# EECS 10: Assignment 5

October 22, 2004

Due Monday 11/01/2004 12:00pm

## 1 ATM Menu [20 points ]

Write a program to display the menu of an ATM where the user can do several operations such as ”check balance”, ”deposit money”,”withdraw money”, ”add interest”, ”exit”. The menu should be repeated until the user chooses option ’e’.
The menu should display as following:

```
****************************
* Welcome to EECS10 ATM!
* a)Check balance
* b)Deposit money
* c)Withdraw money
* d)Add interest
* e)Exit
* Please input your choice:
****************************
```

If the user chooses ”a”, it will print out current balance of the account as following:

```
The total balance is 123.45
```

If the user chooses ”b”, it will display the current balance and ask the user to input the amount of deposit.

```
Current balance is 900.23
Deposit:  please input the amount:100
```

Then it adds the amount and print out the new balance.

```
Current balance is 1000.23
```

If the user chooses ”c”, it will display the current balance and ask the user to input the amount.

```
Current balance is 120.90
Withdraw:  please input the amount:
```

If the user inputs an amount larger than current balance, it will reject the operation and ask the user to input the amount again until the input amount is no larger than current balance.

```
Sorry, there isn’t enough money in your account!
Withdraw:  please input the amount again:
```

It will display current balance again after withdrawal.

If the user chooses ”d”, it will print out current balance and ask the user input the value of apr.

```
The total balance is 123.23
Please input APR:
```

It will calculate the interest as following: interest=balance*(apr/100)
Your program should print the interest, add it to the balance and then show the new balance again.

If the user chooses ”e”, the program will end.

**HINT**
To choose a certain opertaion on the menu, you may use ”switch” and ”break”. The ”switch” statement should be nested inside the loop that repeatedly shows the menu and executes the selected option until ”Exit” is chosen.

Implement the program in C and record the results in your `bank.script` file for the following case:
The test case includes serveral steps. You MUST follow those steps once you run your program!
1) Deposit 700.50
2) Check balance
3) Withdraw 100.50
4) Withdraw 800 (display reject operation info.)
5) Withdraw 600.50 (display reject operation info. again)
6) Withdraw 200.00
7) Check balance (should be 400.00 now)
8) Add interest
9) Input APR as 1.5
10)Check balance
11)Exit

## 2 Square root[20 points + 5 points (extra credit)]

Write a program to calculate the square root of a positive floating point value.
At the beginning, the program will ask the user to input the positive integer (range between 1 and 1000).

```
Please input the positive number(1 to 1000):
```

We will use an approximation technique for this assignment. Instead of computing the squareroot directly, the program will try different values repeatedly, coming closer to the real value with every iteration.

For example, for the squareroot of 3, the program will start with guess = 0 and find that 0*0 = 0 is lower than 3. So, it will try again with guess = 1, which still is too low (1*1 = 1 smaller than 3). In the next interation, the program will then try 2*2=4 which is too large. At this point, the increment of 1 is too big, so a smaller increment should be used, say 0.1, and the trial-and-error approximation will continue with the guesses 1.1, 1.2, 1.3, ... 1.7. At 1.8, again the square will be larger than than 3 (1.8*1.8=3.24), so the increment should be reduced to 0.01, and so on.

Theoretically, this algorithm will come infinitely close to the actual squareroot value. However, the precision of floating-point values is limited, so we will stop the approximation after 5 digits after the decimal point are reached.

For example:
```
Please input the positive number(1 to 1000):  278
The square root of 278 is 16.67333
```

**For 5 extra credits**:
Explain how to improve your algorithm so that it takes less steps to get an accurate square root.

## 3 What to turn in

Use the submission method as for the previous assignments to submit the following files:

- `bank.c`

- `bank.script`
- `root.c`
- `root.script`
- `root-extra.txt`