**Bank Application**

**vi Banktask.cpp:**

1 #include <iostream>

2 #include <vector>

3 #include <thread>

4 #include <fstream>

5 #include <sstream>

6 #include <string>

7 #include <Banktask.h>

8

9 using namespace std;

10

11 bool Account::deposit(int a){

12 if(a>0){

13 this->balance+=a;

14 return true;

15 }

16 return false;

17 }

18 bool withdraw(int a){

19

20 if(this->balance>a){

21 this->balance-=a;

22 return true;

23 }

24 return false;

25 }

26 void Login::setPass(string p){

27 pass=p;

28 }

**vi Banktasknain.cpp:**

#include <iostream>

2 #include <vector>

3 #include <thread>

4 #include <fstream>

5 #include <sstream>

6 #include <string>

7 #include <Banktask.h>

8

9 using namespace std;

10

11 int main()

12 {

13 //cout<<"first"<<"\n";

14 ifstream fin("user.txt");

15 ifstream fin1("login.txt");

16 string num,T,curr\_user="";

17 int j=0;

18 vector<Account> vec;

19 vector<Login> vec1;

20 string str[10];

21 //cout <<"before"<<"\n";

22 while(getline(fin,num)){

23 //cout <<num<<"\n";

24 stringstream X(num);

25 if(num.length()>1){

26 //cout << num <<"\n";

27 j=0;

28 while(getline(X,T,' ')){

29 str[j++]=T;

30 }

31 cout << str[0] << " " << str[1]<< " " << str[2] <<"\n";

32 Account a(str[0],str[1],stoi(str[2]));

33 Login l(str[0],"",a);

34 vec.push\_back(a);

35 vec1.push\_back(l);

36 }

37 }

38 while(getline(fin1,num)){

39 stringstream X(num);

40 if(num.length()>1){

41 j=0;

42 while(getline(X,T,' ')){

43 str[j++]=T;

44 }

45 for(int i=0;i<vec1.size();i++){

46 if(vec1.at(i).getuser()==str[0]){

47 vec1.at(i).setPass(str[1]);

48 }

}

50 }

51 }

52 cout << vec.at(0).getUsername()<<"\n";

53 cout << vec1.at(0).getpass() <<"\n";

54 int in=0,bal,q=0;

55 string user,pass;

56 Login l;

57 //thread t1(Account::getAmount,3);

58 //thread t2(Account::deposit,3);

59 while(1){

60 while(1){

61 cout<< "Menu - Enter your choice \n1.Login\n2.Account\_details\n3.Deposit\n4.WithDraw\n5.Balance\n6.logout\n";

62 cin >> in;

63 switch(in){

64 case 1: cout << "Enter username and password \n";

65 cin >>user;

66 cin >>pass;

67 q=0;

68 for(int i=0;i<vec1.size();i++){

69 if(vec1.at(i).getuser()==user && vec1.at(i).getpass()==pass){

70 curr\_user=user;

71 q=1;

72 l=vec1.at(i);

73 cout <<"login successful\n";

74 }

75 }

76 if(q!=1){

77 curr\_user="";

78 cout <<"login unsuccessful\n";

79 }

80 break;

81 case 2: if(curr\_user.length()>0) {

82 cout << "User Name : " << l.getuser() <<"\n";

83 cout << "Balance : " << l.getAcc().getAmount()<<"\n";

84 }else{ cout <<" login to see details \n";}

85 break;

86 case 3:if(curr\_user.length()>0) {

87 cout << "Balance before : " << l.getAcc().getAmount() <<"\n";

88 int amo;

89 cout << "Enter amount to deposit\n";

90 cin >> amo;

91 bool b1=l.getAcc().deposit(amo);

92 //t2.join();

93 cout << b1 << "\n";

cout << "Balance : " << l.getAcc().getAmount()<<"\n";

95 }else{ cout <<" login to see details \n";}

96 break;

97 case 4: if(curr\_user.length()>0) {

98 cout << "Balance before : " << l.getAcc().getAmount() <<"\n";

99 int amo;

100 cout << "Enter amount to withdraw\n";

101 cin >> amo;

102 bool b=l.getAcc().withdraw(amo);

103 cout << b << "\n";

104 if(b)

105 cout << "Balance : " << l.getAcc().getAmount()<<"\n";

106 else

107 cout << "Insufficent balance \n";

108 }

109 else

110 { cout <<" login to see details \n";}

111 break;

112 case 5: cout <<"Balance: "<<l.getAcc().getAmount()<<"\n";

113 break;

114 case 6: curr\_user="";

115 cout <<"logout successful\n";

116 break;

117 }

118 }

119 }

120 return 0;

121

122 }

**vi Banktask.h:**

#include <iostream>

2 #include <vector>

3 #include <thread>

4 #include <fstream>

5 #include <sstream>

6 #include <string>

7

8 using namespace std;

9

10 class Account {

11 private:

12 string name;

13 string bankname;

14 int balance;

15 public:

16 Account(){}

17 Account(string n,string bn,int b){

18 name=n;

19 bankname=bn;

20 balance=b;

21 }

22 bool Account ::deposit(int a);

23

24 string getUsername(){

25 return name;

26 }

27 bool Account::withdraw(int a);

28 string getbank(){

29 return bankname;

30 }

31 int getAmount(){

32 return balance;

33 }

34 };

35 class Login : public Account{

36 private:

37 string name;

38 string pass;

39 Account acc;

40 public:

41 Login(){}

42 Login(string n,string p,Account a){

43 name=n;

44 pass=p;

45 acc=a;

46 }

47 Account getAcc(){

48 return acc;

}

50 string getuser(){

51 return name;

52 }

53 string getpass(){

54 return pass;

55 }

56 void::setPass(string p);

57

58 };

**vi Makefile:**

C = g++

2 CPPFLAGS = -c -Wall -g

3 LDFLAGS = -lcppunit -lm

4 OBJ = obj/

5 SRC = src/

6 BIN = bin/

7 INCLUDE = inc/

8

9 all: $(BIN)app

10

11 $(BIN)app: $(OBJ)Banktask.o $(OBJ)Banktaskmain.o

12 $(CC) $(CPPFLAGS) $(LDFLAGS) -o app $(OBJ)Banktask.o $(OBJ)Banktaskmain.o

13 mv app $(BIN)

14

15 $(OBJ)Banktask.o: $(SRC)Banktask.cpp

16 $(CC) $(CPPFLAGS) $(SRC)Banktask.cpp -I $(INCLUDE)

17 mv Banktask.o $(OBJ)

18

19 $(OBJ)Banktaskmain.o: $(SRC)Banktaskmain.cpp

20 $(CC) $(CPPFLAGS) $(SRC)Banktaskmain.cpp -I $(INCLUDE)

21 mv Banktaskmain.o $(OBJ)

22 valgrind:

23 valgrind --leak-check=full ./main

24 clean:

25 rm -f a.out app \*.o

26 rm -f $(OBJ)\*.o

27 rm -f $(BIN)app