Heaven’s light is our guide

Rajshahi University of Engineering &Technology

Assignment on CSE 1203

Department of Computer Science & Engineering

Course Code: CSE 1203

Course Title: Object Oriented Programming

Date- 26 .11.2023

Submitted to:

Dr. Md. Shahid Uz Zaman
Professor

Shyla Afroge
Assistant Professor

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

Submitted by:

Md. Al- Mottaki

Roll No: 1503028

Sec: B

Problem No & Name: We know bKash is a pioneering mobile financial service in Bangladesh that has played a significant role in transforming the country's financial landscape. Launched in 2011, bKash was established as a joint venture between BRAC Bank Limited, one of Bangladesh's leading private banks, and Money in Motion LLC, a US-based technology company. Our goal is to write a similar type of program that would be very close to their functionalities. The name of our application is myCash.

Write a C++ program to create myCash application using class and objects to fulfill the following criteria.

i. The program is capable of storing data for each member permanently in a file. The member data includes mobile-no, name, amount and pin.

ii. To use the system a user must register first.

iii. The program should be menu operated

iv. The program starts with a login menu. If login successful then main menu appears.

v. Define a class with data members mobile, name, amount and pin.

vi. Treat mobile number as the key information for each member.

vii. Create facilities for cash-in, cash-out, send-money and pay- bill.

viii. The system must check for duplicate member.

ix. The application should produces appropriate error message when required (table 1)

x. The system must use OTP (one time passcode) when more security is required like Add Member, Change pin, Send money, Cash-out and Pay bill. The time validity for the OTP is 2 minutes.

xi. The History stores the details of each transaction like Transaction ID, Customer Mobile, Type of Transaction etc. with Date and Time.

Code:

```
1.  #include <bits/stdc++.h>
2.  #include <unistd.h>
3.  #include <chrono>
4.  using namespace std;
5.
6.  class Member
7.  {
```

```
 8.  private:
 9.      string mobile;
10.     string name;
11.     double amount;
12.     string pin;
13.
14. public:
15.     Member() {}
16.     Member(const string &m, const string &n, double a, const string &p)
17.         : mobile(m), name(n), amount(a), pin(p) {}
18.
19.     string getMobile() const { return mobile; }
20.     string getName() const { return name; }
21.     double getAmount() const { return amount; }
22.     string getPin() const { return pin; }
23.
24.     void setName(const string &n) { name = n; }
25.     void setAmount(double a) { amount = a; }
26.     void setPin(const string &p) { pin = p; }
27. };
28.
29. class History
30. {
31. private:
32.     struct Transaction
33.     {
34.         int transactionID;
35.         string description;
36.         double amount;
37.         double balance;
38.     };
39.
40.     vector<Transaction> transactions;
41.
42. public:
43.     void addTransaction(int id, const string &desc, double amt, double bal)
44.     {
45.         Transaction t;
46.         t.transactionID = id;
47.         t.description = desc;
48.         t.amount = amt;
49.         t.balance = bal;
50.         transactions.push_back(t);
51.     }
52.
53.     void displayHistory()
54.     {
55.         cout << "Tran ID\tDescription\tAmount\tBalance" << endl;
56.         for (const auto &t : transactions)
57.         {
58.             cout << t.transactionID << "\t" << t.description << "\t"
```

```
                << t.amount << "\t" << t.balance << endl;
        }
    }
};

class MyCash
{
private:
    vector<Member> members;
    History history;
    string mobile;

    std::chrono::time_point<std::chrono::system_clock> loginTime;

    void startSession()
    {
        loginTime = std::chrono::system_clock::now();
    }

    bool isSessionExpired() const
    {
        auto currentTime = std::chrono::system_clock::now();
        auto elapsedSeconds =
   std::chrono::duration_cast<std::chrono::seconds>(currentTime - loginTime).count();
        return elapsedSeconds >= 120; // 2 minutes session timeout
    }

    string trim(const string &str) const
    {
        size_t start = str.find_first_not_of(" \t");
        size_t end = str.find_last_not_of(" \t");

        if (start == string::npos || end == string::npos)
        {
            return "";
        }

        return str.substr(start, end - start + 1);
    }

    bool isMemberExists(const string &mobile) const
    {
            string trimmedMobile = trim(mobile);
            auto it = find_if(members.begin(), members.end(),
                              [this, trimmedMobile](const Member &m)
                              { return trim(m.getMobile()) == trimmedMobile; });

            return it != members.end();
        }

        Member &getMember(const string &mobile)
```

```
109.            {
110.                auto it = find_if(members.begin(), members.end(),
111.                                  [mobile](const Member &m)
112.                                  { return m.getMobile() == mobile; });
113.
114.                if (it == members.end())
115.                {
116.                    throw logic_error("Member not found");
117.                }
118.
119.                return *it;
120.            }
121.
122.            void saveDataToFile()
123.            {
124.                char buffer[PATH_MAX];
125.                if (getcwd(buffer, sizeof(buffer)) != NULL)
126.                {
127.                    string filePath = string(buffer) + "/myCashData.txt";
128.                    ofstream outFile(filePath);
129.
130.                    if (outFile.is_open())
131.                    {
132.                        for (const auto &member : members)
133.                        {
134.                            outFile << member.getMobile() << " "
135.                                    << member.getName() << " "
136.                                    << member.getAmount() << " "
137.                                    << member.getPin() << "\n";
138.                        }
139.
140.                        outFile.close();
141.                    }
142.                    else
143.                    {
144.                        cerr << "Error opening file for writing." << endl;
145.                    }
146.                }
147.                else
148.                {
149.                    cerr << "Error getting the current working directory." << endl;
150.                }
151.            }
152.            // Function to load member data from a file
153.            void loadDataFromFile()
154.            {
155.                char buffer[PATH_MAX];
156.                if (getcwd(buffer, sizeof(buffer)) != NULL)
157.                {
158.                    string filePath = string(buffer) + "/myCashData.txt"; // Combine with
    the file name
```

```
159.                 ifstream inFile(filePath);
160.
161.                 if (inFile.is_open())
162.                 {
163.                     members.clear();
164.                     string mobile, name, pin;
165.                     double amount;
166.
167.                     while (inFile >> mobile >> name >> amount >> pin)
168.                     {
169.                         members.emplace_back(mobile, name, amount, pin);
170.                     }
171.
172.                     inFile.close();
173.                 }
174.             }
175.             else
176.             {
177.                 cerr << "Error getting the current working directory." << endl;
178.             }
179.         }
180.
181.         int generateOTP() const
182.         {
183.             int otp = rand() % 9000 + 1000;
184.             cout << "myCash OTP: " << otp << endl;
185.             return otp;
186.         }
187.
188.         bool verifyOTP(int otpEntered, int otpGenerated) const
189.         {
190.             return otpEntered == otpGenerated;
191.         }
192.
193.     public:
194.         MyCash()
195.         {
196.             srand(time(0));
197.             loadDataFromFile();
198.         }
199.         void registerMember()
200.         {
201.             cout << "Enter Mobile No. (11-digit): ";
202.             string mobile;
203.             cin >> mobile;
204.
205.             if (isMemberExists(mobile))
206.             {
207.                 cout << "1. Member already exists" << endl;
208.                 return;
209.             }
```

```
210.
211.                 cout << "Enter Name: ";
212.                 string name;
213.                 cin.ignore();
214.                 getline(cin, name);
215.
216.                 cout << "Enter pin (5-digit): ";
217.                 string pin;
218.                 cin >> pin;
219.
220.                 cout << "Reconfirm pin: ";
221.                 string confirmPin;
222.                 cin >> confirmPin;
223.
224.                 int confirmedOtp;
225.
226.                 if (pin != confirmPin)
227.                 {
228.                     cout << "7. Pins must be same" << endl;
229.                     return;
230.                 }
231.
232.                 confirmedOtp = generateOTP();
233.                 cout << "Enter OTP: ";
234.                 int otpEntered;
235.                 cin >> otpEntered;
236.
237.                 // Simulating OTP verification
238.                 if (!verifyOTP(otpEntered, confirmedOtp))
239.                 {
240.                     cout << "5. OTP does NOT matched" << endl;
241.                     return;
242.                 }
243.
244.                 members.emplace_back(mobile, name, 0.0, pin);
245.                 cout << "Registration is Successful" << endl;
246.             }
247.
248.             bool login()
249.             {
250.                 cout << "Enter Mobile No. (11-digit): ";
251.                 cin >> mobile;
252.
253.                 if (!isMemberExists(mobile))
254.                 {
255.                     cout << "2. Member NOT exists" << endl;
256.                     return false;
257.                 }
258.
259.                 cout << "Enter pin: ";
260.
```

```
261.                string pin;
262.                cin >> pin;
263.
264.                Member &member = getMember(mobile);
265.
266.                if (member.getPin() != pin)
267.                {
268.                    cout << "8. Invalid login" << endl;
269.                    return false;
270.                }
271.
272.                cout << "Login is Successful" << endl;
273.                startSession();
274.                return true;
275.            }
276.
277.            void updateMember()
278.            {
279.                if (isSessionExpired())
280.                {
281.                    cout << "6. OTP time has expired. Please log in again." << endl;
282.                    return;
283.                }
284.                try
285.                {
286.                    Member ¤tMember = getMember(mobile);
287.
288.                    cout << "Old Name: " << currentMember.getName() << endl;
289.
290.                    cout << "New Name (enter to ignore): ";
291.                    string newName;
292.                    cin.ignore();
293.                    getline(cin, newName);
294.
295.                    cout << "Old pin: " << currentMember.getPin() << endl;
296.
297.                    cout << "New pin (enter to ignore): ";
298.                    string newPin;
299.                    cin >> newPin;
300.
301.                    if (!newName.empty())
302.                    {
303.                        currentMember.setName(newName);
304.                    }
305.
306.                    if (!newPin.empty())
307.                    {
308.                        cout << "Confirm New pin: ";
309.                        string confirmPin;
310.                        cin >> confirmPin;
311.
```

```
                        if (newPin != confirmPin)
                        {
                            cout << "7. Pins must be same" << endl;
                            return;
                        }

                        int confirmedOtp = generateOTP();
                        cout << "Enter OTP: ";
                        int otpEntered;
                        cin >> otpEntered;

                        // Simulating OTP verification
                        if (!verifyOTP(otpEntered, confirmedOtp))
                        {
                            cout << "5. OTP does NOT matched" << endl;
                            return;
                        }

                        currentMember.setPin(newPin);
                    }

                    cout << "Update is Successful" << endl;
                }
                catch (const logic_error &e)
                {
                    cout << e.what() << endl;
                }
            }

            void removeMember()
            {
                int confirmedOtp = generateOTP();
                cout << "Enter OTP: ";
                int otpEntered;
                cin >> otpEntered;

                // Simulating OTP verification
                if (!verifyOTP(otpEntered, confirmedOtp))
                {
                    cout << "5. OTP does NOT matched" << endl;
                    return;
                }

                members.erase(std::remove_if(members.begin(), members.end(),
                                             [this](const Member &m)
                                             { return m.getMobile() == mobile; }),
                              members.end());

                cout << "Remove is Successful" << endl;
            }

```

```
363.            void sendMoney()
364.            {
365.                if (isSessionExpired())
366.                {
367.                    cout << "6. OTP time has expired. Please log in again." << endl;
368.                    return;
369.                }
370.                cout << "Enter Destination no. (11-digit): ";
371.                string destMobile;
372.                cin >> destMobile;
373.
374.                if (!isMemberExists(destMobile))
375.                {
376.                    cout << "9 Destination Mobile no. is invalid" << endl;
377.                    return;
378.                }
379.
380.                cout << "Enter Amount: ";
381.                double amount;
382.                cin >> amount;
383.
384.                Member &sender = getMember(mobile);
385.                Member &receiver = getMember(destMobile);
386.
387.                if (sender.getAmount() < amount)
388.                {
389.                    cout << "3 Insufficient Fund" << endl;
390.                    return;
391.                }
392.
393.                cout << "Sending " << amount << " to " << destMobile << endl;
394.                cout << "Are you sure(Y/N)? ";
395.                char choice;
396.                cin >> choice;
397.
398.                if (choice == 'Y' || choice == 'y')
399.                {
400.                    int confirmedOtp = generateOTP();
401.                    cout << "Enter OTP: ";
402.                    int otpEntered;
403.                    cin >> otpEntered;
404.
405.                    // Simulating OTP verification
406.                    if (!verifyOTP(otpEntered, confirmedOtp))
407.                    {
408.                        cout << "5. OTP does NOT matched" << endl;
409.                        return;
410.                    }
411.
412.                    sender.setAmount(sender.getAmount() - amount);
413.                    receiver.setAmount(receiver.getAmount() + amount);
```

```
414.
415.                     history.addTransaction(rand() % 1000, "Send Money", amount,
    sender.getAmount());
416.                     cout << "Send Money is Successful" << endl;
417.                 }
418.                 else
419.                 {
420.                     cout << "Send Money Cancelled" << endl;
421.                 }
422.             }
423.
424.             void cashIn()
425.             {
426.                 cout << "Enter Amount: ";
427.                 double amount;
428.                 cin >> amount;
429.
430.                 Member &member = getMember(mobile);
431.
432.                 cout << "Cash-in " << amount << endl;
433.                 cout << "11. Are you sure(Y/N)? ";
434.                 char choice;
435.                 cin >> choice;
436.
437.                 if (choice == 'Y' || choice == 'y')
438.                 {
439.                     member.setAmount(member.getAmount() + amount);
440.                     history.addTransaction(rand() % 1000, "Cash-in", amount,
    member.getAmount());
441.                     cout << "Cash-in is Successful" << endl;
442.                 }
443.                 else
444.                 {
445.                     cout << "Cash-in Cancelled" << endl;
446.                 }
447.             }
448.
449.             void cashOut()
450.             {
451.                 cout << "Enter Amount: ";
452.                 double amount;
453.                 cin >> amount;
454.
455.                 Member &member = getMember(mobile);
456.
457.                 int confirmedOtp = generateOTP();
458.                 cout << "Enter OTP: ";
459.                 int otpEntered;
460.                 cin >> otpEntered;
461.
462.                 // Simulating OTP verification
```

```
463.                if (!verifyOTP(otpEntered, confirmedOtp))
464.                {
465.                    cout << "5 OTP does NOT matched" << endl;
466.                    return;
467.                }
468.
469.                if (member.getAmount() < amount)
470.                {
471.                    cout << "3 Insufficient Fund" << endl;
472.                    return;
473.                }
474.
475.                cout << "Cash-out " << amount << endl;
476.                cout << "11. Are you sure(Y/N)? ";
477.                char choice;
478.                cin >> choice;
479.
480.                if (choice == 'Y' || choice == 'y')
481.                {
482.                    member.setAmount(member.getAmount() - amount);
483.                    history.addTransaction(rand() % 1000, "Cash-out", amount,
    member.getAmount());
484.                    cout << "Cash-out is Successful" << endl;
485.                }
486.                else
487.                {
488.                    cout << "Cash-out Cancelled" << endl;
489.                }
490.            }
491.
492.            void payBill()
493.            {
494.                cout << "Enter Bill Type (Gas/Electricity/Water/Internet-1/2/3/4): ";
495.                int billType;
496.                cin >> billType;
497.
498.                cout << "Your ";
499.                switch (billType)
500.                {
501.                case 1:
502.                    cout << "Gas";
503.                    break;
504.                case 2:
505.                    cout << "Electricity";
506.                    break;
507.                case 3:
508.                    cout << "Water";
509.                    break;
510.                case 4:
511.                    cout << "Internet";
512.                    break;
```

```
513.                default:
514.                    cout << "Invalid";
515.                    break;
516.                }
517.
518.                cout << " Bill: ";
519.                double billAmount;
520.                cin >> billAmount;
521.
522.                cout << "11. Want to pay(Y/N)? ";
523.                char choice;
524.                cin >> choice;
525.
526.                if (choice == 'Y' || choice == 'y')
527.                {
528.                    int confirmedOtp = generateOTP();
529.                    cout << "Enter OTP: ";
530.                    int otpEntered;
531.                    cin >> otpEntered;
532.
533.                    // Simulating OTP verification
534.                    if (!verifyOTP(otpEntered, confirmedOtp))
535.                    {
536.                        cout << "5 OTP does NOT matched" << endl;
537.                        return;
538.                    }
539.
540.                    Member &member = getMember(mobile);
541.                    if (member.getAmount() < billAmount)
542.                    {
543.                        cout << "3 Insufficient Fund" << endl;
544.                        return;
545.                    }
546.
547.                    member.setAmount(member.getAmount() - billAmount);
548.                    history.addTransaction(rand() % 1000, "Pay Bill", billAmount,
    member.getAmount());
549.                    cout << "Bill Payment is Successful" << endl;
550.                }
551.                else
552.                {
553.                    cout << "Bill Payment Cancelled" << endl;
554.                }
555.            }
556.
557.            void checkBalance()
558.            {
559.                cout << "Balance: " << getMember(mobile).getAmount() << endl;
560.            }
561.
562.            void displayHistory()
```

```
563.            {
564.                history.displayHistory();
565.            }
566.            ~MyCash()
567.            {
568.                saveDataToFile();
569.            }
570.        };
571.
572.        int main()
573.        {
574.
575.            MyCash myCash;
576.
577.            int option;
578.
579.            do
580.            {
581.                cout << "\n*** MyCash Login Menu ***" << endl;
582.                cout << "1. Login\n2. Register\n3. Exit\nEnter Your Option: ";
583.                cin >> option;
584.
585.                switch (option)
586.                {
587.                case 1:
588.                    if (myCash.login())
589.                    {
590.                        do
591.                        {
592.                            cout << "\n********** MyCash Menu ********" << endl;
593.                            cout << "1. Update Me\n2. Remove Me\n3. Send Money\n4. Cash-
    in\n5. Cash-out\n"
594.                                 << "6. Pay Bill\n7. Check Balance\n8. History\n9.
    Logout\nEnter Your Option (1-9): ";
595.                            cin >> option;
596.
597.                            switch (option)
598.                            {
599.                            case 1:
600.                                myCash.updateMember();
601.                                break;
602.                            case 2:
603.                                myCash.removeMember();
604.                                option = 9;
605.                                break;
606.                            case 3:
607.                                myCash.sendMoney();
608.                                break;
609.                            case 4:
610.                                myCash.cashIn();
611.                                break;
```

```
                            case 5:
                                myCash.cashOut();
                                break;
                            case 6:
                                myCash.payBill();
                                break;
                            case 7:
                                myCash.checkBalance();
                                break;
                            case 8:
                                myCash.displayHistory();
                                break;
                            case 9:
                                cout << "Logout Successful" << endl;
                                break;
                            default:
                                cout << "10 Invalid Option" << endl;
                            }
                        } while (option != 9);
                    }
                    break;
                case 2:
                    myCash.registerMember();
                    break;
                case 3:
                    cout << "Exiting MyCash Application" << endl;
                    break;
                default:
                    cout << "10.Invalid Option" << endl;
                }

            } while (option != 3);

            return 0;
        }
```

Output:

*** MyCash Login Menu ***
1. Login
2. Register
3. Exit
Enter Your Option: 2
Enter Mobile No. (11-digit): 01910982274
Enter Name: al mottaki
Enter pin (5-digit): 12345
Reconfirm pin: 12345
myCash OTP: 9820

Enter OTP: 9820
Registration is Successful

*** MyCash Login Menu ***
1. Login
2. Register
3. Exit
Enter Your Option: 2
Enter Mobile No. (11-digit): 01910982274
1. Member already exists

*** MyCash Login Menu ***
1. Login
2. Register
3. Exit
Enter Your Option: 1
Enter Mobile No. (11-digit): 01710982274
2. Member NOT exists

*** MyCash Login Menu ***
1. Login
2. Register
3. Exit
Enter Your Option: 1
Enter Mobile No. (11-digit): 01910982274
Enter pin: 12345
Login is Successful

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 3
Enter Destination no. (11-digit): 01710982274
9 Destination Mobile no. is invalid

********** MyCash Menu ********
1. Update Me
2. Remove Me

3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 5
Enter Amount: 10000
myCash OTP: 7619
Enter OTP: 7619
3 Insufficient Fund

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 4
Enter Amount: 15000
Cash-in 15000
11. Are you sure(Y/N)? Y
Cash-in is Successful

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 5
Enter Amount: 20000
myCash OTP: 4665
Enter OTP: 4000
5 OTP does NOT matched

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 11
10 Invalid Option

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 9
Logout Successful

*** MyCash Login Menu ***
1. Login
2. Register
3. Exit
Enter Your Option: 1
Enter Mobile No. (11-digit): 01910982274
Enter pin: 11111
8. Invalid login

*** MyCash Login Menu ***
1. Login
2. Register
3. Exit
Enter Your Option: 2
Enter Mobile No. (11-digit): 01900000000
Enter Name: Kamal Khan
Enter pin (5-digit): 12345
Reconfirm pin: 54321
7. Pins must be same

*** MyCash Login Menu ***
1. Login
2. Register
3. Exit
Enter Your Option: 1
Enter Mobile No. (11-digit): 01910982274
Enter pin: 12345
Login is Successful

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 7
Balance: 15000

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 5
Enter Amount: 5000
myCash OTP: 3218
Enter OTP: 3218
Cash-out 5000
11. Are you sure(Y/N)? Y
Cash-out is Successful

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money

4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 7
Balance: 10000

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 8
Tran ID Description Amount Balance
132 Cash-in 15000 15000
984 Cash-out 5000 10000

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 6
Enter Bill Type (Gas/Electricity/Water/Internet-1/2/3/4): 1
Your Gas Bill: 1000
11. Want to pay(Y/N)? Y
myCash OTP: 3735
Enter OTP: 3735
Bill Payment is Successful

********** MyCash Menu ********
1. Update Me
2. Remove Me

3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 7
Balance: 9000

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 3
Enter Destination no. (11-digit): 0170112345678
9 Destination Mobile no. is invalid

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 6
Enter Bill Type (Gas/Electricity/Water/Internet-1/2/3/4): 3
Your Water Bill: 1000
11. Want to pay(Y/N)? Y
myCash OTP: 8101
Enter OTP: 8234
5 OTP does NOT matched

********** MyCash Menu ********
1. Update Me
2. Remove Me

3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9): 3
6. OTP time has expired. Please log in again.

********** MyCash Menu ********
1. Update Me
2. Remove Me
3. Send Money
4. Cash-in
5. Cash-out
6. Pay Bill
7. Check Balance
8. History
9. Logout
Enter Your Option (1-9):

Conclusion and Discussion: The MyCash application is a simple banking system implemented in C++. It provides basic functionalities such as member registration, login, updating member information, sending money, cash-in, cash-out, paying bills, checking balance, and viewing transaction history.

Key Features:

- **User Registration and Login:** Users can register with a unique mobile number and PIN. The login process ensures secure access to the system.
- **Transaction Operations:** Users can perform various financial transactions, including sending money, cash-in, cash-out, and paying bills.
- **Error Handling:** The application includes error messages for scenarios such as duplicate member registration, insufficient funds, invalid input, and incorrect OTP verification.