**Review – III**



**Project title**

**The Ultimate Banking Management System.**

Name: Hari Krishna Shah

VIT ID: 21BCS0167

Team Members: Hari Krishna Shah only.

Contact Number: +977 9844523112

Email: [harikrishna.shah@vitstudent2021.ac.in](mailto:harikrishna.shah@vitstudent2021.ac.in)

Abstract

Banking sector is regarded as the backbone of the economy. A more digital and technologized banking system significantly helps the nation’s economy to grow and flourish. In this project, I will be digitalizing the traditional banking system in order to boost the output of the bank. To do so, I will be taking object-oriented approach. My program will also be data centric which will try to maximize the protection of data of the bank and the clients of the bank because such data are extremely confidential and can cause huge damage if they are corrupted or manipulated by the wrong person. Hence, the concept of classes and object, that are part of object-oriented programming, which provides us data protection and abstraction along with encapsulation will be beneficial. We will also be using the concept of polymorphism to create multiple functions and operators of same name but different functions by the means of function overloading and operator overloading. This will make our program hassle free.

I aim to achieve below features in my program.

* Bank account creation and deletion.
* Deposit and withdrawal feature
* Overdraft feature
* Loan disbursement feature
* Advanced loan tracking and analysis
* Client risk profiling for the applied loan
* Collateral management and analysis
* Interest on loan management features
* Interest payable to client management
* Collateral evaluation features
* Statutory Liquidity Ratio monitoring feature
* Credit to Core Capital monitoring feature
* Bad debt alert system
* Decision support system
* Database management feature
* Data Protection
* Employee management features

As you can see there are a lot of features mentioned above. My program will have all those features and make banking system more efficient and effective. It will aid the bank in all the possible ways to carry out their functions.

These all features make my program the ultimate solution for the banking management system.

**Introduction**

The program will be made using the concepts of objected oriented programming. Following classes will be used in this program:

|  |  |  |  |
| --- | --- | --- | --- |
| Classes | Aim | Data Members | Member Function |
| 1. Bank | To store and calculate bank related transactions. | Name, id number, date of birth. | Get(), display(), withdraw(), deposit(), balance check() |
| 1. Address | To store the addresses of the clients. | Area, city, district, country. | Get(), display() |
| 1. Assets | To store the asset details of the clients in case they apply for loan.  The data will be used to calculate the optimal loan amount that can be given to them. | Asset type, price, total worth, | Get(), display(), calculate loan optimal amount() |
| 1. Saving Account | To store money details in saving account | Float amount | Get(), display(), withdraw(), deposit() |
| 1. Current Account | To store money details in current account | Float amount | Get(), display(), withdraw(), deposit() |

The following concepts are used:

1. Classes and Objects
2. Operator overloading: to calculate total balance
3. Friend Function: to allow employee or tax officials to check balance amount
4. Inheritance: to allow derived classes like saving account and current account to get details of the employee
5. Pass by reference: used while calculating the total balance amount

**Implementation**

#include <iostream>

#include <malloc.h>

**using** **namespace** std**;**

**class** Address**{**

**protected:**

**char** city**[**100**];**

**char** district**[**100**];**

**char** country**[**100**];**

**public:**

**void** get**();**

**void** display**();**

**};**

**class** saving\_account**{**

**protected:**

**float** amount**;**

**float** acc\_no**;**

**public:**

**void** get**();**

**void** display**();**

**void** withdraw**();**

**void** deposit**();**

**};**

**class** current\_account**{**

**protected:**

**float** amount**;**

**float** acc\_no**;**

**public:**

**void** get**();**

**void** display**();**

**void** withdraw**();**

**void** deposit**();**

**};**

**class** asset**{**

**protected:**

**char** asset\_name**[**100**];**

**float** amount**;**

**public:**

**void** get**();**

**void** display**();**

**};**

**class** Bank**:** **public** Address**,** **public** saving\_account**,** **public** current\_account**,** **public** asset**{**

**protected:**

**char** name**[**100**];**

**long** **long** **int** national\_id**;**

**struct** Day\_of\_Birth**{**

**int** year**;**

**int** month**;**

**int** day**;**

**}**dob**;**

**public:**

**void** get**();**

**void** display**();**

**void** calculate\_optimal\_loan**();**

**void** give\_loan**();**

**void** calculate\_interest\_amount**;**

**};**