#include <iostream.h>

class BankAccount

{

public:

BankAccount(int=0,float=0);

void deposit(float amount) { bal += amount; }

int account\_num() const { return acctnum; }

float balance() const { return bal; }

virtual void print() = 0;

protected:

int acctnum;

float bal;

};

BankAccount::BankAccount(int num, float ibal)

{

acctnum = num;

bal = ibal;

}

class Checking : public BankAccount

{

public:

Checking(int=0,float=0,float=1000,float=.25);

int cash\_check(float);

virtual void print();

protected:

float minimum;

float charge;

};

Checking::Checking(int num, float ibal, float min, float chg)

: BankAccount(num,ibal)

{

minimum = min;

charge = chg;

}

int Checking::cash\_check(float amount)

{

char pause;

if (amount >= bal)

{

cout << endl << "Cannot cash check for Rs" << amount << " on account "

<< acctnum << "; insufficient funds." << endl;

cout << "Press enter to continue." << endl;

cin.get(pause);

return 0;

}

if (bal < minimum)

bal -= amount + charge;

else

bal -= amount;

return 1;

}

void Checking::print()

{

cout << "Checking Account: " << acctnum << endl;

cout << "\tBalance: " << bal << endl;

cout << "\tMinimum to Avoid Charges: " << minimum << endl;

cout << "\tCharge per Check: " << charge << endl << endl;

}

class ServiceChargeChecking : public Checking

{

public:

ServiceChargeChecking(int=0,float=0,float=1000,float=2500,float=.25,

float=2.5,float=10);

void interest();

virtual void print();

protected:

float intrate;

float minint;

float moncharge;

};

ServiceChargeChecking::ServiceChargeChecking(int num, float ibal, float cmin, float imin,

float chg, float rate, float monchg) : Checking(num,ibal,cmin,chg)

{

intrate = rate;

minint = imin;

moncharge = monchg;

}

void ServiceChargeChecking::interest()

{

const int nummths = 12;

const int cvtpct = 100;

if (bal >= minint)

{

float intamt = bal \* intrate / (nummths \* cvtpct);

bal += intamt;

}

else

bal -= moncharge;

}

void ServiceChargeChecking::print()

{

cout << "Interest Checking Account: " << acctnum << endl;

cout << "\tBalance: " << bal << endl;

cout << "\tMinimum to Avoid Charges: " << minimum << endl;

cout << "\tCharge per Check: " << charge << endl;

cout << "\tMinimum for Interest and No Monthly Fee: " << minint << endl;

cout << "\tInterest: " << intrate << "%" << endl;

cout << "\tMonthly Fee: " << moncharge

<< "\n\n";

}

class Savings : public BankAccount

{

public:

Savings(int=0,float=0,float = 3.5);

void interest();

int withdraw(float);

virtual void print();

protected:

float intrate;

};

Savings::Savings(int num, float ibal, float rate) : BankAccount(num,ibal)

{

intrate = rate;

}

nt funds; otherwise return true

int Savings::withdraw(float amount)

{

char pause;

if (bal <= amount)

{

cout << endl << "Withdrawal of Rs" << amount << " from account "

<< acctnum << " not permitted; insufficient funds." << endl;

cout << endl << "Press Enter to continue." << endl;

cin.get(pause);

return 0;

}

bal -= amount;

return 1;

}

void Savings::interest()

{

const int nummths = 12;

const int cvtpct = 100;

float intamt = bal \* intrate / (nummths \* cvtpct);

bal += intamt;

}

void Savings::print()

{

cout << "Savings Account: " << acctnum << endl;

cout << "\tBalance: " << bal << endl;

cout << "\tInterest: " << intrate << "%" << endl << endl;

}

void main()

{

BankAccount \* accounts[4];

// define bank accounts

Checking sara(1001,750);

ServiceChargeChecking tahir(1005,3500);

Savings kashif(1022,1000);

Checking rashid(1014,600);

accounts[0] = &sara;

accounts[1] = &tahir;

accounts[2] = &kashif;

accounts[3] = &rashid;

// set up output for dollar amounts

cout.setf(ios::fixed,ios::floatfield);

cout.setf(ios::showpoint);

cout << setprecision(2);

// checking account transactions

sara.deposit(1500);

sara.cash\_check(250);

sara.cash\_check(195.99);

sara.cash\_check(650);

sara.cash\_check(1195);

rashid.cash\_check(125.50);

rashid.deposit(1200);

rashid.cash\_check(369.99);

tahir.cash\_check(365.55);

tahir.deposit(965);

kashif.withdraw(450);

kashif.deposit(300);

kashif.withdraw(400);

tahir.interest();

kashif.interest();

// report on account balances

cout << endl << endl << "\t\tAccount Balances" << endl << endl;

cout << endl << "Account Number: " << sara.account\_num();

cout << " Balance: Rs " << sara.balance() << endl;

cout << endl << "Account Number: " << kashif.account\_num();

cout << " Balance: Rs " << kashif.balance() << endl;

cout << endl << "Account Number: " << tahir.account\_num();

cout << " Balance: Rs " << tahir.balance() << endl;

cout << endl << "Account Number: " << rashid.account\_num();

cout << " Balance: Rs " << rashid.balance() << endl;

cout << endl;

for (int i = 0 ; i < 4 ; i++)

accounts[i]->print();

}