between 0.00 and 999.99, output:
  "*** Your balance is getting low! ***"

So, in the above example, the last line of sample output would be:

***** Your balance is getting low! *****

Regarding ***error checking*** on ***all*** user input, the following 5 error checks (on user input) should be included in your program:

1. Ensure that the starting (current) balance is a positive number. If not, the following error message should be displayed:

***** Beginning balance must be at least zero, please re-enter.**

2. Ensure that the number of deposits is between 0 and 5. If not, the following error message should be displayed:

***** Invalid number of deposits, please re-enter.**

3. Ensure that the number of withdrawals is between 0 and 5. If not, the following error message should be displayed:

***** Invalid number of withdrawals, please re-enter.**

4. Ensure that the deposit amount is equal to or greater than zero. If not, the following error message should be displayed:

***** Deposit amount must be greater than zero, please re-enter.**

5. Ensure that the withdrawal amount does not exceed the current balance (including the new deposits). If so, the following error message should be displayed:

*** **Withdrawal amount exceeds current balance.**

So, for example, a sample run of the program ***with error checking*** might look like:

```
Enter current balance in dollars and cents: -56.40
*** Beginning balance must be at least zero, please re-enter.

Enter current balance in dollars and cents: 256.40

Enter the number of deposits (0 - 5): -3
*** Invalid number of deposits, please re-enter.
Enter the number of deposits: 3

Enter the number of withdrawals (0 - 5): 2

Enter the amount of deposit #1: 10.50
Enter the amount of deposit #2: -12.25
*** Deposit amount must be greater than zero, please re-enter.

Enter the amount of deposit #2: 12.25
Enter the amount of deposit #3: 125.30

Enter the amount of withdrawal #1: 5000.00
*** Withdrawal amount exceeds current balance, please re-enter.

Enter the amount of withdrawal #1: 120.35
Enter the amount of withdrawal #2: 35.60

*** The closing balance is $248.50 ***
*** Your balance is getting low! ***
```

Now, the reason we restrict the maximum number of deposits and withdrawals to 5 has to do with the *array sizes*. You want to be sure that the user does not want to enter more information than you can store in your arrays. In my example above, I made the array sizes "5". That means that I can store up to 5 deposits and 5 withdrawals. So, I want to test that the user does not plan to enter more than that.

So, for example, if the user enters 10 as the number of deposits, the following should be displayed;

```
Enter the number of deposits (0 - 5): 10
*** Invalid number of deposits, please re-enter.
```

Lastly, you are to keep track of all of the deposits and all of the withdrawals so that you can print them out in "record" form. You do this by storing them in **arrays**. You will have one array to hold the deposit amounts entered by the user, and another array to hold the withdrawal amounts. You want to be sure that the size of the arrays are large enough to handle 5 deposits and 5 withdrawals. Perhaps:

**float deposits[5], withdrawals[5];**

**Based on the above program run, the Bank Record should look like:**

```
   *** Bank Record ***

Starting Balance: $ 256.40

Deposit #1:          10.50
Deposit #2:          12.25
Deposit #3:         125.30

Withdrawal #1:      120.35
Withdrawal #2:       35.60

Ending Balance: $  248.50
```

*Try to align the decimal points as closely as possible in your output.*

**Note**: There are some bothersome complications with the entry and storage of some float numbers. Because of the way data is stored the float number 22.39 may be stored as 22.389999 and this may cause a problem with your final if statements and even with balance equaling zero. In other words, it is nearly impossible to test for equality with variables of type float. ***This is a well known anomaly in C.*** You may not encounter these problems, I just wanted to warn you of the potential problem.

Read all the specifications carefully. A good detailed design before you start to code, will go a long way here.

Good luck!