LP104 Object-Oriented Programming
Exercise – Classes and Objects

In this lab, you are required to complete the following tasks. Your programs must pass the compilation and testing.

Tasks

Given the following class definition (use the skeleton code below).

```
class Account{
public:
      //Constructor
      Account(){ balance = interestRate = 0.0; term = 0;}
      Account(double b, double r, int t);

      //Accessor
      double getBalance() { return balance; }
      double getInterestRate() { return interestRate;}
      int getTerm() { return term;}

      //Mutator
      void Deposit(double amount);
      int WithDraw(double amount);
      void AdjustRate(double newRate);
      void SetTerm(int newTerm);
      void CalBalanceMaturity();

private:
      double balance;
      double balanceMaturity;
      double interestRate;
      int term;
};
```

You need to complete the definitions of the **member functions**:

1. **`Account()`**

Constructor **`Account(double b, double r, int t)`**, which is not the default constructor, is used to initialize the values for **`balance`**, **`interestRate`** and **`term`** by **`b`**, **`r`** and **`t`**, respectively.

2. **`Deposit()`**

This function will deal with the deposit request from the users, i.e., increase the balance with the amount of money specified by **`amount`**. For example, when Tom deposits MOP 200 to his account with the initial balance MOP 1,000, then the new balance for his account will be MOP 1,200.

3. **`WithDraw()`**

Similar to **`Deposit()`**, **`WithDraw()`** will deal with the withdraw request from the users, i.e, decrease the balance with the amount of money specified by **`amount`**. However, your function must check whether the amount of the withdrawn money is greater than the balance of the account. If so, your function should do nothing to the balance but just display error information.

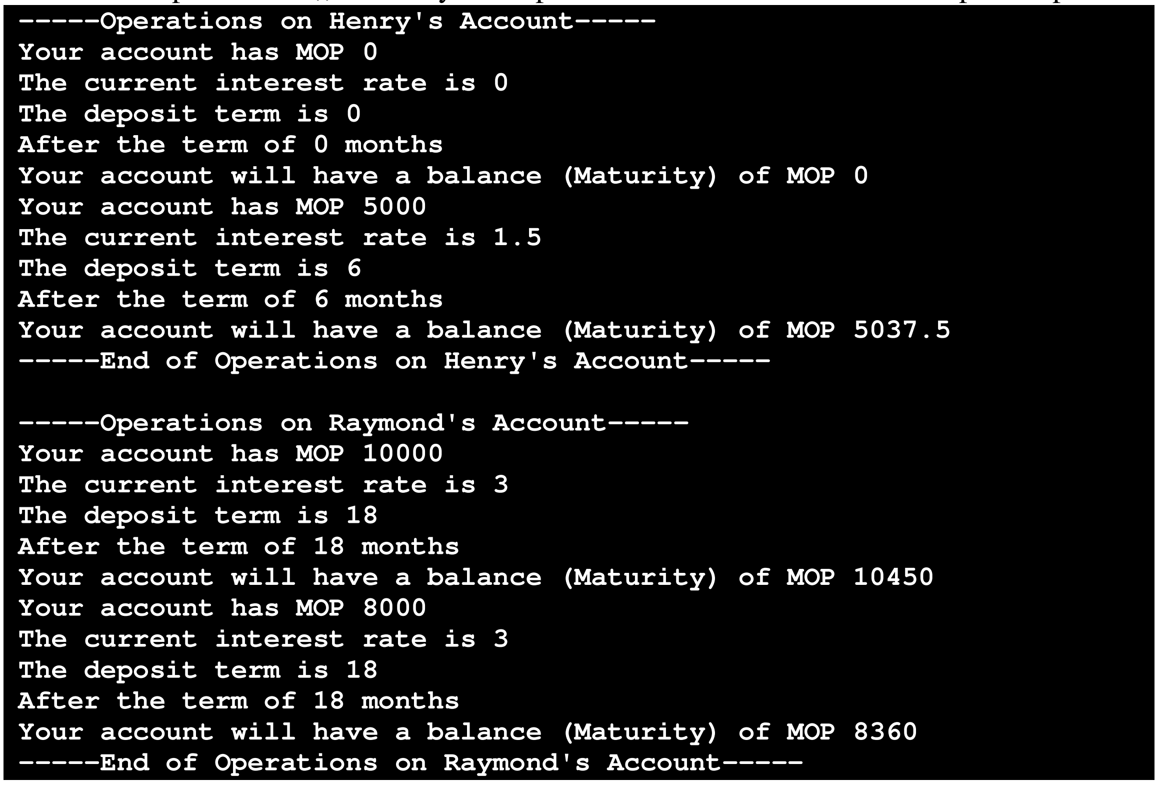
4. **`AdjustRate()`**

It is just the set the interest rate for an account.

5. **`SetTerm()`**
It is just the set the number of terms for an account.

6. **`CalBalanceMaturity()`**
When a mount of money is deposited in the bank for several months or terms, the user can obtain a certain mount interest given by the bank. To calculate the new balance after the maturity (when the terms reach), you can use the following equation:
$i = b \cdot r \cdot t$ where i denotes the interest (MOP), b denotes the balance (MOP), r denotes the interest rate (you need to convert it from percentage to real numbers) and t denotes the terms in real numbers. For example, the original balance for Raymond's account is MOP 10,000, the interest rate for 12 months is 3 and the term is 18 months. Then, Raymond's account will obtain an interest of $i = 10000 \cdot (\frac{3}{100} \cdot \frac{18}{12}) = 450$. Then, after the maturity, the account will have a new balance: $m = b + i$ where m denotes the balance after the maturity, b denotes the balance and i denotes the interest. Let us consider the previous example again. After the maturity of 18 months, Raymond's account will have a new balance of MOP 10,450. **`CalBalanceMaturity()`** will also output the mature balance after the given terms.

7. Complete **`main()`** to test your implemented member methods. Sample Output:



```
-----Operations on Henry's Account-----
Your account has MOP 0
The current interest rate is 0
The deposit term is 0
After the term of 0 months
Your account will have a balance (Maturity) of MOP 0
Your account has MOP 5000
The current interest rate is 1.5
The deposit term is 6
After the term of 6 months
Your account will have a balance (Maturity) of MOP 5037.5
-----End of Operations on Henry's Account-----

-----Operations on Raymond's Account-----
Your account has MOP 10000
The current interest rate is 3
The deposit term is 18
After the term of 18 months
Your account will have a balance (Maturity) of MOP 10450
Your account has MOP 8000
The current interest rate is 3
The deposit term is 18
After the term of 18 months
Your account will have a balance (Maturity) of MOP 8360
-----End of Operations on Raymond's Account-----
```