/\*bank management system to create account, display, account details, modify account, deposit or withdraw from account, or generate a report of a particular account \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HEADER FILE USED IN PROJECT**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream.h>

#include <iomanip.h>

#include <string.h>

#include <ctype.h>

#include <fstream.h>

#include <conio.h>

#include <process.h>

#include <stdio.h>

fstream file; //data file path to facilitate write and read operations

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CLASS USED IN PROJECT**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Bank\_Sys//object that represents bank account of one particular user

{

private:

int acc\_no; //user account number

char name[30]; //user account name

char acc\_type; //user account type(credit/saving)

float amount; //user bank balance

public:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Function declaration**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void new\_acc(); //function prototype to create a new account

void show\_acc(); //function prototype to display the account details

void mod\_acc(); /\*function prototype to modify the details of particular

account\*/

void deposit(); /\*function prototype to deposit into user account and

make changes accordingly\*/

void withdraw(); /\*function prototype to withdraw from the account and

make changes accordingly\*/

void report(); /\*function to prototype generate a report of the user's

account and display relevant data\*/

void del\_acc(int); //function to delete a user’s account from the database

int ret\_accno() /\*function to return private data member 'account

number'\*/

{

return acc\_no;

}

};

void Bank\_Sys :: new\_acc() //function definition to create a new account

{

Bank\_Sys temp; //temporary container class object

file.open("Bank.dat", ios::binary | ios::in);

if(!file)

{

cout <<"File does not exist." <<endl;

exit(0);

}

cout <<"Enter the required account details." <<endl;

cout <<"Account Number: "; cin>>acc\_no;

while(!file.eof()) //prevents redundancy of account numbers

{

file.read((char\*)&temp, sizeof(temp));

while(acc\_no == temp.ret\_accno())

{

cout <<"Account number already in use. Please enter a new account number." <<endl;

cin >>acc\_no;

}

}

file.close();

cout <<"Name of Account Holder: "; gets(name);

cout <<"Type of Account ('C' for Credit/ 'S' for Savings): "; cin >>acc\_type;

while(acc\_type != 'C' && acc\_type != 'c' && acc\_type != 'S' && acc\_type != 's') //prevents invalid entry

{

cin >>acc\_type;

if(acc\_type != 'C' && acc\_type != 'c' && acc\_type != 'S' && acc\_type != 's')

{

cout <<"Invalid entry. Re-enter account type." <<endl;

}

}

acc\_type = toupper(acc\_type);

cout <<"Enter the initial amount (Minimum Rs. 1500): "; cin >>amount;

while(amount < 1500)

{

cout <<"The initial account balance is too low. Please re-enter the starting amount." <<endl;

cin >>amount;

}

cout <<"The account has been created successfully." <<endl;

getch();

}

void Bank\_Sys :: show\_acc() /\*function definition to show the account details

of a particular account\*/

{

cout <<"The details of the desired account are given below." <<endl

<<"Account Number: " <<acc\_no <<endl

<<"Name of Account Holder: " <<name <<endl

<<"Type of account: " <<acc\_type <<endl

<<"Balance: " <<amount <<endl;

getch();

}

void Bank\_Sys :: mod\_acc() /\*function definition to modify the details of a

particular account\*/

{

cout <<"Please make the desired changes to the account details." <<endl;

cout <<"Name of Account Holder: "; gets(name);

cout <<"Type of Account ('C' for Credit/ 'S' for Savings): "; cin >>acc\_type;

while(acc\_type != 'C' && acc\_type != 'c' && acc\_type != 'S' && acc\_type != 's') //prevents invalid entry

{

cin >>acc\_type;

if(acc\_type != 'C' && acc\_type != 'c' && acc\_type != 'S' && acc\_type != 's')

{

cout <<"Invalid entry. Re-enter account type." <<endl;

}

}

acc\_type = toupper(acc\_type);

cout <<"Enter the initial amount: "; cin >>amount;

cout <<"The changes have successfully been made." <<endl;

getch();

}

void Bank\_Sys :: deposit() /\*function definition to deposit into a particular

account\*/

{

float x = 0;//variable to store the amount of money to be deposited

cout <<"Enter the amount you would like to deposit into this account. (Cannot be more than Rs. 20000 at a time)" <<endl;

cin >>x;

while(x > 20000)/\*ensures that more than Rs.20000 is not deposited in one transaction\*/

{

cout <<"Invalid amount. Please enter an amount less than 20000." <<endl;

cin >>x;

}

amount += x;

cout <<"The account balance has been updated successfully." <<endl;

getch();

}

void Bank\_Sys :: withdraw() /\*function definition to withdraw from a certain

account\*/

{

float x; /\*variable to store the amount of money to be

withdrawn\*/

cout <<"Enter the amount you would like to withdraw from your account. (Must be less than 10000 at a time)" <<endl;

cin >> x;

while(x > 10000)/\*ensures that more than Rs. 10000 is not withdrawn in a single transaction\*/

{

cout <<"Invalid amount. Please withdraw less than 10000 at a time." <<endl;

cin >>x;

}

amount -= x;

cout <<"The account balance has been updated successully." <<endl;

getch();

}

void Bank\_Sys :: report() /\*function definition to generate a report for a

particular account\*/

{

cout <<endl;

cout <<acc\_no <<setw(10)<<" " <<name <<setw(10) <<" " <<acc\_type <<setw(6) <<amount <<endl;

getch();

}

void Bank\_Sys :: del\_acc(int n) /\*function to delete the desired account from the

database\*/

{

int flag = 1;

Bank\_Sys temp;//temporary container variable

ifstream infile;

ofstream outfile;

infile.open("Bank.dat", ios::binary | ios::beg);

outfile.open("temp.dat", ios::binary);

if(!infile || !outfile)

{

cout <<"The file does not exist." <<endl;

exit(0);

}

while(!infile.eof()) /\*fucnton to transfer all the required data to a new file

and delete the old file which contains all the

unnecessary data\*/

{

infile.read((char\*)&temp, sizeof(temp));

if(temp.ret\_accno() != n)

{

outfile.write((char\*)&temp, sizeof(temp));

}else

flag = 0;

}

infile.close();

outfile.close();

if(flag == 0) /\*renames the new file with the name of the old file in

case the condition is fulfilled\*/

{

remove("Bank.dat");

rename("temp.dat", "Bank.dat");

cout <<"The account has been deleted successfully." <<endl;

}else

{

cout <<"Account does not exist." <<endl;

}

getch();

}

void menu() /\*menu function to help the user traverse the bank

system and perform the desired operation using the

system.\*/

{

clrscr();

char n; /\*the data stored in this variable decides the course of

the program by satisfying one of the eight given

options\*/

int no = 0, flag; /\*n takes in user input to determine the desired

operation, no holds the account number to check

against the bank records during search operations\*/

double pos; /\*posistion is used to determine and place the pointer

in a particular memory location\*/

Bank\_Sys temp; /\*temporary container object to faciitate storage of

data on the binary file system\*/

do

{

cout <<endl <<"Enter the desired operation." <<endl

<<"01. Create a new account." <<endl

<<"02. Display the details of an account." <<endl

<<"03. Modify the details of an account." <<endl

<<"04. Deposit into account." <<endl

<<"05. Withdraw from account." <<endl

<<"06. View account report." <<endl

<<"07. Exit System." <<endl

<<"08. Delete an account." <<endl;

cin >>n;

switch (n)

{

/\*case 1 calls the function to create a new account and

stores the same on the binary file\*/

case '1':

clrscr();

temp.new\_acc();

file.open("Bank.dat", ios::binary | ios::app);

if(!file)

{

cout <<"File does not exist." <<endl;

exit(0);

}

file.write((char\*) &temp, sizeof(Bank\_Sys));

file.close();

break;

/\*case 2 calls the function to display the account

details by obtaining the desired data from the binary

file in which it is stored\*/

case '2':

flag = 1;

file.open("Bank.dat", ios::in | ios::binary);

if(!file)

{

cout <<"File does not exist." <<endl;

exit(0);

}

cout <<"Enter the Account Number." <<endl;

cin >>no;

while((!file.eof()) && flag != 0)

{

pos = file.tellg();

file.read((char\*) &temp, sizeof(Bank\_Sys));

if(temp.ret\_accno() == no)

{

clrscr();

temp.show\_acc();

flag = 0;

}

}

file.close();

if(flag == 1)

cout <<”Account does not exist. Please enter a valid account number.” <<endl;

break;

/\*case 3 calls the function to modify the account details and makes

the desired changes in the binary file holding the data\*/

case '3':

flag = 1;

cout <<"Enter the Account Number." <<endl;

cin >>no;

file.open("Bank.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout <<"File does not exist." <<endl;

exit(0);

}

while(!file.eof() && flag == 1)

{

pos = file.tellg();

file.read((char\*)&temp, sizeof(temp));

if(temp.ret\_accno()==no)

{

clrscr();

temp.mod\_acc();

file.seekp(pos);

file.write((char\*) (&temp), sizeof(temp));

cout<<"The record has been updated successfully." <<endl;

flag = 0;

}

}

file.close();

if(flag == 1)

cout <<”Account does not exist. Please enter a valid account number.” <<endl;

break;

/\*case 4 calls the function that allows the user to deposit the desired

amount and makes the changes in the account details in the binary file\*/

case '4':

flag = 1;

file.open("Bank.dat", ios::in | ios::out | ios::binary);

if(!file)

{

cout <<"File does not exist." <<endl;

exit(0);

}

cout <<"Enter the Account Number." <<endl;

cin >>no;

while(!file.eof() && flag!= 0)

{

pos = file.tellg();

file.read((char\*) &temp, sizeof(Bank\_Sys));

if(temp.ret\_accno() == no)

{

clrscr();

temp.deposit();

flag = 0;

file.seekg(pos);

file.write((char\*) &temp, sizeof(Bank\_Sys));

}

file.close();

}

if(flag == 1)

cout <<”Account does not exist. Please enter a valid account number.” <<endl;

break;

/\*case 5 calls the function that allows the user to withdraw from the

account and updates the balance in the binary file\*/

case '5':

flag = 1;

file.open("Bank.dat", ios::in | ios::out | ios::binary);

if(!file)

{

cout <<"File does not exist." <<endl;

exit(0);

}

cout<<"Enter the Account Number." <<endl;

cin >>no;

while(!file.eof() && flag!= 0)

{

pos = file.tellg();

file.read((char\*) &temp, sizeof(Bank\_Sys));

if(temp.ret\_accno() == no)

{

clrscr();

temp.withdraw();

flag = 0;

file.seekg(pos);

file.write((char\*) &temp, sizeof(Bank\_Sys));

}

file.close();

}

if(flag == 1)

cout <<”Account does not exist. Please enter a valid account number.” <<endl;

break;

/\*case 6 obtaines the details of the desired account from the binary

File and generates a report based on the account details and

displays the tabulated details in the form of a report\*/

case '6':

file.open("Bank.dat", ios::in | ios::binary);

if(!file)

{

cout <<"File does not exist." <<endl;

exit(0);

}

cout<<"The account holder report is given below." <<endl;

cout<<"====================================================\n";

cout<<"A/c no. NAME Type Balance\n";

cout<<"====================================================\n";

while(file.read((char\*) &temp, sizeof(Bank\_Sys)))

{

clrscr ();

temp.report();

}

file.close();

break;

case '7':

exit(0); //invocation of this case ends the program

case '8': /\*case 8 allows the administrator to delete an account from

the database\*/

cout <<"Enter the account number of the account you would like to delete." <<endl;

cin >>no;

clrscr();

temp.del\_acc(no);

break;

default: /\*default case that displays an error message whenever the

user tries to follow a path that is not present\*/

cout <<"Invalid entry." <<endl;

break;

}

}while(n!=7);

}

void introduction()/\*function to introduce the developers or administrators of the

bank management system\*/

{

cout<<"\n\n\n\t BANK";

cout<<"\n\n\tMANAGEMENT";

cout<<"\n\n\t SYSTEM";

cout<<"\n\n\n\nMADE BY : Aneez Jaheez and Srivatsan T.V\n\n\n\n";

}

void authorization()/\*funtion to ensure only the administrator is able to gain

direct access to the database\*/

{

char pass[10];

cout <<"Password: "; cin >>pass;

if(strcmp (pass, "12345") != 0)

{

cout <<"Incorrect Password." <<endl;

exit(0);

}

clrscr();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//         **THE MAIN FUNCTION OF PROGRAM**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void main() //main funtion; program execution begins here

{

clrscr();

authorization();

introduction();

menu();

getch();

}//end of the program