ANAS KHAN

SAP ID 11339

REVISED PROJECT

ATM Management system

```
1. #include <stdio.h>
2. #include <conio.h>
3. #include <iostream>
4. #include <string>
5. #include <iomanip>

6. using namespace std;
```

```
7.  class BAcc {
8.  public:
9.  string pinNum;
10. char accType;
11. double bAL;
12. double amt;

13. BankAccount(){
14. bAL = 0.0;
15. amt = 0.0;
16. }


17. void pinNumber();
18. void accountType();
19. void bALChecking();
20. void bALSavings();
21. void deposit();
22. void withdraw();
23. void display();
24. };

25. void BankAccount::pinNumber()
26. {
27. cout <<
    "
    _ _____";
28. cout << "\n\t\t\t       ATM DEMO PROJECT" << endl;
29. cout <<
    "
    _ _____";
30. cout << "\n\t\t\t       ENTER PIN NUMBER: \n" <<endl;
31. cout << "\t\t\t\t      ";


32. for(int i = 0; i <= 3; i++) {
33. char ch;
34. cout << setw(1);
35. ch = _getch();
36. while(ch != 13){
37. pinNum.push_back(ch);
38. cout << '*';
```

```
39. ch = _getch();
40. }


41. if(pinNum == "1234") {
42. cout << "\n\n\t\t\t   Welcome to your account!" << endl;

43. while(!(cin >> pinNum)) { 44. cin.clear();
    45.     while (cin.get() != '\n')
    46.     continue;
    47.     cout << "\n\t\t\t   Please enter a numeric Pin:  ";
    48.     }

    49.     } else if (i >= 4) {
    50.     cout << "\n\t\t\t   You account is Locked!\n";
    51.     cout << "\t\t\t   You have reached the limit of tries." << endl;

    52.     } else {
    53.     cout << "\n\t\t\t   Invalid Pin Number \n";
    54.     cout << "\t\t\t   Enter Pin Number Again: \n";
    55.     cout << "\t\t\t\t   ";
    56.     cin >> pinNum;
    57.     } 58. } 59. }

60. void BAcc::accountType(){
61. cout << "\n\t\t\t Select Account Type:\n";
62. cout << "\t\t\t Checking 'C' or Savings 'S': ";
63. cin >> accType;


64. if(accType == 'C' || accType == 'c') {
65. cout << "\t\t\t================================" << endl << endl; 66. cout <<
    "\t\t\t\tChecking Account" << endl << endl;
67. cout << "\t\t\t================================" << endl << endl;
68. } else if (accType == 'S' || accType == 's') {
69. cout << "\t\t\t================================" << endl << endl; 70. cout <<
    "\t\t\t\tSavings Account" <<endl << endl;
71. cout << "\t\t\t================================" << endl << endl;
72. } else {
73. cout << "\t\t\tInvalid Account Type.\n";
74. cout << "\t\t\tSelect Account Type: 'C' or 'S': ";
75. cin >> accType;
76. }
77. }
```

```
78. void BAcc::deposit(){
79. cout << "\n\t\t\t\t    DEPOSIT\n";
80. cout << "\t\t\t   Enter Deposit Amount $";
81. cin >> amt;
82. cout << "\n\t\t\t========Deposit Receipt========" << endl << endl;
83. cout << setprecision(2) << fixed;
84. cout << "\t\t\t Amount Deposited = $" << amt << endl << endl; 85. bAL += amt;
86. cout << "\t\t\t New balance $" << bAL << endl << endl;
87. cout << "\t\t\t================================" << endl;
88. }

89. void BAcc::withdraw(){
90. cout << "\n\t\t\t\t    WITHDRAW\n";
91. cout << "\t\t\t   Enter Withdraw Amount $";
92. cin >> amt;
93. cout << "\n\t\t\t=======Withdraw Receipt==========" << endl << endl;
94. cout << "\t\t\t Amount withdrawn = $" << amt << endl << endl; 95. bAL -= amt;
    96.     cout << "\t\t\t New balance $" << bAL << endl <<endl;
    97.     cout << "\t\t\t================================" << endl;
    98.     }

    99.     void BAcc::bALChecking() {
    100.    cout <<"\n\t\t\t=====Checking Account Balance=====";
    101.    cout << "\n\t\t\t Checking Account # xxxxxxxx4567\n";
    102.    cout << "\n\t\t\t Balance = $" << bAL << endl;
    103.    cout <<"\n\t\t\t================================" << endl;
    104.    }


    105.    void BAcc::bALSavings() {
    106.    cout << "\n\t\t\t=====Savings Account Balance=====";
    107.    cout << "\n\t\t\t Savings Account # xxxxxxxx8901\n";
    108.    cout << "\n\t\t\t Balance = $" << bAL << endl;
    109.    cout << "\n\t\t\t================================" << endl;
    110.    }


    111.    void BAcc::display(){
    112.    int options = 1;
    113.    while( options != 0 ) {
    114.    cout << "\n\t\t\t  1. Select Account Type\n";
    115.    cout << "\t\t\t  2. Deposit\n";
    116.    cout << "\t\t\t  3. Withdraw\n";
```

```
117.    cout << "\t\t\t  4. Checking Account Balance\n";
118.    cout << "\t\t\t  5. Savings Account Balance\n";
119.    cout << "\t\t\t  0. End Transactions\n";
120.    cout <<"\n\t\t\t\t\t\t";
121.    cin >> options;


122.        if(option==1)
123.        pinNumber();
124.        else if(option==2)
125.        accountType();
126.        else if(option==2)
127.        deposit();
128.        else if(option==2)
129.        withdraw();
130.        else if(option==2)
131.        bALChecking();
132.        else if(option==2)
133.        bALSavings();
134.        else cout << "\n\t\t\t    Invalid Option" << endl;
135.        }
136.        }
137.        }

138.        int main()
139.        {
140.        system("color 0b");

141.        BAcc b;
142.        b.pinNumber();
143.        b.accountType();
144.        b.bALChecking();
145.        b.bALSavings();
146.        b.deposit();
147.        b.withdraw();

148.        return 0;
149.        }
```

Characteristics:

- Naming convention of class is not proper (Line # 7)

- Naming convention of Function is not proper (Line # 18,19,20)
- Naming Convention of Variables are not Proper (Line# 9 to 12)
- Dispensable Code : Display() is implemented but not used. (Line # 111 to 121)
- Naming convention of Object is not proper (Line # 141)
- Excessive use of if-else statements (Line # 122 to 135)
- Comments are not present in the code which makes code complex to read and understand
- Variables should be private or protected in class (Line # 8)
- This is legacy code as it does not implements checks when amount is withdrawn (Line # 89)

ARTIFACT 1.2

Refactored Code:

```
1. #include <stdio.h>
2. #include <conio.h>
3. #include <iostream>
4. #include <string>
5. #include <iomanip>

6. using namespace std;

7. class BankAccount {   // redability,understandibility :naming convention of
   class
8. private: // (variables are private or protected in oo approach)
9. string pinNumber;
10.AccountType type; //    (object created for class)
11.double balance;
12.double amount;

13.public:

14.BankAccount(){
```

```
15.balance = 0.0;
16.amount = 0.0;
17.}

18.void pinNumber(); // ( redability :naming convention of functions)
19.void accountType();
20.void balanceChecking();
21.void balanceSavings();
22.void deposit();
23.void withdraw();
24.void display();
25.};

26.void BankAccount::pinNumber()
27.{
28.cout <<
29."_____
   _______";
30.cout << "\n\t\t\t        ATM DEMO PROJECT" << endl;
31.cout <<
32."_____
   _______";
33.cout << "\n\t\t\t        ENTER PIN NUMBER: \n" <<endl;
34.cout << "\t\t\t\t      ";

35.for(int i = 0; i <= 3; i++) {
36.char ch;
37.cout << setw(1);
38.ch = _getch();
39.while(ch != 13){
40.pinNumber.push_back(ch);
41.cout << '*';
42.ch = _getch();
43.}

44.if(pinNumber == "1234") {
45.cout << "\n\n\t\t\t   Welcome to your account!" << endl;
46.while(!(cin >> pinNumber)) {
47.cin.clear();
48.while (cin.get() != '\n')
49.continue;
50.cout << "\n\t\t\t   Please enter a numeric Pin:  ";
51.}
52.}
```

```
54.cout << "\n\t\t\t   You account is Locked!\n";
55.cout << "\t\t\t   You have reached the limit of tries." << endl;
56.}
57.else {
58.cout << "\n\t\t\t   Invalid Pin Number \n";
59.cout << "\t\t\t   Enter Pin Number Again: \n";
60.cout << "\t\t\t\t   ";
61.cin >> pinNumber; // (naming conventions corrected)
62.}
63.}
64.}


65.void BankAccount::deposit(){
66.cout << "\n\t\t\t\t    DEPOSIT\n";
67.cout << "\t\t\t   Enter Deposit Amount $";
68.cin >> amount;
69.cout << "\n\t\t\t========Deposit Receipt========" << endl << endl;
70.cout << setprecision(2) << fixed;
71.cout << "\t\t\t Amount Deposited = $" << amount << endl << endl;  72. balance
   += amount;
73.cout << "\t\t\t New balance $" << balance << endl << endl;
74.cout << "\t\t\t==============================" << endl;
75.}
76.void BankAccount::withdraw(){
77.cout << "\n\t\t\t\t    WITHDRAW\n";
78.cout << "\t\t\t   Enter Withdraw Amount $";  79. cin >> amount;
80.if(amount<=balance){ // (checks implemented)
81.cout << "\n\t\t\t=======Withdraw Receipt==========" << endl << endl;
82.cout << "\t\t\t Amount withdrawn = $" << amount << endl << endl;
83.balance -= amount;
84.cout << "\t\t\t New balance $" << balance << endl <<endl;
85.cout << "\t\t\t================================" << endl;
86.}
87.else{
88.cout << "\t\t\t insuficient Balance $" << balance << endl <<endl;
89.} //(checks implemented)
90.}
```

```
91.void BankAccount::balanceChecking() {
92.cout <<"\n\t\t\t=====Checking Account Balance=====";
93.cout << "\n\t\t\t Checking Account # xxxxxxxx4567\n";
```

```
94.cout << "\n\t\t\t Balance = $" << balance << endl;
95.cout <<"\n\t\t\t==================================" << endl;
```

96.}

```
97.void BankAccount::balanceSavings() {
98.cout << "\n\t\t\t=====Savings Account Balance=====";
99.cout << "\n\t\t\t Savings Account # xxxxxxxx8901\n";
100.  cout << "\n\t\t\t Balance = $" << balance << endl;
101.  cout << "\n\t\t\t================================" << endl;
102.  }

103.      void BankAccount::display(){
104.         int options = 1;
105.  while( options != 0 ) {
106.  cout << "\n\t\t\t  1. Select Account Type\n";
107.  cout << "\t\t\t  2. Deposit\n";
108.  cout << "\t\t\t  3. Withdraw\n";  //   (dispensable code used in line167)
109.  cout << "\t\t\t  4. Checking Account Balance\n";
110.  cout << "\t\t\t  5. Savings Account Balance\n";
111.  cout << "\t\t\t  0. End Transactions\n";
112.  cout <<"\n\t\t\t\t\t\t";
113.  cin >> options;

114.  switch(option) //   (switch statement used instead of if else line    114
   to line 136)
115.  case 1:
116.  pinNumber();
117.  break;
118.  case 2:
119.  type.selectAccountType();
120.  break;
121.  case 3:
122.  deposit();
123.  break;
124.  case 4:
125.  withdraw();
126.  break;
127.  case 5:
128.  balanceChecking();
129.  break;
130.  case 6:
131.  balanceSavings();
132.  break;
133.  default:
134.  cout << "\n\t\t\t Invalid Option" << endl;
135.  break;
136.  }
137.  }
```

```
138.  class AccountType{  // class created for removal of oo abuse )
139.  private:
140.  char type;
141.  public:
142.  void selectAccountType(){
143.  cout << "\n\t\t\t Select Account Type:\n";
144.  cout << "\t\t\t Checking 'C' or Savings 'S': ";
145.      cin >> accountType;  ( naming convention corrected)
146.  if(accountType == 'C' || accountType == 'c') {
147.  cout << "\t\t\t==============================" << endl << endl;
148.  cout << "\t\t\t\tChecking Account" << endl << endl;
149.  cout << "\t\t\t==============================" << endl << endl;
150.  }
151.  else if (accountType == 'S' || accountType == 's') {
152.  cout << "\t\t\t==============================" << endl << endl;
153.  cout << "\t\t\t\tSavings Account" <<endl << endl;
154.  cout << "\t\t\t==============================" << endl << endl;
155.  }
156.  else {
157.  cout << "\t\t\tInvalid Account Type.\n";
158.  cout << "\t\t\tSelect Account Type: 'C' or 'S': ";
159.  cin >> accountType;
160.  }

161.  }
162.  }
```

```
163.  int main()
164.  {
165.  system("color 0b");
166.  BankAccount b;
167.  b.display(); // (dispensable code used)
168.  return 0;
169.  }
```

ARTIFACT 1.3
CODE SLICING
FORWARD SLICING

Slicing goal: Dead code removal
Slicing criteria : amount
Slicing history(4,8,12,13,18,19,22,30,31,32)

```
1. void BankAccount::deposit(){
2. cout << "\n\t\t\t\t    DEPOSIT\n";
3. cout << "\t\t\t   Enter Deposit Amount $";
4. cin >> amount;
5. cout << "\n\t\t\t========Deposit Receipt========" << endl << endl;
6. cout << setprecision(2) << fixed;
7. cout << "\t\t\t Amount Deposited = $" << amount << endl << endl;
8. balance += amount;
9. cout << "\t\t\t New balance $" << balance << endl << endl;
10.cout << "\t\t\t==============================" << endl;
11.}
12. cin >> amount;
13. balance += amount;
14. cout << "\t\t\t New balance $" << balance << endl << endl;
15.void BankAccount::withdraw(){
16. cout << "\n\t\t\t\t    WITHDRAW\n";
17. cout << "\t\t\t   Enter Withdraw Amount $";
18. cin >> amount;
19.if(amount<=balance){
20. cout << "\n\t\t\t=======Withdraw Receipt==========" << endl << endl;
 21.cout << "\t\t\t Amount withdrawn = $" << amount << endl << endl;
 22.balance -= amount;
 23.cout << "\t\t\t New balance $" << balance << endl <<endl;
 24.cout << "\t\t\t================================" << endl;
 25.}
 26.else{
 27.cout << "\t\t\t insuficient Balance $" << balance << endl <<endl;
 28.}
29. cin >> amount;
30. if(amount<=balance){
31. balance -= amount;
32. cout << "\t\t\t New balance $" << balance << endl <<endl;
```

Slicing goal: Dead code removal
Slicing criteria : amount
Slicing history(4,8,12,13,18,19,22,30,31,32)

```
1. void BankAccount::pinNumber()
2. {
3. cout <<
4. "_____
   _______";
5. cout << "\n\t\t\t        ATM DEMO PROJECT" << endl;
6. cout <<
7. "_____
   _______";
8. cout << "\n\t\t\t        ENTER PIN NUMBER: \n" <<endl;
9. cout << "\t\t\t\t      ";

10.for(int i = 0; i <= 3; i++) {
11.char ch;
12.cout << setw(1);
13.ch = _getch();
14.while(ch != 13){
15.pinNumber.push_back(ch);
16.cout << '*';
17.ch = _getch();
18.}

19.if(pinNumber == "1234") {
20.cout << "\n\n\t\t\t   Welcome to your account!" << endl;
21.while(!(cin >> pinNumber)) {
22.cin.clear();
23.while (cin.get() != '\n')
24.continue;
25.cout << "\n\t\t\t   Please enter a numeric Pin:  ";
26.}
27.}
28.else if (i >= 4) {
29.cout << "\n\t\t\t   You account is Locked!\n";
30.cout << "\t\t\t   You have reached the limit of tries." << endl;
31.}
32.else {
33.cout << "\n\t\t\t   Invalid Pin Number \n";
34.cout << "\t\t\t   Enter Pin Number Again: \n";
35.cout << "\t\t\t\t   ";
36.cin >> pinNumber; }
37.}
38.}
```

```
39.    for(int i = 0; i <= 3; i++) {
40.    char ch;
41.    cout << setw(1);
42.    ch = _getch();
43.    while(ch != 13){
44.    pinNumber.push_back(ch);
45.    cout << '*';
46.    ch = _getch();
47.    }

48.    if(pinNumber == "1234") {
49.    while(!(cin >> pinNumber)) {
50.    cin.clear();
51.    while (cin.get() != '\n')
52.    continue;
53.    else if (i >= 4) {
54.    else {
55.    cin >> pinNumber; }
56.    }
```

ARTIFACT 2.1 AND 2.2

TEST CASES

WITHDRAW AMOUNT CODE :

```
void bankAccount::withdraw(){
  cout << "\n\t\t\t\t    WITHDRAW\n";
  cout << "\t\t\t   Enter Withdraw Amount $";
  cin >> amount;
  if(amount<=balance)
  cout << "\n\t\t\t=======Withdraw Receipt==========" << endl << endl;
  cout << "\t\t\t Amount withdrawn = $" << amount << endl << endl;
  balance -= amount;
  cout << "\t\t\t New balance $" << balance << endl <<endl;
  cout << "\t\t\t================================" << endl;
  else
  cout<<"Invalid Balance!"<<endl;
```

TEST CASES:

Input: Amount

Valid class

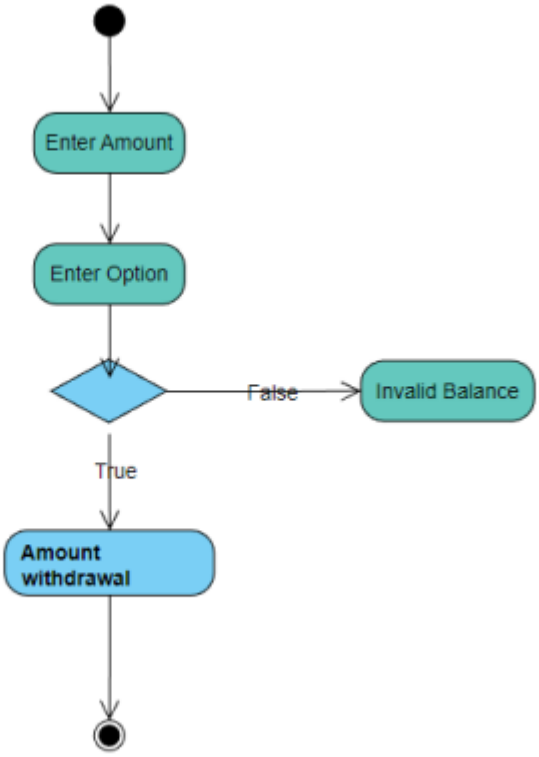
Amount > 0

Invalid Class:

Amount< =0

Test Case Id	Amount	ECP	Actual output
1	1000	Amount > 0	Withdraw Receipt Amount Withdraw
2	0	Amount< =0	Invalid Balance

ACTIVITY DIAGRAM :



PIN NUMBER CODE

```
void BankAccount::pinNumber()
{
   cout <<
"
 ____________________________________________________________________________
";
   cout << "\n\t\t\t        ATM DEMO PROJECT" << endl;
   cout <<
"
 ____________________________________________________________________________
";
   cout << "\n\t\t\t        ENTER PIN NUMBER: \n" <<endl;
   cout << "\t\t\t\t      ";


   for(int i = 0; i <= 3; i++) {
     char ch;
     cout << setw(1);
     ch = _getch();
     while(ch != 13){
        pinNum.push_back(ch);
        cout << '*';
        ch = _getch();
     }


      if(pinNum == "1234") {
         cout << "\n\n\t\t\t   Welcome to your account!" << endl;

         while(!(cin >> pinNum)) {
            cin.clear();
          while (cin.get() != '\n')
              continue;
           cout << "\n\t\t\t   Please enter a numeric Pin:  ";
         }


    } else if (i >= 4) {
         cout << "\n\t\t\t   You account is Locked!\n";
          cout << "\t\t\t   You have reached the limit of tries." << endl;

      } else {
         cout << "\n\t\t\t   Invalid Pin Number \n";
         cout << "\t\t\t   Enter Pin Number Again: \n";
         cout << "\t\t\t\t   ";
         cin >> pinNum;
```

TEST CASES:

Input: Pin number

Valid class

pinNumber.length=4

Invalid Class:

pinNumber.length<4

pinNumber.length>4

Test Case Id	Amount	ECP	Actual output
1	4	pinlength=4	**** Welcome To Your Account
2	5	>4	**** Invalid Pin Number
3	3	<4	**** Invalid pin Number

ACTIVITY DIAGRAM: