

**SUBMITTED BY:**

* ROHAN MUNIR

01-134241-040

* MUHAMMAD UMER

01-134241-036

**SUBMITTED TO:**

***Sir Mohsin Javed***

**CP LAB PROJECT**

**ATM MACHINE**

**INTRODUCTION:**

This Project represents a code for a basic ATM(). The code is like a pretend ATM machine that works on your computer. It lets you do stuff you'd usually do at a real ATM, like take out money, put money in, check how much money you have, and send money to someone else. To use it you have to enter a secret PIN number just like a real ATM machine. You can select your desired actions from menu displayed on your screen.

**OBJECTIVES:**

* To let people practice basic banking without real money.
* Make it safe with a PIN.
* Make it easy to use and understand
* Allow user to perform actions such as withdrawing cash, depositing cash, checking their balance, transferring funds, and changing their PIN.
* Handles mistakes like typing the wrong PIN or trying to take out more money than you have. regenerate these lines.

**FEATURES:**

* Prior to any transaction, a secret PIN is required.
* There's a menu on the screen so you can pick what you want to do easily.
* Before allowing cash withdrawal, the system verifies if the user has sufficient funds.
* Makes sure you're putting in a reasonable amount if you want to put money in.
* You can check your balance anytime.
* You can send money to others if you have enough.
* Helps you change your secret PIN safely.
* Tells you if something goes wrong, like typing the wrong PIN or trying to do something you can't.

**EXPLANATION:**

#include<iostream>

#include<fstream>

using namespace std;

This code includes the necessary libraries for input/output operations (iostream) and file handling (fstream) and allows the use of standard library features without starting them with std::

double withdrawal(int& balance);

double deposit(int& balance);

double fundtransfer(int& balance);

double PIN(int& pin, int& your\_pin);

These functions handle bank tasks: withdrawal, deposit, fundtransfer, and PIN check, all updating the account balance or PIN.

int main() {

int pin;

char option;

int balance = 50000;

int your\_pin = 1926;

It initializes four variables: **pin** to store the user's PIN input, **option** to store the user's menu choice, **balance** to represent the user's account balance starting at 50,000, and **your\_pin** which is set to 1926, the correct PIN

do {

while (true) {

cout << "--------Welcome to Habib Bank Limited-------" << endl;

cout << endl;

cout << "Please Insert your Card" << endl;

cout << endl;

cout << "ENTER YOUR PIN" << endl;

cin >> pin;

if (pin == your\_pin) {

cout << "\t\t\t\t\t\*Your transaction has been processed\*\t\t\t\t\t" << endl;

cout << endl;

break;

}

else {

cout << "INVALID PIN. Please Try Again" << endl;

}

}

This code is part of a banking program that checks the user's PIN. It shows a welcome message from Habib Bank Limited and asks the user to insert their card. The user is then prompted to enter their PIN. If the entered PIN matches the stored PIN (**your\_pin**), a message confirms that the transaction is processed, and the loop ends. If the PIN is incorrect, an "INVALID PIN" message appears, and the user is asked to try again.

cout << "Please Select Your Mode Of Transaction" << endl;

cout << endl;

cout << "A. Withdrawal" << endl;

cout << "B. Deposit" << endl;

cout << "C. Fund Transfer" << endl;

cout << "D. Balance Inquiry" << endl;

cout << "E. Change PIN" << endl;

cout << "F. EXIT" << endl;

cout << endl;

cin >> option;

switch (option) {

case 'A':

withdrawal(balance);

break;

case 'B':

deposit(balance);

break;

case 'C':

fundtransfer(balance);

break;

case 'D':

cout << "Your Current Balance is " << balance << endl;

break;

case 'E':

PIN(pin, your\_pin);

break;

case 'F':

cout << "Thank you for using this ATM" << endl;

break;

default:

cout << "Invalid Option. Please try again." << endl;

}

cout << endl;

cout << "\t\t\t\t\t\tTHANK YOU!\t\t\t\t\t\t\t\t\t" << endl;

cout << "\t\t\t\t\t\*Please take your card\*\t\t\t\t\t" << endl;

cout << endl;

} while (option != 'F');

return 0;

}

This section of code presents a menu that allows the user to select their desired banking transaction. After displaying the options, it allows the user to input their choice. Based on the input, it executes the corresponding transaction function: withdrawal, deposit, fund transfer,balance inquiry, or changing the PIN. If the user selects 'EXIT', the program ends. If an invalid option is entered, it allows the user to try again. The loop continues until the user selects 'EXIT'.

**WITHDRAWAL:**

double withdrawal(int& balance) {

int amount\_1;

cout << "Please enter the amount for the transaction" << endl;

cin >> amount\_1;

cout << endl;

if (amount\_1 > 0 && amount\_1 <= balance) {

cout << "TRANSACTION SUCCESSFUL" << endl;

cout << "Please Collect Your Cash" << endl;

balance -= amount\_1;

cout << "Your Remaining Balance is " << balance << endl;

}

else {

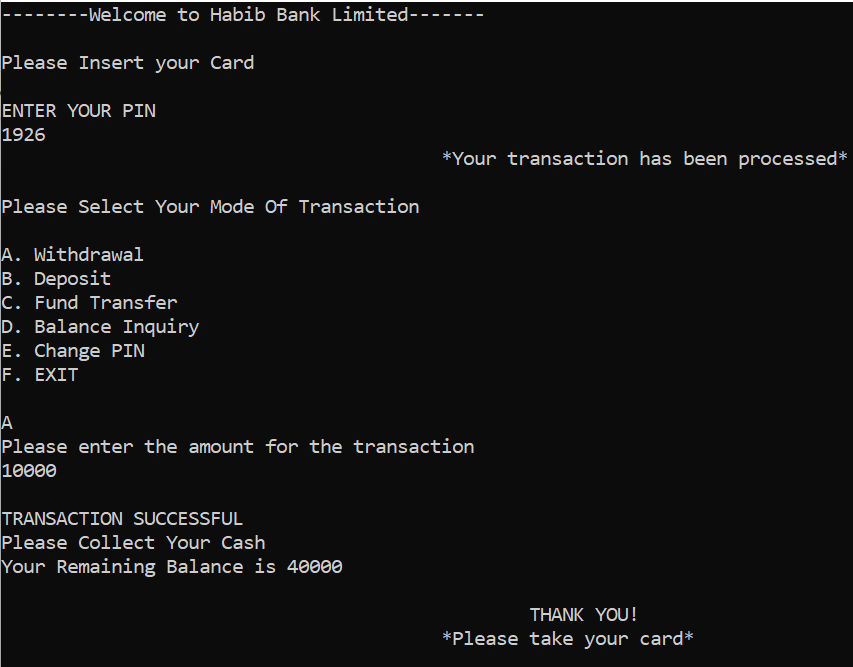
cout << "INSUFFICIENT FUNDS" << endl;

}

return balance;

}

This function handles withdrawals from the account balance. It prompts the user to input the withdrawal amount, then checks if the amount is within the available balance. If so, it processes the transaction, deducts the amount from the balance, and displays the new balance. If the amount exceeds the balance or is invalid, it notifies the user of insufficient funds. Finally, it returns the updated balance.

****

**DEPOSIT:**

double deposit(int& balance) {

int amount\_2;

cout << "Enter the amount you want to deposit" << endl;

cin >> amount\_2;

cout << endl;

if (amount\_2 > 0) {

cout << "DEPOSIT SUCCESSFUL" << endl;

balance += amount\_2;

cout << "Your new balance is " << balance << endl;

}

else {

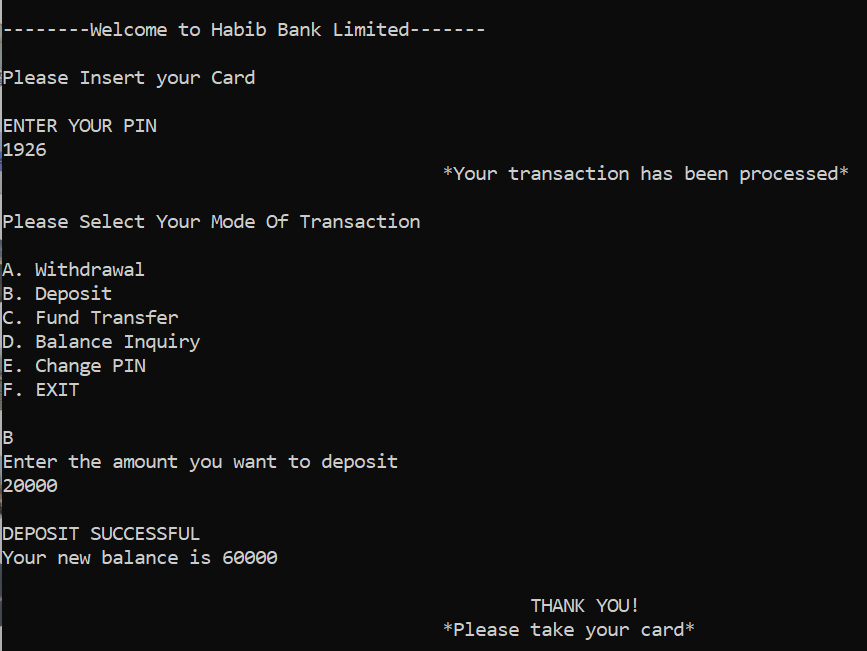
cout << "INVALID DEPOSIT. Please Try Again" << endl;

}

return balance;

}

This function manages deposits into the account balance. It prompts the user to input the deposit amount, ensuring it's a positive value. If valid, it adds the amount to the balance and displays the new balance, confirming the successful deposit. If the input is invalid (zero or negative), it notifies the user and asks them to try again. Finally, it returns the updated balance.

****

**FUND TRANSFER:**

double fundtransfer(int& balance) {

int amount\_3;

cout << "Enter the amount you want to transfer" << endl;

cin >> amount\_3;

cout << endl;

if (amount\_3 > 0 && amount\_3 <= balance) {

cout << "Your amount has been transferred" << endl;

balance -= amount\_3;

cout << "Your remaining balance is " << balance << endl;

}

else {

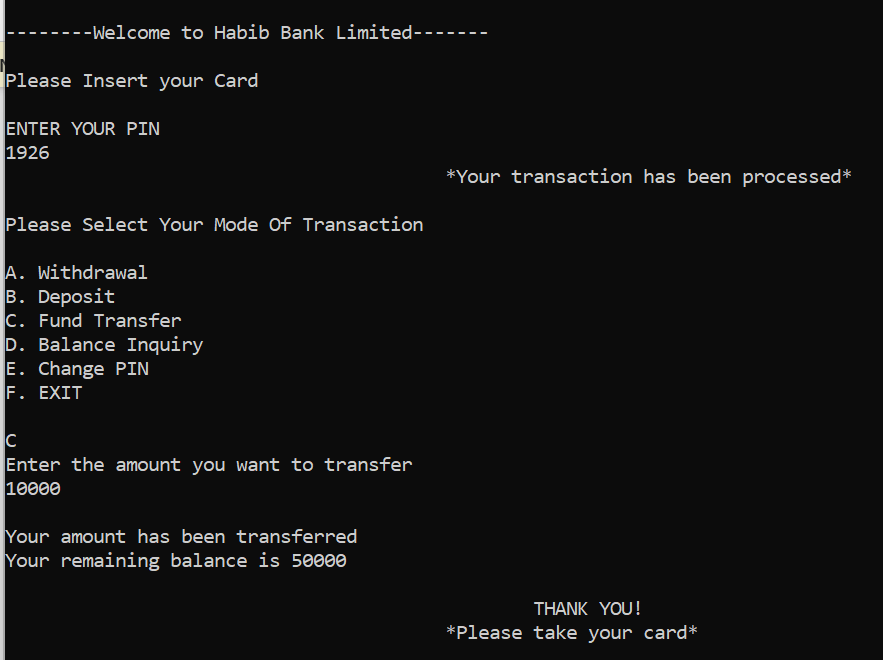
cout << "INVALID TRANSFER" << endl;

}

return balance;

}

This function facilitates fund transfers from the account balance. It prompts the user to input the transfer amount, verifying it's within the available balance. If valid, it deducts the amount from the balance, confirms the transfer, and displays the new balance. If the transfer amount exceeds the balance or is invalid, it notifies the user accordingly. Finally, it returns the updated balance.

****

**PIN:**

double PIN(int& pin, int& your\_pin) {

int old\_pin, new\_pin, confirm\_pin;

cout << "Enter Your Previous PIN" << endl;

cin >> old\_pin;

if (old\_pin == your\_pin) {

cout << "Enter your new PIN" << endl;

cin >> new\_pin;

cout << "Confirm your new PIN" << endl;

cin >> confirm\_pin;

if (new\_pin == confirm\_pin) {

your\_pin = new\_pin;

cout << "Your PIN has been changed successfully to " << new\_pin << endl;

}

else {

cout << "New PIN and confirm PIN do not match. Please try again." << endl;

}

}

else {

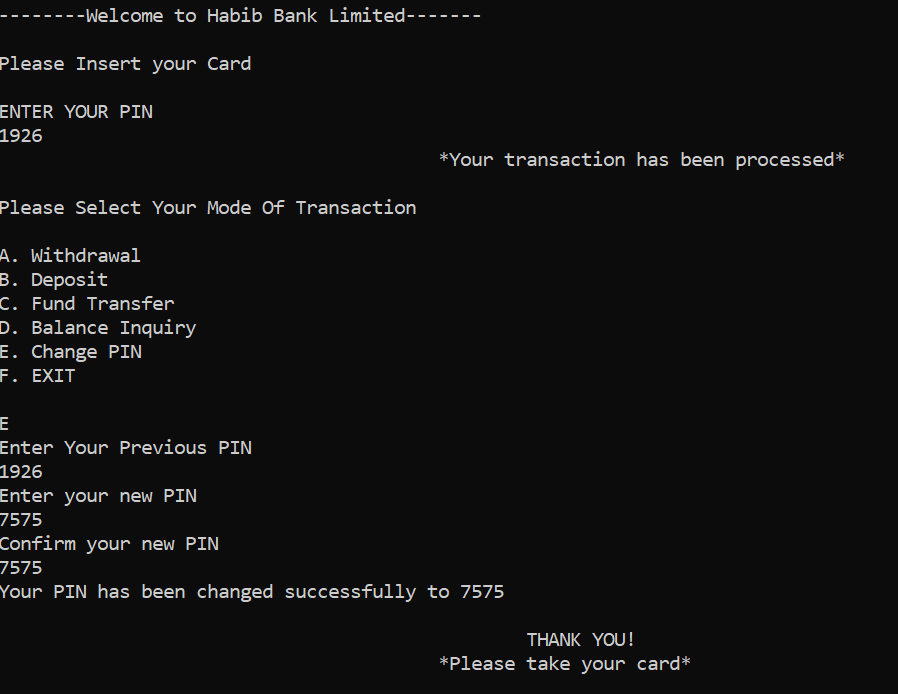
cout << "OLD PIN INCORRECT" << endl;

}

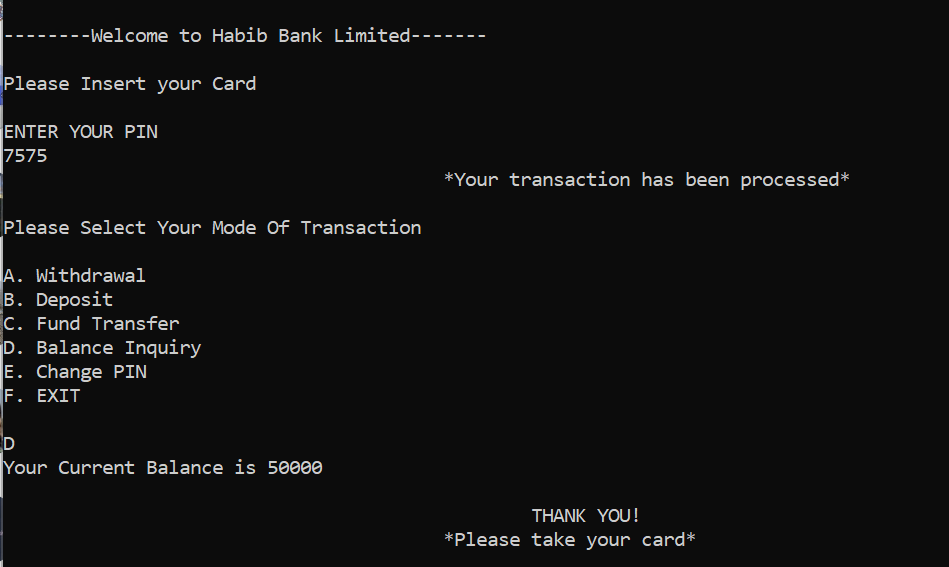
return your\_pin;

}

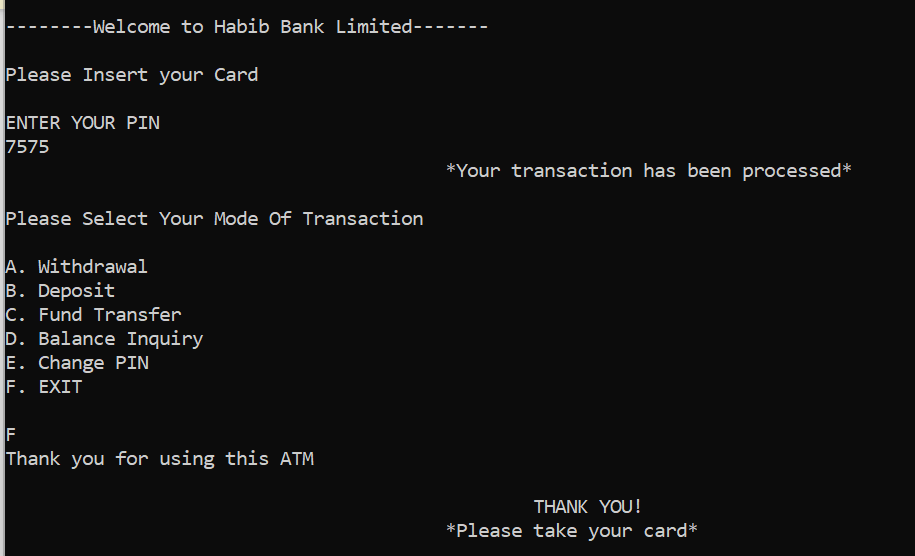
This function manages the changing of the PIN. It first prompts the user to input their old PIN for verification. If the old PIN matches the stored PIN (**your\_pin**), it allows the user to input a new PIN and confirm it. If the new PIN matches the confirmation, it updates **your\_pin** with the new PIN and confirms the change. If the old PIN is incorrect, it notifies the user accordingly. Finally, it returns the updated PIN.

****

**BALANCE INQUIRY:**

****

**EXIT:**

****

**FILE HANDLING:**

The program reads the current balance from a file named balance.txt when it starts. If the file cannot be opened, it sets a default balance, such as 50,000. Similarly, the program reads the PIN from a file named pin.txt. If the file cannot be opened, a default PIN, like 1926, is used. After any transaction that changes the balance, such as a withdrawal, deposit, or fund transfer, the new balance is written back to balance.txt to save the updated balance. When the user changes their PIN, the new PIN is written to pin.txt to save the updated PIN.

**CONCLUSION:**

This pretend ATM is a simple way for people to practice using basic banking stuff on their computers. It's not real, but it works like a real ATM. It's good for learning because it shows how computers can handle things like taking out money and keeping track of accounts. Overall, it's a good way for beginners to learn about how programming works in the real world.