CSE/ECE/ISE Department – Faculty of Engineering - MSA
Spring 2024
COM265 Computer Programming II
Assignment/Lab No ….
Course Instructor Dr. Ahmed AlAnany
TA. Eng:hussein

Name:william tamer	**ID:232455**
Name:Boula kameel	**ID:233389**

Account.h

```
#pragma once
#include <string>
#include <vector>
#include "transaction.h"
#include <iostream>
using namespace std;
class Account
{
private:
    string CustomerName;
    int MobileNumber;
    double Balance;
    int AccountNumber;
    static int NumAcc;
    vector<transaction>Alltransactions;

public:
    Account() {}
    int  getAccountNumber() {
        return AccountNumber;
    }
    void setCustomerName(string name) {
        this->CustomerName = name;
    }
    string getCustomerName() {
        return CustomerName;
    }
    void setMobileNumber(int M_N) {
        this->MobileNumber = M_N;
    }
```

```
int getMobileNumber() {
    return MobileNumber;
}
void setBalance(double a) {
    this->Balance = a;
}
double getBalance() {
    return Balance;
}

vector<transaction> getTransactions(){
    return Alltransactions;
}
Account(string CustomerName, int MobileNumber, double  Balance) {
    this->CustomerName = CustomerName;
    this->MobileNumber = MobileNumber;
    this->Balance = Balance;
    NumAcc++;
    AccountNumber = NumAcc;
}
void DepositeAmount(double am) {
    Balance = Balance + am;
    transaction trans;
    trans.setAmount(am);
    trans.setTransactionType('D');
    trans.setAccountNumber(getAccountNumber());
    Alltransactions.push_back(trans);
}
void withdrawAmount(double am) {
    Balance = Balance - am;
    transaction trans;
    trans.setAmount(am);
    trans.setTransactionType('w');
    trans.setAccountNumber(getAccountNumber());
    Alltransactions.push_back(trans);
}



void DisplayAllTransactions() {
    for (auto i : Alltransactions) {
        i.Display();
        cout << "=========================================\n";
    }
}

void AccountDisplay(){
    cout << "Customer Account Number: " << CustomerName << endl;
    cout << "CustomerName: " << MobileNumber << endl;
    cout << "Customer Mobile Number: " << AccountNumber<< endl;
    cout << "Customer Balance: " << setprecision(15) << Balance<<endl;
```

```
        cout << "=========================================\n";
    }
};
int Account:: NumAcc = 10000;
```

Transaction.h

```
#pragma once
#include"Account.h"
#include <ctime>
#include <cstring>
#include <iostream>
#include<chrono>
using namespace std;

time_t t;

class transaction {
private:
    int TransactionId;
    static int num;
    char TransactionType;
    double amount;
    int AccountNumber;
    char date [90] ;
public:
    transaction() {
        num++;
        TransactionId = num;
        t = time(0);
        const char* format = "%Y-%m-%d %H:%M:%S";
        struct tm* timeinfo = localtime(&t);
        strftime(date, sizeof(date), format, timeinfo);
    }
    void setTransactionId(int t) {
        TransactionId = t;
    }
    int getTransactionId() {
        return TransactionId;
    }
    void setTransactionType(char type) {
        TransactionType = type;
    }

    char getTransactionType() {
        return TransactionType;
    }

    void setAmount(double amount) {
        this->amount = amount;
    }
```

```
    double getAmount() {
        return amount;
    }

    void setAccountNumber(int accNum) {
        AccountNumber = accNum;
    }

    string getDate(){
        return date;
    }
    void Display() {
        cout << "Transaction ID: " << TransactionId << endl;
        cout << "Transaction Type: " << TransactionType << endl;
        cout << "Account Number: " << AccountNumber<< endl;
        cout << "Amount: " << amount << endl;
        cout << "Date: " << date << endl;
    }

};
int transaction::num = 0;
```

Search.h

```
#include "Account.h"
#include <iostream>
#include <string>
#include <vector>

using namespace std;

class Search{
    public:
    vector<Account>::iterator searchByAccountNum(int id, vector<Account>&
Accounts){
        vector<Account>::iterator it;
        bool found = false;
        it = Accounts.begin();
        for (auto account : Accounts) {
            if (account.getAccountNumber() == id) {
                found = true;
                break;
            }
            it++;
        }
        if (!found) {
            cout << "Account is not found" << endl;
        }
        else{
            return it;
```

```
        }
     }
    void searchByDate(char x [90], vector<Account>::iterator y){
        bool found = false;
        for (auto transaction : y->getTransactions()){
            if(transaction.getDate().find(x) != -1){
                transaction.Display();
                found = true;
                break;
            }
        }
        if (!found){
            cout<<"There is no transaction that occured in that date."<<endl;
        }
    }
};
```

Loan.h

```
#pragma once
#include<string>
#include <iostream>
class Loan
{
    public:
    int loanId;
    int accountNumber;
    double loanAmount;
    string loanType;
    int intrestRate;
    int installmentAmount;


Loan(int Iid, int accNumber, double amount, string type, double rate){
        loanId=Iid;
        accountNumber=accNumber;
        loanAmount=amount;
        loanType=type;
        intrestRate=rate;
        installmentAmount=calculateInstallment();
    }
     double calculateInstallment(){
        double totalAmount=loanAmount+(loanAmount*intrestRate/100);
        return totalAmount/12;

     }
     void payInstallment(){
        if(loanAmount>0){
            loanAmount-=installmentAmount;
```

```
            cout<<"installement is successfully paid/Remaining Loan
amount$"<<fixed<<setprecision(2)<<loanAmount<<endl;}
            else{
                cout<<"loan not paid"<<endl;
            }

        }
        void displayLoanInfo(){
        cout <<"Loan Id:" <<loanId <<endl;
        cout<<"Account Number:"<<accountNumber<<endl;
        cout<<"Loan Amount:$"<<fixed <<setprecision(2)<<loanAmount<<endl;
        cout<<"Loan Type:"<<loanType<<endl;
        cout<<"intrest Rate:"<<fixed<<setprecision(2)<<intrestRate<<"%"<<endl;
        cout<<"monthly Installment
Amount:$"<<fixed<<setprecision(2)<<installmentAmount<<endl;
        }

     };
```

Validations.h

```
class Validation{
    public:
    bool validateName(string name){
        for(char c:name){
            if(!isalpha(c) && c!=' '){
                return false;
            }
        }
        return true;
    }
    bool validateMobileNumber(string mobileNumber){
        if(mobileNumber.length()!=10){
            return false;
        }
        for(char c:mobileNumber){
            if(!isdigit(c)){
                return false;
            }

        }
        return true;
    }
    bool validateNumericInput(string input){
        for(char c:input){
            if(!isdigit(c)&&c!='.'){
                return false;
            }
        }
        return true;
    }
```

```
    bool validateChoice(int choice,int min,int max){
        return choice>=min&&choice<=max;
    }




};
```

Main.cpp

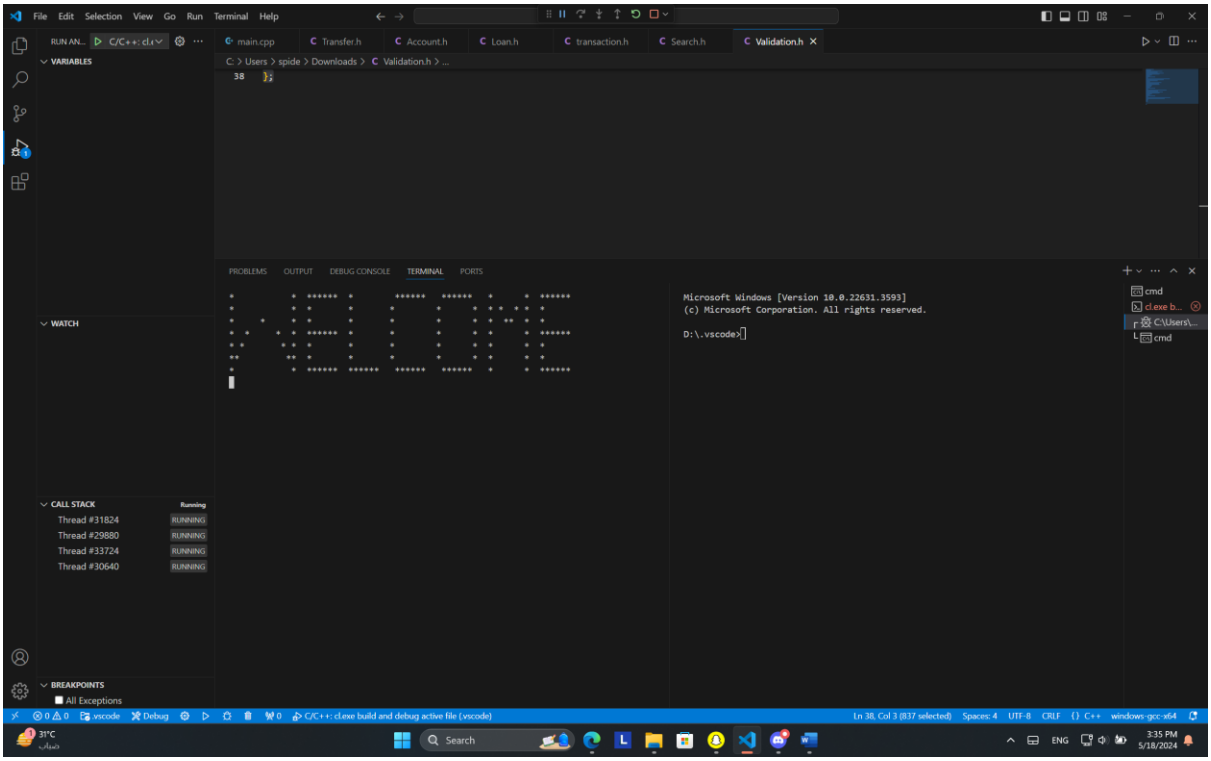
```
#include <iostream>
#include <iomanip>
#include <stdlib.h>
#include "Account.h"
#include <string>
#include <vector>
#include <limits>
#include"transaction.h"
#include"Loan.h"
#include"Search.h"
#include"Transfer.h"
#include "Validation.h"
#include<chrono>

using namespace std;
class Menu {
public:
    void mainmenu() {
        cin.ignore();
        cout << "\t=================+==================\n";
        cout << "\t|               Menu                |\n";
        cout << "\t=================+==================\n";
        cout << "\t 1-    Open Account \n\t 2-    Deposite Account \n\t
3-    withdraw Amount \n\t ";
        cout << "4-    Display ALL Account \n\t 5-    Display Transaction \n\t
6-    Delete Account \n\t";
        cout << " 7-    Loan  \n\t 8-    Search \n\t 9-   Transfer Amount
\n\t";
        cout << " 10-    Exit \n\t";

    }
    Menu() {

    }

};
int main()
{
    cout << setw(6) << left << "*           *" << "  ****** " << " *      " <<
"  ****** " << "  ******  " << " *      *" << "  ****** " << endl;
```

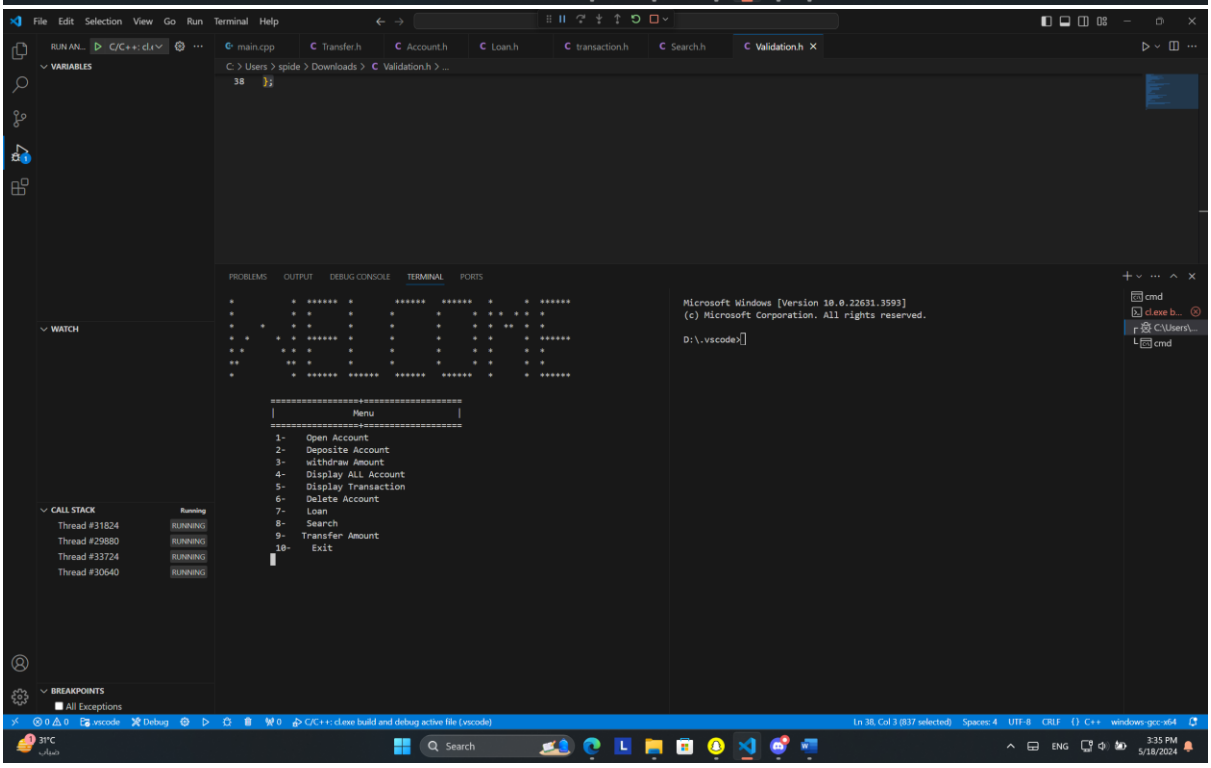
```
    cout << setw(6) << left << "*           *" << "  *      " << " *      " <<
" *       " << " *      * " << " * *  * *" << "  *      " << endl;
    cout << setw(6) << left << "*     *     *" << "  *      " << " *      " <<
" *       " << " *      * " << " *  **  *" << "  *      " << endl;
    cout << setw(6) << left << "*  *     *  *" << "  ****** " << " *      " <<
" *       " << " *      * " << " *      *" << "  ****** " << endl;
    cout << setw(6) << left << "* *       * *" << "  *      " << " *      " <<
" *       " << " *      * " << " *      *" << "  *      " << endl;
    cout << setw(6) << left << "**         **" << "  *      " << " *      " <<
" *       " << " *      * " << " *      *" << "  *      " << endl;
    cout << setw(6) << left << "*           *" << "  ****** " << " ****** " <<
"  ****** " << "  ******  " << " *      *" << "  ****** " << endl;
    Menu MENU1;
    vector<Account>Accounts;
    vector<Account>::iterator it;
    int mainAccNum;
    int menu;
    bool t = true;

    while (t) {
        MENU1.mainmenu();
        cin >> menu;
        system("cls");
        switch (menu) {
        case (1): {
            cout << "Enter Your Account Number or Type (0) to create a new
Account: ";
            cin >> mainAccNum;
            if (mainAccNum == 0){
                string name;
                int mob;
                double balance;
                cout << "Enter your Name: ";
                cin.ignore();
                getline(cin, name);
                cout << "Enter your mobile number: ";
                cin >> mob;
                cout << "Enter your Balance: ";
                cin >> balance;
                Account account(name, mob, balance);
                Accounts.push_back(account);
                system("cls");
            }
            else {
                Search search;
                it = search.searchByAccountNum(mainAccNum, Accounts);
            }

            break;
        }
```

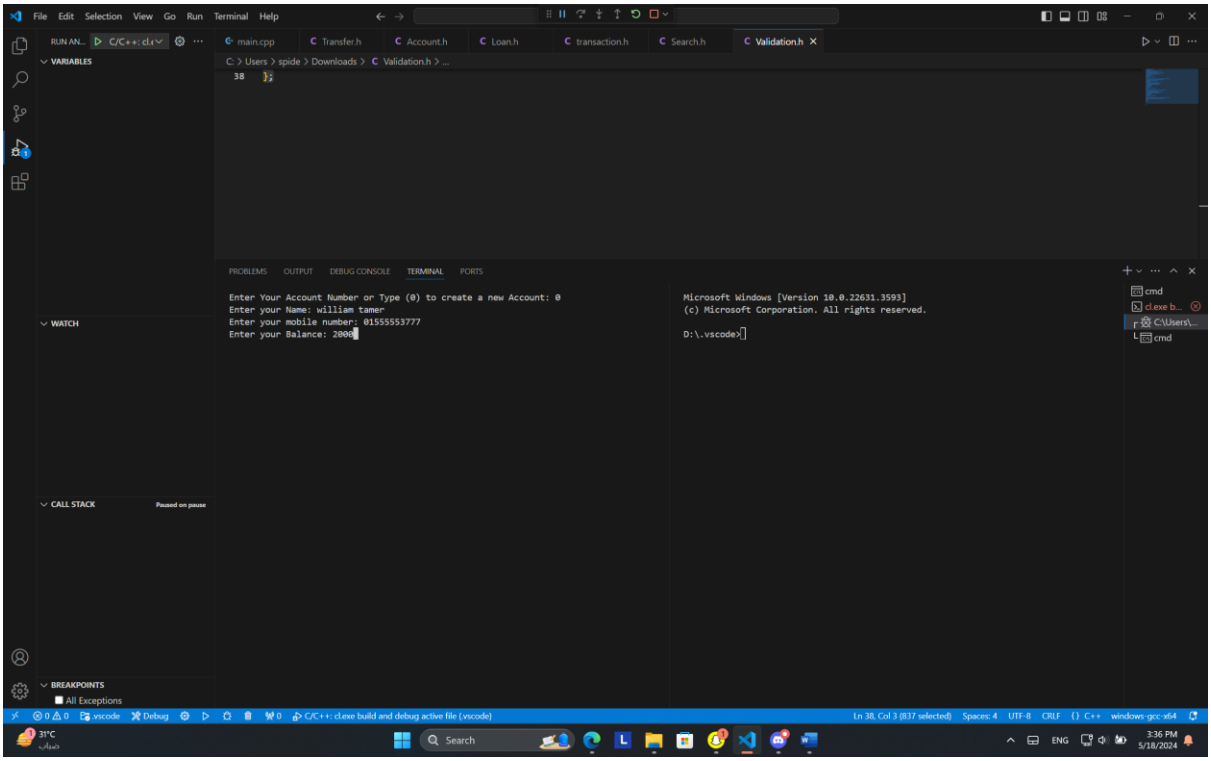
```
        case (2): {
            double amount;
            cout << "Enter Amount to Deposite: ";
            cin >> amount;
            it->DepositeAmount(amount);
            break;
        }

        case 3:
            double amount;
            cout << "Enter Amount to Withdraw: ";
            cin >> amount;
            it->withdrawAmount(amount);
            break;

        case (4): {
            for (auto acc : Accounts) {
                acc.AccountDisplay();
            }
            cout << "Type ok to continue: ";
            string dummy;
            cin >> dummy;
            system("cls");
            break;

        }


        case (5): {
            it->DisplayAllTransactions();
            cout << "Type ok to continue: ";
            string dummy;

            cin >> dummy;
            system("cls");
            break;
        }
        case (6):{

            break;
        case (7):{
           int loanId, accountNumber;
    double loanAmount, interestRate;
    string loanType;

    cout << "Enter Loan ID: ";
    cin >> loanId;

    cout << "Enter Account Number: ";
    cin >> accountNumber;
```

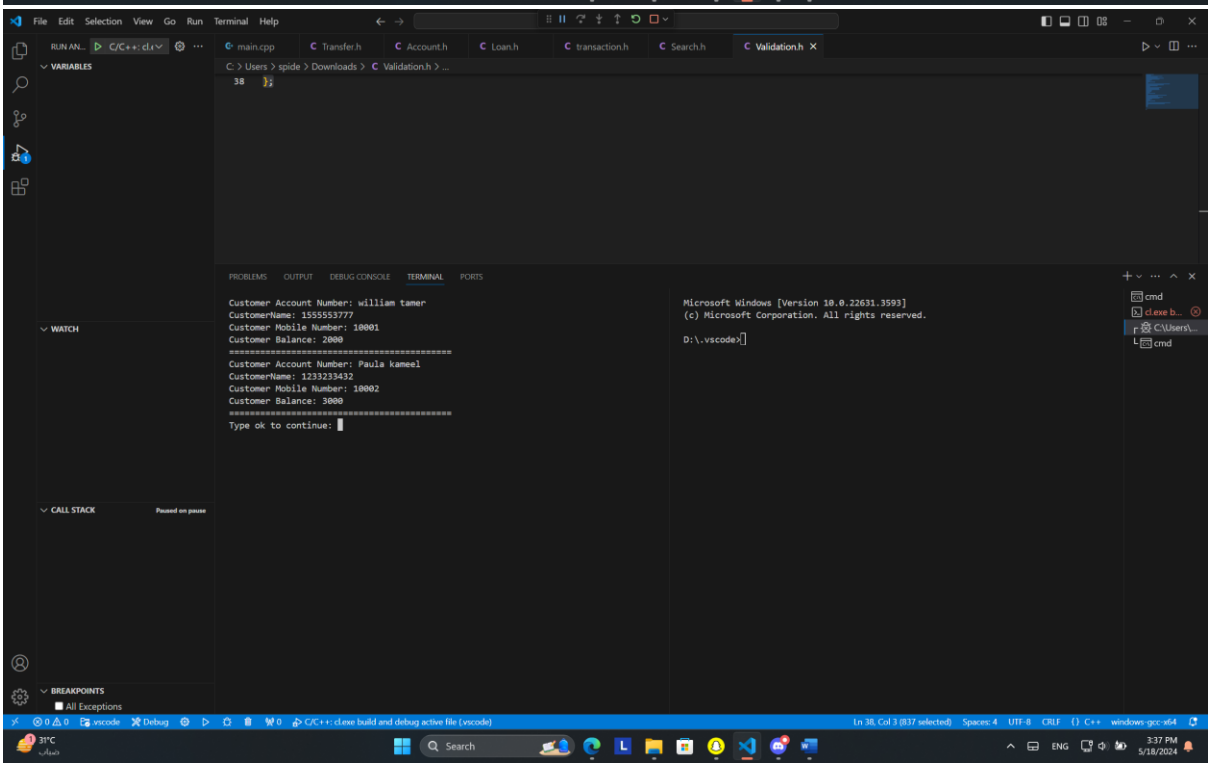
```
    cout << "Enter Loan Amount: $";
    cin >> loanAmount;

    cout << "Enter Loan Type: ";
    cin.ignore();
    getline(cin, loanType);

    cout << "Enter Interest Rate (%): ";
    cin >> interestRate;


    Loan userLoan(loanId, accountNumber, loanAmount, loanType, interestRate);


    userLoan.displayLoanInfo();


    userLoan.payInstallment();

}
            break;
        case (8):{
            cout<<"Type 1 to search for an Account"<<endl;
            cout<<"Type 2 to search for a transaction"<<endl;
            int option;
            cin>>option;
            system("cls");
            Search search;
            if (option == 1){
                    cout<<"Type 1 to search by Account number"<<endl;
                    cout<<"Type 2 to search by Account amount range"<<endl;
                    int option2;
                    cin>>option2;
                    system("cls");
                    switch (option2){
                        case(1):{
                            cout << "Enter Your Account Number: ";
                            int AccNum;
                            cin>>AccNum;
                            vector<Account>::iterator x;
                            x = search.searchByAccountNum(AccNum, Accounts);
                            x->AccountDisplay();
                            break;
                        }
                        case(2):{

                            break;
                        }
                    }

                }
```

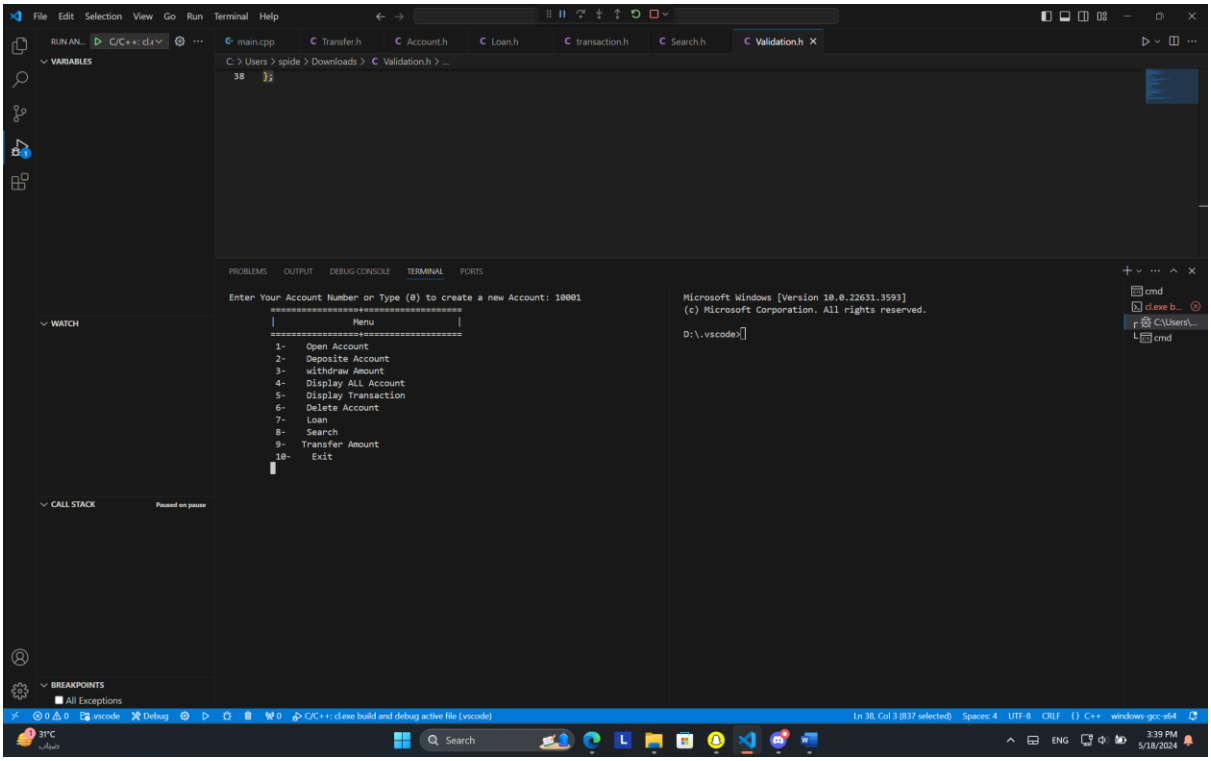
```
            else if(option == 2){
                    cout<<"Type 1 to search by Date"<<endl;
                    cout<<"Type 2 to search by TransactionID"<<endl;
                    int option2;
                    cin>>option2;
                    switch (option2){
                        case(1):{
                            char x [90];
                            cout<<"Enter the date of the transaction: ";
                            cin>>x;
                            search.searchByDate(x, it);
                            break;
                        }
                        case(2):{
                            break;
                        }
                    }
                }
            cout << "Type ok to continue: ";
            string dummy;
            cin >> dummy;
            system("cls");
            break;
        }
        case (9):{
            double amount;
            cout << "Enter Amount to transfer: ";
            cin >> amount;
            it->transferAmount(amount);
            break;
            break;
        }
        case (10): {
            t = false;
            break;
        }
        default:
            cout << "choose a valid choice" << endl;

        }
    }

}
```

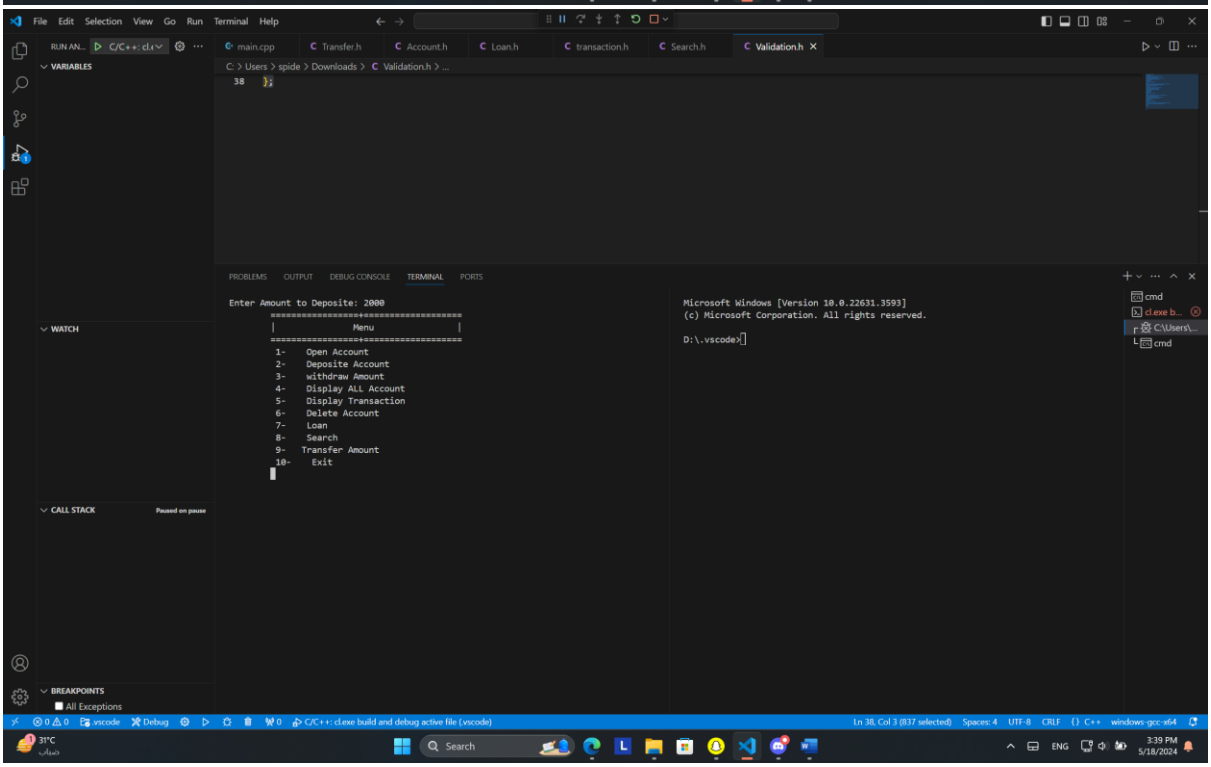

File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
C: > Users > spide > Downloads > Validation.h > ...
38 };
VARIABLES
WATCH
CALL STACK
Thread #31824 RUNNING
Thread #29880 RUNNING
Thread #33724 RUNNING
Thread #30640 RUNNING
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>

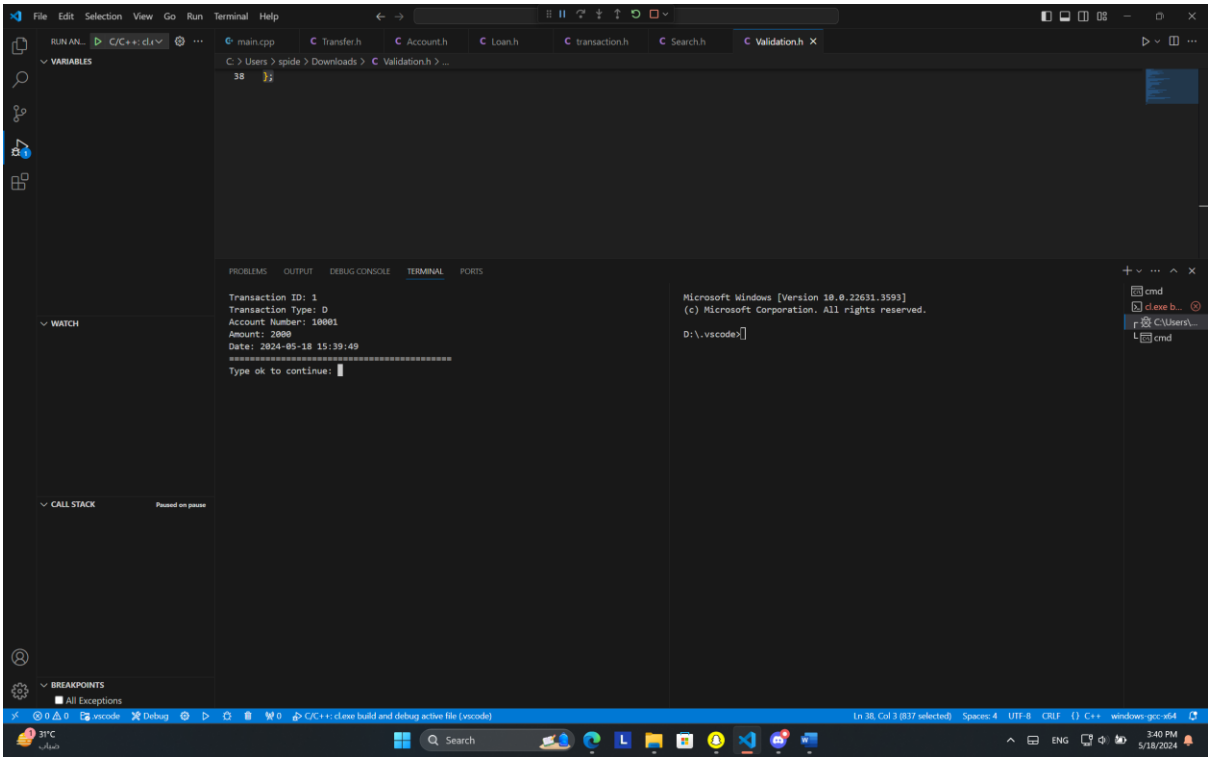

File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
C: > Users > spide > Downloads > Validation.h > ...
38 };
VARIABLES
WATCH
CALL STACK
Thread #31824 RUNNING
Thread #29880 RUNNING
Thread #33724 RUNNING
Thread #30640 RUNNING
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
Menu
1- Open Account
2- Deposite Account
3- withdraw Amount
4- Display ALL Account
5- Display Transaction
6- Delete Account
7- Loan
8- Search
9- Transfer Amount
10- Exit

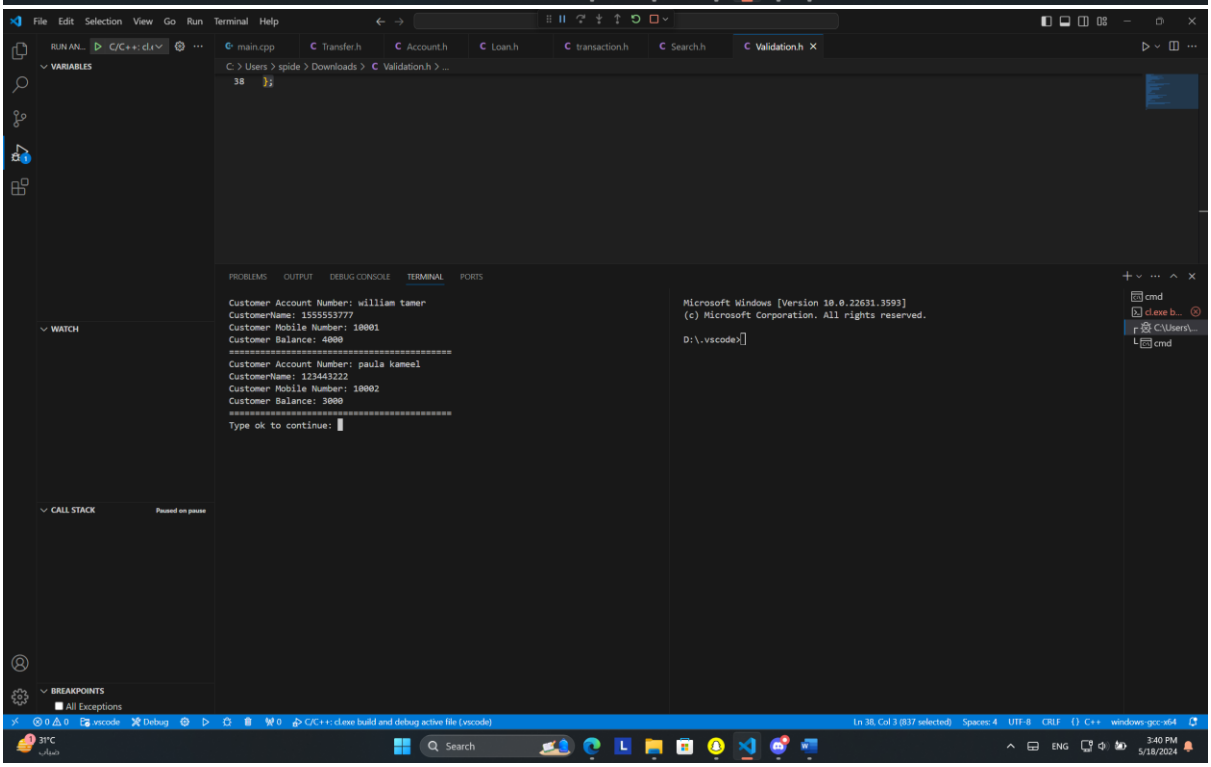

File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
C: > Users > spide > Downloads > Validation.h > ...
38 };
VARIABLES
WATCH
CALL STACK
Paused on pause
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter Your Account Number or Type (0) to create a new Account: 0
Enter your Name: william tamer
Enter your mobile number: 01555553777
Enter your Balance: 2000
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
cmd
cl.exe b...
C:\Users\...
cmd
0 0 vscode Debug C/C++: cl.exe build and debug active file (vscode)
Ln 38, Col 3 (837 selected) Spaces: 4 UTF-8 CRLF C++ windows-gcc-x64
31°C
Search
ENG
3:36 PM
5/18/2024

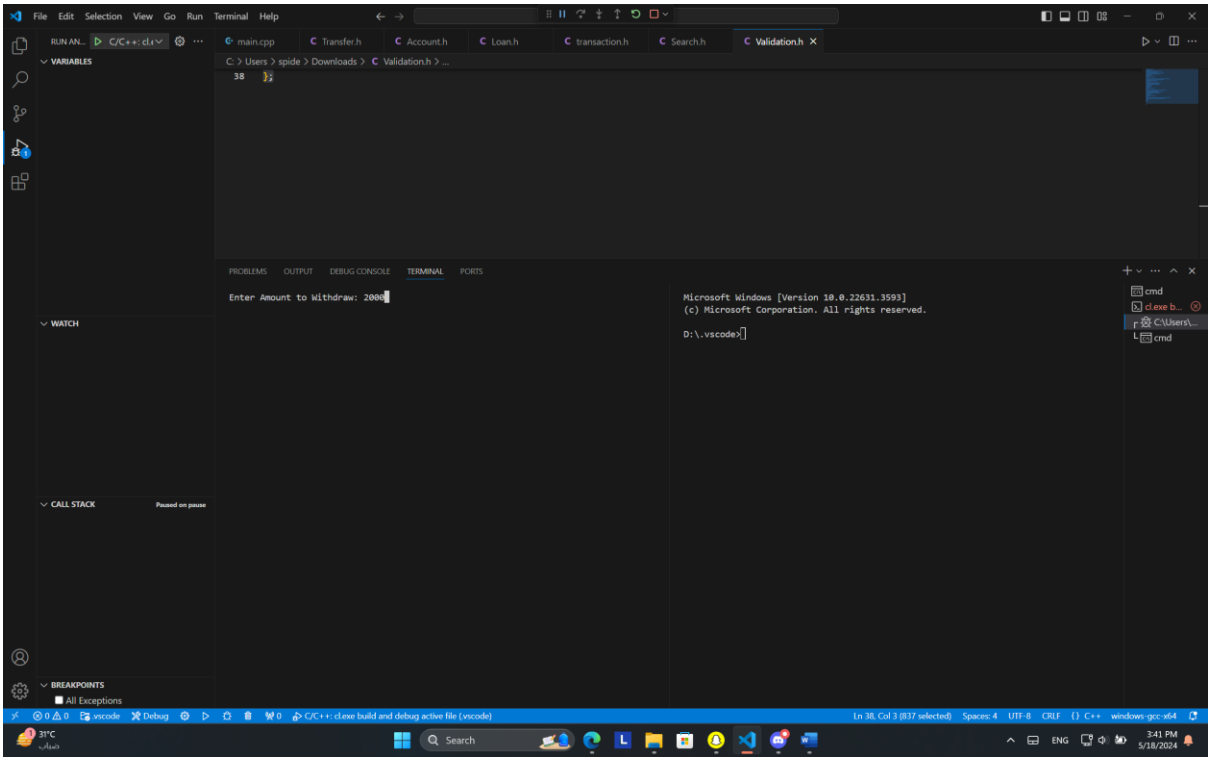

File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
C: > Users > spide > Downloads > Validation.h > ...
38 };
VARIABLES
WATCH
CALL STACK
Paused on pause
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Customer Account Number: william tamer
CustomerName: 1555553777
Customer Mobile Number: 10001
Customer Balance: 2000
==========
Customer Account Number: Paula kameel
CustomerName: 1233233432
Customer Mobile Number: 10002
Customer Balance: 3000
==========
Type ok to continue:
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
cmd
cl.exe b...
C:\Users\...
cmd
0 0 vscode Debug C/C++: cl.exe build and debug active file (vscode)
Ln 38, Col 3 (837 selected) Spaces: 4 UTF-8 CRLF C++ windows-gcc-x64
31°C
Search
ENG
3:37 PM
5/18/2024

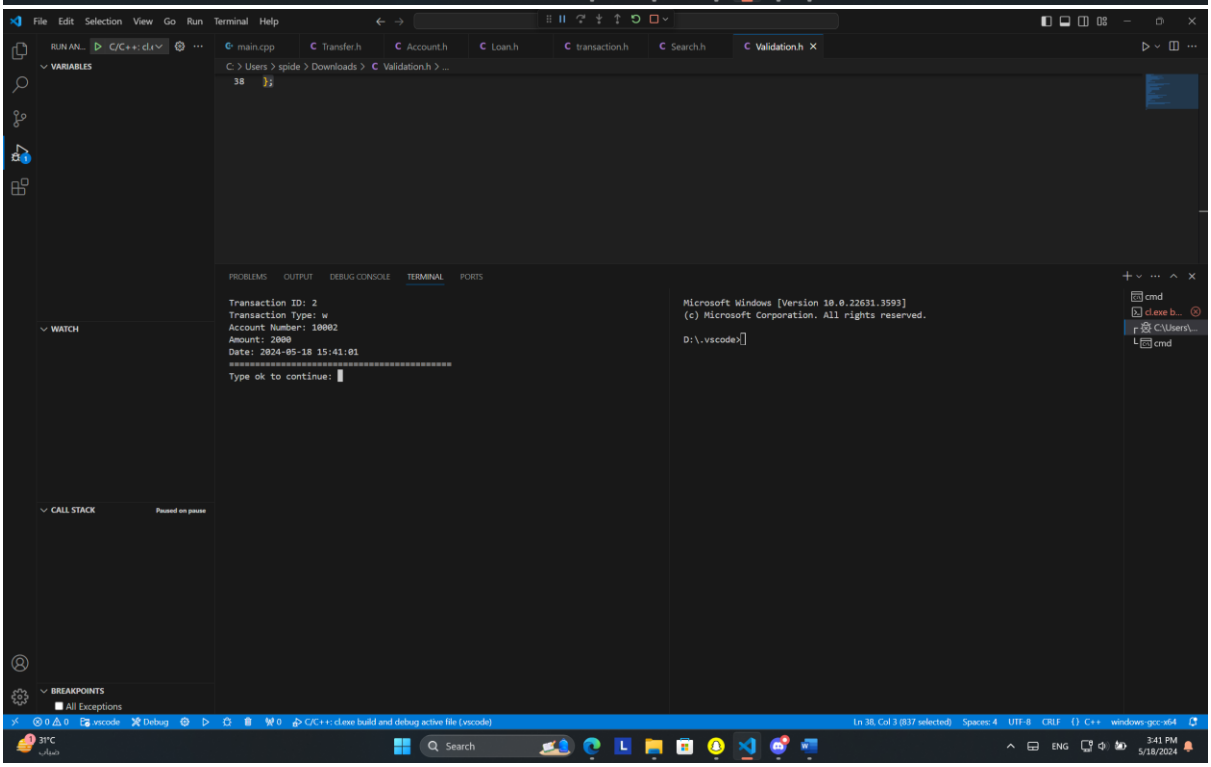

File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
C: > Users > spide > Downloads > Validation.h > ...
38 };
VARIABLES
WATCH
CALL STACK
Paused on pause
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter Your Account Number or Type (0) to create a new Account: 10001
| Menu |
1- Open Account
2- Deposite Account
3- withdraw Amount
4- Display ALL Account
5- Display Transaction
6- Delete Account
7- Loan
8- Search
9- Transfer Amount
10- Exit
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
cmd
cl.exe b...
C:\Users\...
cmd
Ln 38, Col 3 (837 selected) Spaces: 4 UTF-8 CRLF C++ windows-gcc-x64
31°C
Search
ENG
3:39 PM
5/18/2024

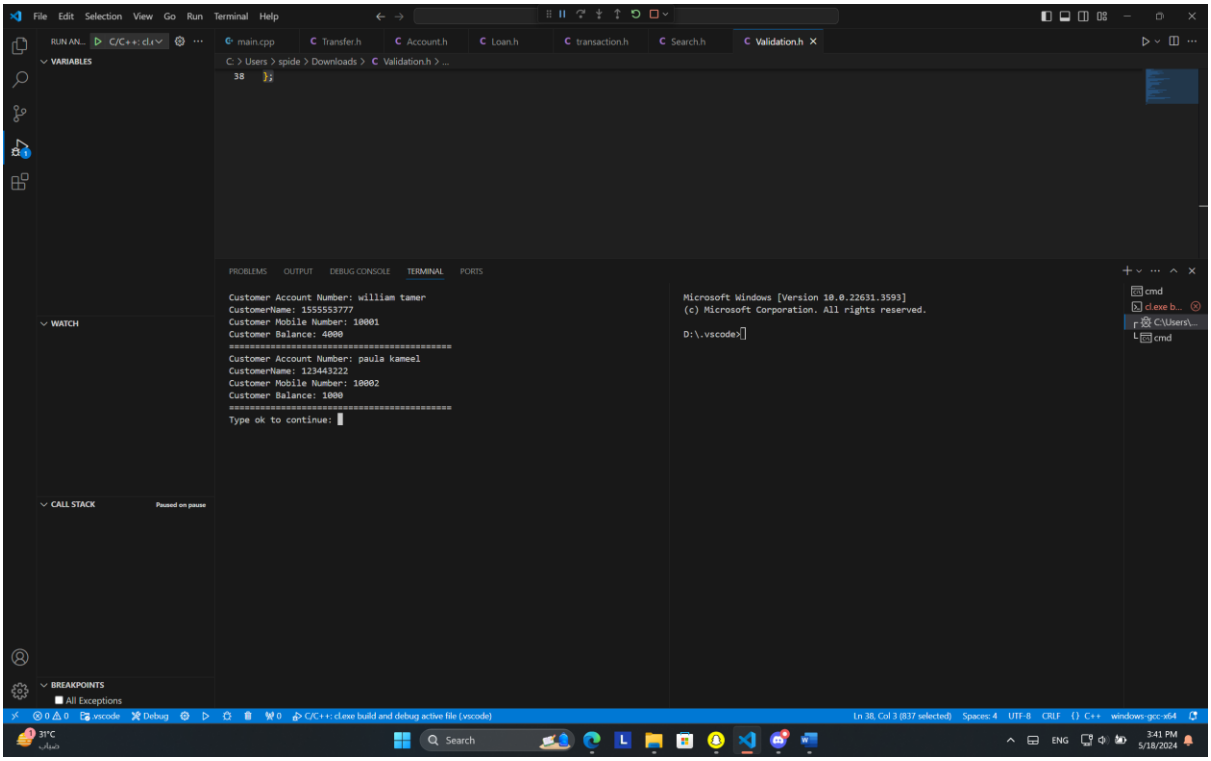

File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
C: > Users > spide > Downloads > Validation.h > ...
38 };
VARIABLES
WATCH
CALL STACK
Paused on pause
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter Amount to Deposite: 2000
| Menu |
1- Open Account
2- Deposite Account
3- withdraw Amount
4- Display ALL Account
5- Display Transaction
6- Delete Account
7- Loan
8- Search
9- Transfer Amount
10- Exit
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
cmd
cl.exe b...
C:\Users\...
cmd
Ln 38, Col 3 (837 selected) Spaces: 4 UTF-8 CRLF C++ windows-gcc-x64
31°C
Search
ENG
3:39 PM
5/18/2024

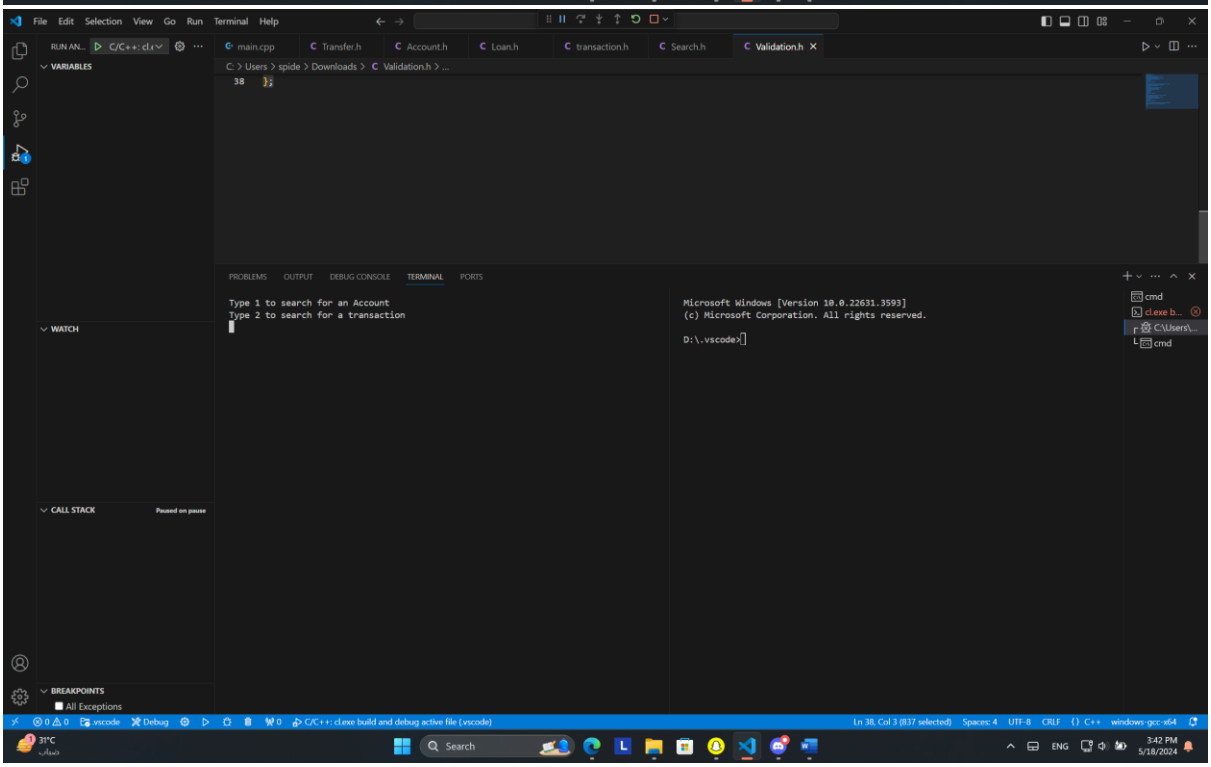

File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
C: > Users > spide > Downloads > Validation.h > ...
38 };
VARIABLES
WATCH
CALL STACK
Paused on pause
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Transaction ID: 1
Transaction Type: D
Account Number: 10001
Amount: 2000
Date: 2024-05-18 15:39:49
==========
Type ok to continue:
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
cmd
cl.exe b...
C:\Users\...
cmd
Ln 38, Col 3 (837 selected) Spaces: 4 UTF-8 CRLF C++ windows-gcc-x64
31°C
Search
ENG
3:40 PM
5/18/2024

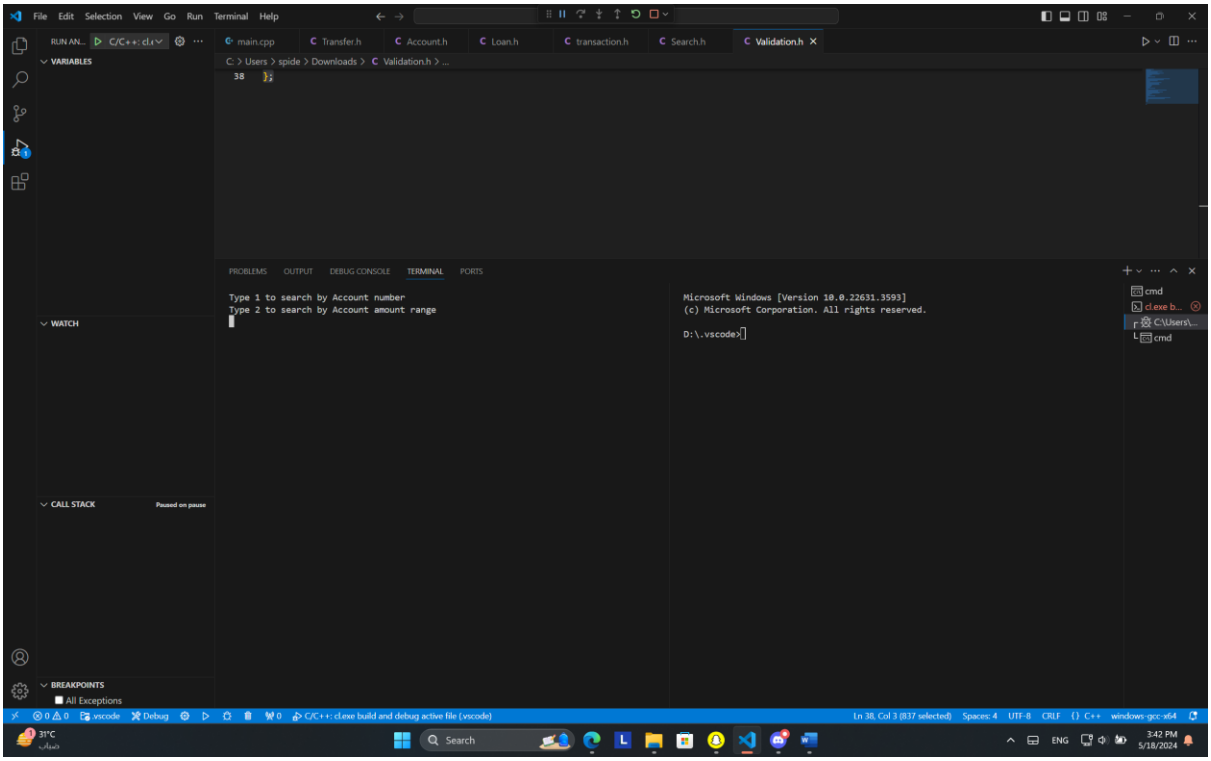

File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
C: > Users > spide > Downloads > Validation.h > ...
38 };
VARIABLES
WATCH
CALL STACK
Paused on pause
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Customer Account Number: william tamer
CustomerName: 1555553777
Customer Mobile Number: 10001
Customer Balance: 4000
==========
Customer Account Number: paula kameel
CustomerName: 123443222
Customer Mobile Number: 10002
Customer Balance: 3000
==========
Type ok to continue:
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
cmd
cl.exe b...
C:\Users\...
cmd
Ln 38, Col 3 (837 selected) Spaces: 4 UTF-8 CRLF C++ windows-gcc-x64
31°C
Search
ENG
3:40 PM
5/18/2024

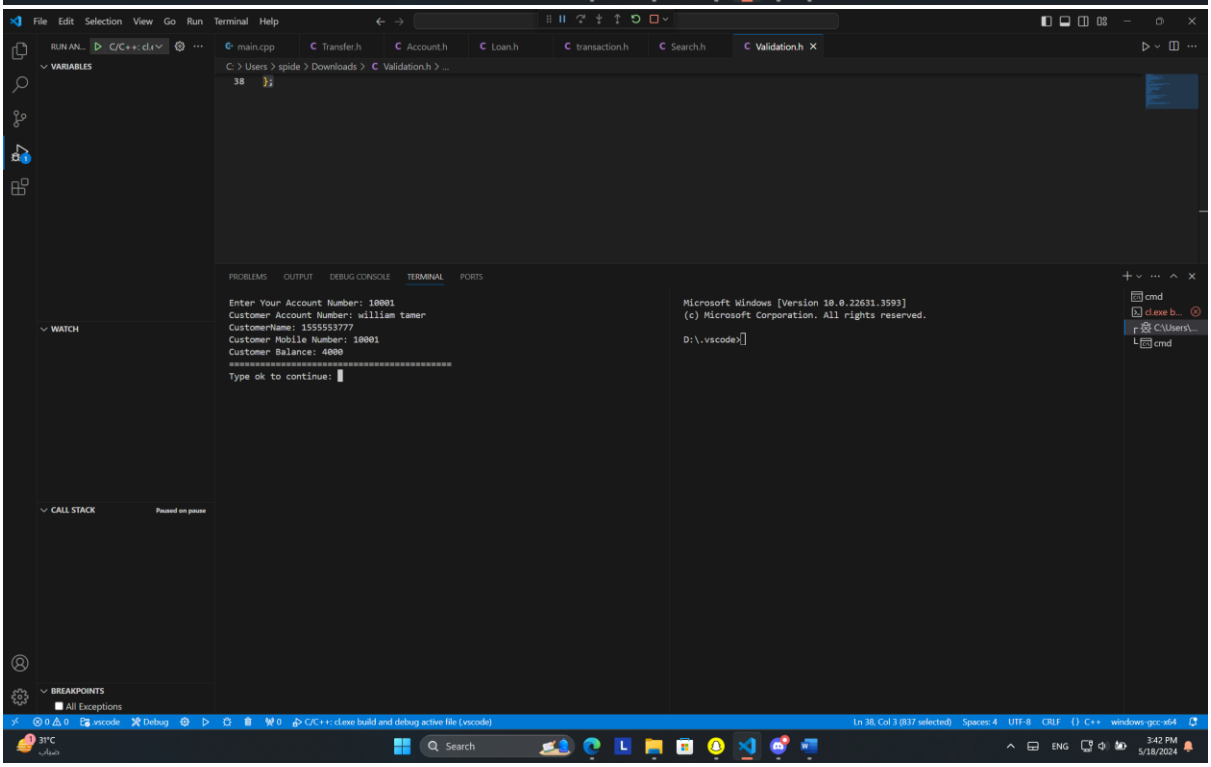

File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
C: > Users > spide > Downloads > Validation.h > ...
38 };
VARIABLES
WATCH
CALL STACK
Paused on pause
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter Amount to Withdraw: 2000
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
cmd
cl.exe b...
C:\Users\...
cmd
Search
ENG
3:41 PM
5/18/2024

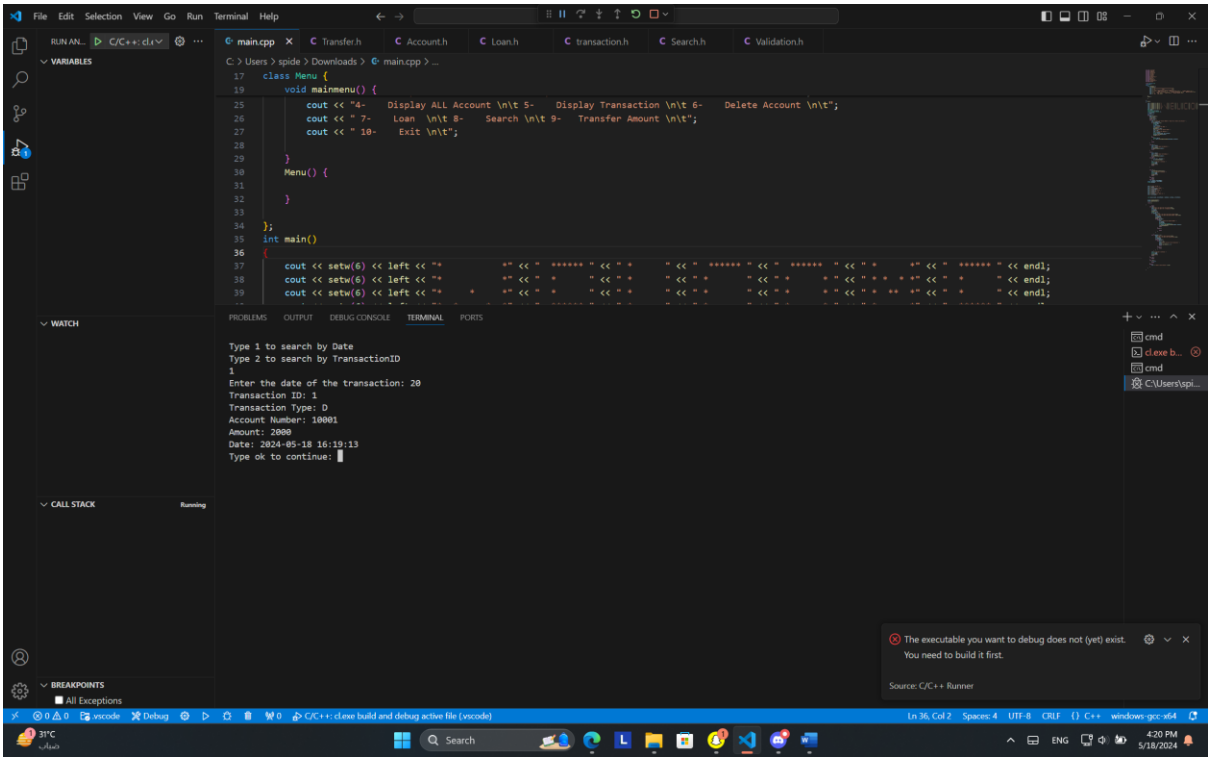

File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
C: > Users > spide > Downloads > Validation.h > ...
38 };
VARIABLES
WATCH
CALL STACK
Paused on pause
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Transaction ID: 2
Transaction Type: w
Account Number: 10002
Amount: 2000
Date: 2024-05-18 15:41:01
Type ok to continue:
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
cmd
cl.exe b...
C:\Users\...
cmd
Search
ENG
3:41 PM
5/18/2024


File Edit Selection View Go Run Terminal Help
main.cpp Transfer.h Account.h Loan.h transaction.h Search.h Validation.h
C: > Users > spide > Downloads > Validation.h > ...
38 };
VARIABLES
WATCH
CALL STACK
Paused on pause
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Customer Account Number: william tamer
CustomerName: 1555553777
Customer Mobile Number: 10001
Customer Balance: 4000
==========
Customer Account Number: paula kameel
CustomerName: 123443222
Customer Mobile Number: 10002
Customer Balance: 1000
==========
Type ok to continue:
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
cmd
cl.exe b...
C:\Users\...
cmd
Ln 38, Col 3 (837 selected) Spaces: 4 UTF-8 CRLF C++ windows-gcc-x64
31°C
Search
ENG
3:41 PM
5/18/2024


File Edit Selection View Go Run Terminal Help
main.cpp Transfer.h Account.h Loan.h transaction.h Search.h Validation.h
C: > Users > spide > Downloads > Validation.h > ...
38 };
VARIABLES
WATCH
CALL STACK
Paused on pause
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Type 1 to search for an Account
Type 2 to search for a transaction
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
cmd
cl.exe b...
C:\Users\...
cmd
Ln 38, Col 3 (837 selected) Spaces: 4 UTF-8 CRLF C++ windows-gcc-x64
31°C
Search
ENG
3:42 PM
5/18/2024


File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
VARIABLES
38 };
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Type 1 to search by Account number
Type 2 to search by Account amount range
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
WATCH
CALL STACK
Paused on pause
BREAKPOINTS
All Exceptions
Search
3:42 PM
5/18/2024


File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
VARIABLES
38 };
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter Your Account Number: 10001
Customer Account Number: william tamer
CustomerName: 1555553777
Customer Mobile Number: 10001
Customer Balance: 4000
Type ok to continue:
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
D:\.vscode>
WATCH
CALL STACK
Paused on pause
BREAKPOINTS
All Exceptions
Search
3:42 PM
5/18/2024


File Edit Selection View Go Run Terminal Help
main.cpp
Transfer.h
Account.h
Loan.h
transaction.h
Search.h
Validation.h
C: > Users > spide > Downloads > main.cpp > ...
class Menu {
void mainmenu() {
cout << "4- Display ALL Account \n\t 5- Display Transaction \n\t 6- Delete Account \n\t";
cout << " 7- Loan \n\t 8- Search \n\t 9- Transfer Amount \n\t";
cout << " 10- Exit \n\t";
}
Menu() {
}
};
int main()
{
cout << setw(6) << left << "+ +" << " ****** " << " + " << " ****** " << " ****** " << " + " << " ****** " << endl;
cout << setw(6) << left << "+ +" << " + " << " + " << " + " << " + " << " + " << " + " << " + " << endl;
cout << setw(6) << left << "+ +" << " + " << " + " << " + " << " + " << " + " << " + " << " + " << endl;
VARIABLES
WATCH
CALL STACK
Running
BREAKPOINTS
All Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Type 1 to search by Date
Type 2 to search by TransactionID
1
Enter the date of the transaction: 20
Transaction ID: 1
Transaction Type: D
Account Number: 10001
Amount: 2000
Date: 2024-05-18 16:19:13
Type ok to continue:
cmd
cl.exe b...
cmd
C:\Users\spi...
The executable you want to debug does not (yet) exist.
You need to build it first.
Source: C/C++ Runner
Ln 36, Col 2 Spaces: 4 UTF-8 CRLF C++ windows-gcc-x64
31°C
Search
ENG
4:20 PM
5/18/2024