**Review – II**

**Project Title: Banking Management System**

**Object Oriented Programming**



**Team Member name: Hari Krishna Shah**

**VIT ID: 21BSC0167**

**Abstract**

This banking management system will be used to manage all the bank related requirements of clients like creating bank account, withdrawing money, depositing money, calculating the optimal amount to lend to a client, give loan to a client, and calculate interest amount to be paid.

**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**;**

**};**