

National University of Computer & Emerging Sciences

CS 103 – Computer Programming
Spring 2014

Assignment # 05: FAST Banking System (FBS)
Deadline : 14 March 2014

FAST Banking System

Goal: To design and build a FAST banking management system in C++ using learned Object Oriented techniques.

Deployment: The built system will be deployed in a bank. [Currently version of the system will be used by a bank employee. We will later add on the functionalities from the user perspective].

FAST Banking System

- Account Holder
 - FBS must record the detailed information of each account holder, such as
 - Type of his account (see next slides for more info)
 - First name, middle name, and last name
 - CNIC number
 - Address
 - Telephone number
 - Date of birth
 - In addition it must allocate a **unique account number** to each account holder.

Account Information

An account holder can have two different types of account those are:

1. Checking account

Customer is not paid any interest

2. Saving account

Customer deposit the money for a fix duration at 5% interest per month over the duration. [Hint: Store the year and month of the deposit]

In addition, both types of account will be liable to service charges and government taxes (details are given on the next few slides).

1. Service charges

- All withdrawal transactions over Rs. 5000/- are subject to a 0.2% service charges deducted automatically from the account

2. Government Tax

- Any withdrawal transaction over Rs. 50,000/- during one day is subject to a 2.5% tax to be deducted from the account automatically
- The profit/interest paid to the customer is subject to a 0.5% tax to be deducted from the account automatically (applicable only to saving accounts)

Following few slides show the desired functionality via output screens.

FAST Banking System

Welcome Screen for Bank Employee (System User)

- A. Create a new account.
- B. Operations over existing customers.
 - 1. Generate Mini Statement of a Bank Customer.
 - 2. Withdraw money from a customer account.
 - 3. Deposit money.
 - 4. Balance Inquiry
 - 5. Transaction history,
 - 6. Any other operations you can think of.
- C. Generate the list of all customers with accounts types and their balance.

A. Create Account

- Add a new account to the existing accounts.
- INPUT:
 - All the customer information

1. Mini-statement

- A customer can request to see mini-statement for any of his accounts
- INPUT:
 - Account number
 - CNIC number
- OUTPUT:
 - Last 10 transactions

2. Withdraw money

- INPUT:
 - Account number
 - CNIC number
 - Amount to be withdrawn
- OUTPUT:
 - System response containing amount withdrawn and the balance
- CHECKS:
 - Make sure that the customer has sufficient funds in his account

3. Deposit money

- INPUT:
 - Account number
 - CNIC number
 - Amount to be deposited
- OUTPUT:
 - System response containing amount deposited and the new balance
- CHECKS:
 - Make sure that the valid amount has been entered.

4. Balance inquiry

- A customer can check the balance in his account
- INPUT:
 - Account number
 - CNIC number
- OUTPUT:
 - Amount in the account

5. Transaction history

- A customer can request to see transaction history for any of his accounts
- INPUT:
 - Account number
 - CNIC number
- OUTPUT:
 - Transaction history for all the transactions.

Implementation Guidelines

Instructions

1. Before writing code, **please** build the class hierarchy i.e. identify different classes and their interaction.
[Follow the principle of *divide and conquer*]
2. Using goto statements or global variables or any other poor design or programming practice will cost you marks. (-1 Marks for each violation).