C++ ATM SIMULATOR

M&S Project First Review

PROPOSED TO:

Rahul Sir

PREPARED BY:

LAKSHAY

2K19/SE/067
KUSHAL JAIN
2K19/SE/066

CONTENTS

ABSTRACT

THE ATM simulation is a project that is used by people to access the bank account in order to withdraw the money and when the one has done withdrawing its money then the money got debited by its bank account balance.

The ATM worked as single server queuing system that means serves one customer at a time. In order to use that customer first insert the card and then enter the PIN. Then these goes to backend for validation and after validation machine asks the user the user for further operation he wants to perform. If the PIN is found to be invalid the program shows that PIN is invalid and asks to enter again. If the PIN entered is wrong three times it may block the user card.

The customer can do the the operations like cash withdrawal, changing PIN, check account balance, checking account details, checking previous transactions .

ACKNOLEDGEMENT

The way can't walk itself. We have to walk on it. For that we must have a guide. Many guides have contributed to the successful completion of the project we would like to place on record my grateful thanks to each one of them who help us in this project.

Before we get into thick of the thing, we would like to add a few heartfelt words for the people who gave us unending time support whichever and whenever necessary, our grateful thanks go to our dept. ,which provides us an opportunity as a project subject in 3" semester to develop a report work skill in this system analysing .

We would like to thank our parents & friends for giving us full feedback when we are in trouble.

Our special thanks go to Rahul Sir to give their expert guidance to us whenever necessary.

INTRODUCTION:

ABOUT THE ATM (AUTOMATIC TELLER MACHINE)

An automated teller machine or automatic teller machine (ATM), also known as an automated banking machine (ABM) in Canada, and a Cashpoint (which is a trademark of Lloyds TSB), cash machine or sometimes a hole in the wall in British English, is a computerized telecommunications device that provides the clients of a financial institution with access to financial transactions in a public space without the need for a cashier, human clerk or bank teller. ATMs are known by various other names including ATM machine, automated banking machine, and various regional variants derived from trademarks on ATM systems held by particular banks.

On most modern ATMs, the customer is identified by inserting a plastic ATM card with a magnetic stripe or a plastic smart card with a chip, that contains a unique card number and some security information such as an expiration date or CVVC (CVV). Authentication is provided by the customer entering a personal identification number (PIN).

Using an ATM, customers can access their bank accounts in order to make cash withdrawals, debit card cash advances, and check their account balances as well as purchase prepaid cell phone credit. If the currency being withdrawn from the ATM is different from that which the bank account is denominated in (e.g.: Withdrawing Japanese Yen from a bank account containing US Dollars), the money will be converted at an official wholesale exchange rate.

SOFTWARE AND HARDWARE REQUIREMENTS

- PRINTER : ANY
- COMPILER : DEV C++, CODE BLOCKS
- OPERATING SYSTEM: WINDOWS XP, 7,8,10.
- RAM : 1 GB OR MORE
- PROCESSOR : DUAL CORE
- HARD DISK : 40GB

THEORY OF CONCEPTS USED

Input /Output with files

C++ provides the following classes to perform output and input of characters to/from files:

S.No	Data Type & Description
1	**Ofstream** *This data type shows the output file stream and used to create files and to write information to files.*
2	**ifstream** *This data type shows the input file stream and is used to read information from files.*
3	**fstream** *This data type represents the file stream generally, and has the capabilities of both ofstream and ifstream which means it can create files, write information to files, and read information from files.*

All C++ compilers come with classes for streaming input from the console and output to the console. These classes are defined by putting the directive #include <iostream> at the top of the code. The istream class has methods for detecting input errors and the end of input data. The ostream class has methods for formatting output, i.e. specifying scientific notation, fixed decimal notation, or a combination thereof, and for specifying the number of decimal digits displayed. Using some of the features of these classes, we add the capability of reading and writing our own custom types. Finally, the ifstream and ofstream classes let us read from and write to named files.

Opening a File

A file must be opened before you can read from it or write to it. Either **ofstream** or **fstream** object may be used to open a file for writing. And ifstream object is used to open a file for reading purpose only.

Following is the standard syntax for open() function, which is a member of fstream, ifstream, and ofstream objects.

```
void open(const char *filename, ios::opening mode);
```

Here, the first argument specifies the name and location of the file to be opened and the second argument of the **open()** member function defines the mode in which the file should be opened.

Sr.No	Mode Flag & Description
1	**ios::app** *Append mode. All output to that file to be appended to the end.*
2	**ios::ate** *Open a file for output and move the read/write control to the end of the file.*
3	**ios::in** *Open a file for reading.*
4	**ios::out** *Open a file for writing.*
5	**ios::trunc** *If the file already exists, its contents will be truncated before opening the file.*

You can combine two or more of these values by **OR**ing them together. For example if you want to open a file in write mode and want to truncate it in case that already exists, **following will be the syntax –**

```
ofstream outfile;
outfile.open("file.dat", ios::out || ios::trunc );
```

Similar way, you can open a file for reading and writing purpose as follows −

```
fstream  afile;
afile.open("file.dat", ios::out || ios::in );
```

Closing a File

When a C++ program terminates it automatically flushes all the streams, release all the allocated memory and close all the opened files. But it is always a good practice that a programmer should close all the opened files before program termination.

Following is the syntax for close() function

```
void close();
```

Writing to a File

While doing C++ programming, you write information to a file from your program using the stream insertion operator (<<) just as you use that operator to output information to the screen. The only difference is that you use an **ofstream** or **fstream** object instead of the **cout** object.

Reading from a File

You read information from a file into your program using the stream extraction operator (>>) just as you use that operator to input information from the keyboard. The only difference is that you use an **ifstream** or **fstream** object instead of the **cin** object.

C++ strtok()

The strtok() function in C++ returns the next token in a null terminated byte string.

strtok() prototype

```
char* strtok( char* str, const char* delim );
```

The strtok() function takes two arguments: str and delim. This function finds the token in the string pointed to by strtok. The pointer delim points to the separator characters.
This function can be called multiple times to obtain tokens from the same string. There are two cases:

- **If str is not NULL:**
  A call to strtok() is considered first call for that string. The function searches for the first character that is not contained in delim. If no such character is found, the string does not contain any token. So a null pointer is returned. If such character is found, from there on the function searches for a character that is present in delim. If no separator is found, str has only one token. If a separator is found, it is replaced by '\0' and the pointer to the following character is stored in a static location for subsequent invocations. Finally, the function returns the pointer to the beginning of the token.

- **If str is NULL:**
  The call is considered as subsequent calls to strtok and the function continues from where it left in previous invocation.

It is defined in <cstring> header file.

strtok() Parameters

- delim: Pointer to the null terminated byte string that contains the separators.

- str : Pointer to the null terminated byte string to tokenize.

strtok() Return value

The strtok() function returns the pointer to the next token if there is any, or it returns NULL if no more tokens are found.

C++ strtol()

The strtol() function in C++ interprets the contents of a string as an integral number of the specified base and return its value as a long int.

The strtol() function in C++ interprets the contents of a string as an integral number of the specified base and return its value as a long int. This function also sets a pointer to point to the first character after the last valid character of the string if there is any, otherwise the pointer is set to null.

```
For base 10 and the string "12abc":
Valid numeric part -> 12
First character after valid numeric part -> a
```

strtol() prototype [As of C++ 11 standard]

```
long int strtol(const char* str, char** end, int base);
```

The strtol() function takes string, a pointer to character and an integer value - base as its parameter, interprets the content of string as an integral number of the given base and returns a long int value.

This function is defined in <cstdlib> header file.

strtol() Parameters

- `str:` A string having the representation of an integral number.
- `end:` Reference to an already allocated object of type char*. The value of end is set by the function to the next character in str after the last valid character. This parameter can also be a null pointer, in which case it is not used.
- `base:` The base of the integral value. The set of valid values for base is {0, 2, 3, …, 35, 36}.

strtol() Return value

The strtol() function returns:

- a long int value (which is converted from the string).
- 0 if no valid conversion could be performed.

INHERITANCE:

Inheritance-object is of the derived class-“atm1”…….base class is “atm”

Inheritance is a mechanism of reusing and extending existing classes without modifying them, thus producing hierarchical relationships between them.

Inheritance is almost like embedding an object into a class. Suppose that you declare an object x of class A in the class definition of B. As a result, class B will have access to all the public data members and member functions of class A. However, in class B, you have to access the data members and member functions of class A through object x

friend Function in C++

A **friend function** can access the **private** and **protected** data of a class. We declare a friend function using the `friend` keyword inside the body of the class.

```
class className {
    ... .. ...
    friend returnType functionName(arguments);
    ... .. ...
}
```

C++ ctime()

The ctime() function in C++ converts the given time since epoch to a calendar local time and then to a character representation.

A call to `ctime(time)` is a combination of asctime() and localtime() functions, as `asctime(localtime(time))`.
It is defined in <ctime> header file.

ctime() prototype

```
char* ctime(const time_t* time_ptr);
```

The ctime() function takes a pointer to `time_t` object as its parameter and returns a text representation of the form:

```
Www Mmm dd hh:mm:ss yyyy
```

ctime() time representation

Type	Description	Values
Www	3 letter day of week	Mon to Sun
Mmm	3 letter month name	Jan to Dec
dd	2 digit day of month	00 to 31
hh	2 digit hour	00 to 23
mm	2 digit minute	00 to 59
ss	2 digit second	00 to 59
yyyy	4 digit year	4 digit year

ctime() Parameters

- time_ptr: pointer to a time_t object to be converted.

ctime() Return value

- Pointer to a null terminated string the points to the character representation of the date and time.

FLOW OF CONTROL IN THE PROGRAMMING:

In main function:

>Pin is inserted and verified

>Options are given to the user.(display menu)

OPTION 1: BALANCE

>Checking the first token

>Tokenizing and going to the balance token

>Displaying the current account balance

OPTION 2: WITHDRAWAL

>Temp file is opened in append mode

>Password is verified to select required line.

>Flow goes to enter_w().

>The tokenized string is converted into long int

>Condition of withdrawal is being checked.

>The flow is send to balance is updated.

>Balance file is updated.

OPTION 3: MINI STATEMENT

>Opening of bal.txt file

>Verification of password

>Tokenization of text file

>Printing the last 10 transaction.

>Closing of bal.txt

OPTION 4: ACCOUNT DETAILS

>Search the pin

>Then tokenize each every detail in the line and then display the content

OPTION 5-UPDATING PASSWORD

> old password is confirmed

> enter the new password

> tokenizing and printing the detail in text file.

>new password is updated .flow is send to updation.

UPDATING BALANCE

>Opening of balance and text file.

>First token (password) is verified to pick the line.

>Balance is updated.

>bal.txt is removed

>Temp file is renamed

SOURCE CODE

Header files and variables

```
#include <iostream>
#include <conio.h>
#include <fstream>
#include <string.h>
#include <cstdlib>
#include <process.h>
#include <ctime>

using namespace std;


string pno = "";
string name = "";
string ano = "";
string acct = "";
string cno = "";
string amt = "";
string pin_no = "";
string pass = "";
int flag, flag2 = 0;
long int a = 0;
int at = 0;
char line[200];

```

FUNCTION FOR ENTERING THE AMOUNT TO BE WITHDRAWN

```
class atm
{
public:
    void trans();
    void enter_w()
    {
        ofstream out;
        out.open("temp1.txt", ios::app);

        if (!out)
        {
            cout << "Cannot open the Record file!!!"<<endl;
            return;
        }

        out << strtok(line, "%") << "%";
        for (int i = 1; i <= 4; i++)
            out << strtok(NULL, "%") << "%";

        char *p = strtok(NULL, "%");
        char *end;
        a = strtol(p, &end, 10);
        cout << "\nEnter the amount in multiple of 100 only : \n"
             << "Rs. : ";
        while (1)
        {
            cin >> at;

            if (!(at % 100))
            {

                if (at < a)
                {

                    if (25000 >= at)
                    {
                        a = a - at;
                        break;
                    }

                else
                {
                    cout << "\nEntered Amount has exceeded the Withdrawal Limit"<<endl;
                    cout << "\nRe Enter the ammount :\n"
                         << "Rs. : ";
                }
            }

            else
            {
                cout << "\nEntered Amount is more than your available Balance\nAs your Available Balance is just : " << a <<endl;
                cout << "\nRe Enter the ammount :\n"
                     << "Rs. : ";
            }
        }

        else
        {
            cout << "\nEntered amount is Not In Multiple of 100"<<endl;
            cout << "\nRe Enter the ammount :\n"
                 << "Rs. : ";
        }
    }

    out << a << "%" << endl;
    out.close();

    cout << "\nYour Transaction Is In Process"<<endl;
    return;
}
```

FUNCTION FOR DISPLAYING THE CURRENT ACCOUNT BALANCE

```
105     void balance()
106     {
107
108         ifstream in;
109         in.open("atm.txt");
110
111         if (!in)
112         {
113             cout << "\nRecord File not found...!!!!"<<endl;
114             return;
115         }
116
117         while (!in.eof())
118         {
119             in.getline(line, 255);
120
121             for (int i = 0; i < 4; i++)
122             {
123                 if (line[i] == pass[i])
124                     flag = 0;
125
126                 else
127                 {
128                     flag = 1;
129                     break;
130                 }
131             }
132
133             if (flag == 0)
134                 break;
135         }
136
137         if (flag == 0)
138         {
139             strtok(line, "%");
140
141             for (int i = 1; i < 5; i++)
142                 strtok(NULL, "%");
143
144                 cout << "\nYour Available Account Balance is : ";
145                 cout << strtok(NULL, "%") << <<endl;
146             }
147
148             in.close();
149         }
150
```

FUNCTION FOR DISPLAYING THE ACCOUNT DETAIL OF THE ACCOUNT RELATED TO ENTERED PIN

```
158     void search()
159     {
160
161         while (1)
162         {
163
164             ifstream in;
165             in.open("atm.txt");
166
167             if (!in)
168             {
169                 cout << "\nRecord File not found...!!!!"<<endl;
170                 return;
171             }
172
173             while (!in.eof())
174             {
175                 in.getline(line, 255);
176
177                 for (int i = 0; i < 4; i++)
178                 {
179                     if (line[i] == pass[i])
180                         flag = 0;
181
182                     else
183                     {
184                         flag = 1;
185                         break;
186                     }
187                 }
188
189                 if (flag == 0)
190                     break;
191             }
192
193             if (flag == 0)
194             {
195                 char *p;
196
197                 p = strtok(line, "%");
```

```
198                 cout << "\nPIN NO    : "
199                      << "****";
200
201                 p = strtok(NULL, "%");
202                 cout << "\nNAME      : " << p;
203
204                 p = strtok(NULL, "%");
205                 cout << "\nAcc. No.  : " << p;
206
207                 p = strtok(NULL, "%");
208                 cout << "\nAcc. Type : " << p;
209
210                 p = strtok(NULL, "%");
211                 cout << "\nCARD No.  : " << p;
212
213                 p = strtok(NULL, "%");
214                 cout << "\nTotal Bal.: " << p;
215
216                 break;
217             }
218
219             in.close();
220         }
221     }
222
```

TO INPUT THE NEW PIN WHEN PASSWORD UPDATE IS REQUIRED

```
228     void input_p()
229     {
230         int count = 1;
231         cout<<"\n\nConfirm Pin No. : ";
232 
233         while(1)
234         {
235             char c = ' ';
236             pass = "";
237 
238             while(c!=13)
239             {
240                 c=getch();
241                 if(c!=13)
242                 {
243                     pass+=c;
244                     cout<<"%";
245                 }
246             }
247 
248             if(pin_no != pass)
249             {
250 
251                 if(count<=3)
252                 {
253                     cout<<"\nWrong PIN"<<endl;
254                     count++;
255 
256                     if(count>3)
257                     {
258                         cout<<"\nNo. of trials for entering your PIN is over and YOUR CARD IS BLOCKED"<<endl;
259                         getch();
260                         exit(0);
261                     }
262 
263                     if(count == 3)
264                         cout<<"\n\nThis is the Last Attempt to enter your correct PIN otherwise : \n\n\t\t YOUR CARD WILL BE BLOCKED"<<endl<<endl;
265 
266                     cout<<"\nRe-Enter the Pin No. : ";
```

```
269
270                 else
271                     break;
272
273             }
274
275             else
276                 break;
277
278         }
279
280         ifstream in;
281         in.open("atm.txt");
282         if(!in)
283         {
284             cout<<"\nRecord File not found...!!!!"<<endl;
285             return;
286         }
287
288         while(1)
289         {
290
291             while(!in.eof())
292             {
293
294                 in.getline(line,255);
295
296                 for(int i=0;i<4;i++)
297                 {
298                     if(line[i] == pass[i])
299                         flag=0;
300
301                     else
302                     {
303                         flag=1;
304                         break;
305                     }
```

```
306
307                 }
308
309                 if(flag==0)
310                     break;
311
312             }
313
314             if(flag==0)
315             {
316
317                 string p = "";
318                 p=strtok(line,"%");
319
320                 cout<<"\n\nEnter the new PIN of exactly 4 DIGITS else first 4 digits will only be valid\n\n";
321                 cout<<"\nNew PIN : ";
322
323                 int x = 0;
324                 char c = ' ';
325                 pass = "";
326
327                 while(c!=13 && x<4)
328                 {
329                     c=getch();
330
331                     x++;
332
333                     if(c!=13)
334                     {
335                         pass+=c;
336                         cout<<"%";
337                     }
338                 }
339
340                 pno = pass;
341
342                 p=strtok(NULL,"%");
343                 name = p;
344
```

```
335                         pass+=c;
336                         cout<<"%";
337                     }
338                 }
339
340                 pno = pass;
341
342                 p=strtok(NULL,"%");
343                 name = p;
344
345                 p=strtok(NULL,"%");
346                 ano = p;
347
348                 p=strtok(NULL,"%");
349                 acct = p;
350
351                 p=strtok(NULL,"%");
352                 cno = p;
353
354                 p=strtok(NULL,"%");
355                 amt = p;
356
357                 break;
358
359             }
360
361             else
362                 return;
363
364         }
365     }
366
```

FUNCTION TO UPDATE THE FILE WITH NEWLY ENTERED PASSWORD

```
370
371     void enter_p()
372     {
373         ofstream out;
374         out.open("atm.txt",ios::app);
375
376         if(!out)
377             cout<<"Cannot open the Record file!!!"<<endl;
378
379         out<<"\n"<<pno<<"%"<<name<<"%"<<ano<<"%"<<acct<<"%"<<cno<<"%"<<amt<<"%"<<endl;
380         out.close();
381
382     }
383
```

FUNCTION TO UPDATE THE PIN

```
388     void update()
389     {
390
391         ifstream in;
392         ofstream out;
393
394         in.open("atm.txt");
395         out.open("temp.txt");
396
397         while(!in.eof())
398         {
399
400             in.getline(line,255);
401
402             for(int i=0;i<4;i++)
403             {
404                 if(line[i] == pin_no[i])
405                     flag=0;
406
407                 else
408                 {
409                     flag=1;
410                     break;
411                 }
412
413             }
414
415             if(flag==1)
416                 out<<line<<endl;
417
418             else
419             {
420                 flag2=1;
421                 input_p();
422                 enter_p();
423             }
424         }
425
```

```
426             in.close();
427             out.close();
428
429             remove("atm.txt");
430             rename("temp.txt","atm.txt");
431
432             if(flag2==0)
433                 cout<<"\nNO RECORD FOUND!!!\n";
434
435             else
436                 cout<<"\nYOUR PASSWORD UPDATED SUCCESSFULLY!!!"<<endl;
437
438         }
439
440
441         void p_bal_update();
442
443
444
```

FUNCTION FOR THE INTRODUCTION

```
448
449     atm()
450     {
451         cout << "\t\t\t  ATM";
452         cout << "\tSIMULATOR\n\n";
453
454         cout << "\n\nUnder The Guidance of Faculty :\n\n 1. Rahul Sir ";
455         cout << "\n\nPROJECT MADE BY :\n\n 1. Kushal Jain" << endl;
456         cout << " 2. Lakshay" << endl;
457         cout << "\n\n\nBranch : Software Engineering \n";
458         cout << "\nSEMESTER : 2nd";
459         cout << "\n\nCOLLEGE : Delhi Technological University";
460         cout << endl
461              << endl
462              << "\n\nPress any key...";
463         getch();
464         system("cls");
465     }
466 };
467
```

FUNCTION FOR UPDATING THE BALANCE WHEN THE TRANSACTION IS COMPLETE

```
477     int bal_update()
478     {   ifstream in;
479         ofstream out;
480         in.open("bal.txt");
481         out.open("temp2.txt");
482
483         while (!in.eof())
484         {
485             in.getline(line, 255);
486
487             for (int i = 0; i < 4; i++)
488             {
489                 if (line[i] == pass[i])
490                     flag = 0;
491
492                 else
493                 {
494                     flag = 1;
495                     break;
496                 }
497             }
498
499             if (flag == 1)
500                 out << line << endl;
501
502             else
503             {
504                 out << "\n"
505                     << strtok(line, "%") << "%" << at << "%";
506                 for (int i = 0; i < 9; i++)
507                     out << strtok(NULL, "%") << "%";
508                 out << endl;
509             }
510         }
511         in.close();
512         out.close();
513         remove("bal.txt");
514         rename("temp2.txt", "bal.txt");
515     }
```

FUNCTION FOR WITHDRAWING THE AMOUNT

```
521     void withdrawl()
522     {
523         int f = 0;
524         ifstream in;
525         ofstream out;
526         in.open("atm.txt");
527         out.open("temp1.txt");
528         while (!in.eof())
529         {
530             in.getline(line, 255);
531             for (int i = 0; i < 4; i++)
532             {
533 
534                 if (line[i] == pass[i])
535                     flag = 0;
536                 else
537                 {   flag = 1;
538                     break;
539                 }
540             }
541             if (flag == 1)
542                 out << line << endl;
543             if (flag == 0 && f == 0)
544             {
545                 out.close();
546                 enter_w();
547 
548                 out.open("temp1.txt", ios::app);
549                 bal_update();
550 
551                 f = 1;
552             }
553         }
554         in.close();
555         out.close();
556         remove("atm.txt");
557         rename("temp1.txt", "atm.txt");
558     }
559 
560 }; //******class "atm1" ends******//
```

FUNCTION TO DISPLAY RECORD OF LAST 10 TRANSACTIONS

```
566  void atm ::trans()
567  {
568      ifstream in;
569      in.open("bal.txt");
570      if (!in)
571      {
572          cout << "\nRecord File not found...!!!!"<<endl;
573          return;
574      }
575      while (!in.eof())
576      {
577          in.getline(line, 255);
578          for (int i = 0; i < 4; i++)
579          {
580              if (line[i] == pass[i])
581                  flag = 0;
582              else
583              {
584                  flag = 1;
585                  break;
586              }
587          }
588          if (flag == 0)
589              break;
590      }
591
592      if (flag == 0)
593      {
594          strtok(line, "%");
595
596          for (int i = 1; i <= 10; i++)
597              cout << "Rs. " << strtok(NULL, "%") <<endl;
598      }
599      in.close();
600  }
601
```

FUNCTION FOR UPDATING THE PASSWORD IN THE BAL.TXT FILE

```
606  void atm :: p_bal_update()
607  {
608
609      ifstream in;
610      ofstream out;
611
612      in.open("bal.txt");
613      out.open("temp2.txt");
614
615      while(!in.eof())
616      {
617          in.getline(line,255);
618          for(int i=0;i<4;i++)
619          {
620              if(line[i] == pin_no[i])
621                  flag=0;
622              else
623              {
624                  flag=1;
625                  break;
626              }
627          }
628          if(flag==1)
629              out<<line<<endl;
630           else
631          {
632              strtok(line,"%");
633              out<<"\n"<<pno<<"%";
634
635              for(int i=0;i<10;i++)
636                  out<<strtok(NULL,"%")<<"%";
637              out<<endl;
638          }
639      }
640      in.close();
641      out.close();
642      remove("bal.txt");
643      rename("temp2.txt","bal.txt");
644  }
```

FUNCTION FOR TIME DELAY

```
649
650  void wait(int seconds)
651  {
652      clock_t endwait;
653      endwait = clock() + seconds * CLOCKS_PER_SEC;
654      while (clock() < endwait)
655      {
656      }
657  }
658
```

THE MAIN FUNCTION

```
666  int main()
667  {
668      atm1 ob;
669      int n,i,count;
670
671   cout<<"\n\nInsert Your Card\n";
672   cout<<"\nVerifying...";
673
674     for (i=1;i>0;i--)
675       {
676        wait(1);
677       }
678
679   cout<<"\n\nCard Accepted\n\n";
680
681   count = 1;
682  while(1)
683  {
684
685  cout<<"\n\n\n\t\tENTER YOUR PIN: ";
686
687  char c= ' ';
688  pass="";
689
690  while(c!=13)
691  {
692  c=getch();
693
694  if(c!=13)
695  {
696  pass+=c;
697  cout<<"%";
698  }
699
700  }
701
702  ifstream in;
703  in.open("atm.txt");
```

```
706  if(!in)
707     {
708      cout<<"\nRecord File not found...!!!!"<<endl;
709      exit(0);
710     }
711
712     while(!in.eof())
713      {
714       in.getline(line,255);
715
716       for(int i=0;i<4;i++)
717        {
718         if(line[i] == pass[i])
719         flag=0;
720
721          else
722           {
723            flag=1;
724            break;
725           }
726        }
727
728        if(flag==0)
729        {
730        flag2=1;
731        break;
732        }
733
734       }
735
736  in.close();
737
738       if(flag2==0)
739       {
740
741       if(count<=3)
742         {
743         cout<<"\n\nIncorrect PIN"<<endl;
744         count++;
745
```

```
746             if(count>3)
747             {
748             cout<<"\n\nNo.of trials for entering the PIN is over and your card is now blocked"<<endl;
749             getch();
750             exit(0);
751             }
752
753             if(count == 3)
754             cout<<"\n\nThis is the Last Attempt to enter your correct PIN otherwise : \n\n\t\t YOUR CARD WILL BE BLOCKED\n\n";
755
756           }
757         }
758         else
759         break;
760 ▼ }
761
762       while(1)
763 ▼     {
764               pin_no = pass;
765
766               //**********************DISPLAY MENU**********************//
767
768               system("cls");
769               cout<<"\n\n\t\t\t\tMAIN MENU"<<endl;
770               cout<<"\n\n\n\t\t1. Balance\t\t2. Cash Withdrawal\n\n\t\t3. Mini Statement\t";
771               cout<<"4. Account Details\n\n\t\t5. Change PIN\t\t6. Exit\n\n\nEnter your Choice : ";
772
773               cin>>n;
774
775               switch(n)
776               {
777 ▼             case 1:
778                     system("cls");
779                     ob.balance();
780
781                     for(i=1;i>0;i--)
782                     {
783                      wait(5);
784                     }
```

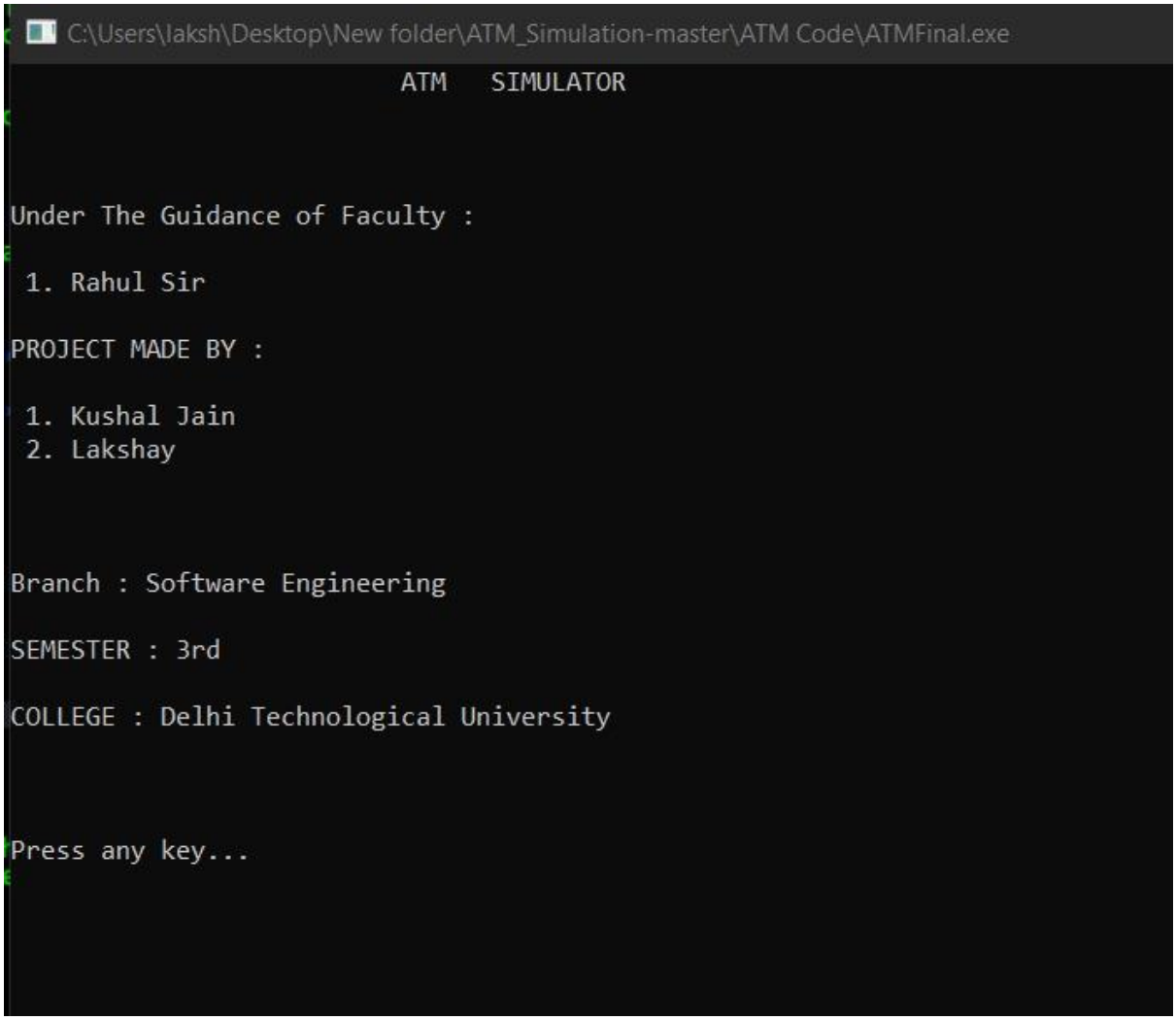
```
785             break;
786
787 ▼           case 2:
788                 system("cls");
789                 ob.withdrawl();
790
791                 for(i=2;i>0;i--)
792                 {
793                  wait(3);
794                 }
795                 system("cls");
796
797                 ob.balance();
798
799                 for(i=2;i>0;i--)
800                 {
801                  wait(2);
802                 }
803                 system("cls");
804
805             break;
806
807 ▼           case 3:
808                 system("cls");
809                 cout<<"\nYour last 10 Transactions :\n\n";
810                 ob.trans();
811
812                 for(i=1;i>0;i--)
813                 {
814                  wait(5);
815                 }
816
817             break;
818
819 ▼           case 4:
820                 system("cls");
821                 cout<<"Your Account details:"<<endl;
822                 ob.search();
823
```

```
                    for(i=1;i>0;i--)
                    {
                      wait(5);
                    }

               break;

               case 5:
                    system("cls");
                    ob.update();
                    ob.p_bal_update();

                    for(i=1;i>0;i--)
                    {
                     wait(5);
                    }

               break;

               case 6:
                    system("cls");
                    cout<<"Thank You for using the ATM simulator...";

                    int n;
                    for(i=1;i>0;i--)
                    {
                      wait(1);
                    }

                    cout<<"\n\nExiting...";

                    for(i=2;i>0;i--)
                    {
                     wait (1);
                    }
                    exit(0);

               break;

```

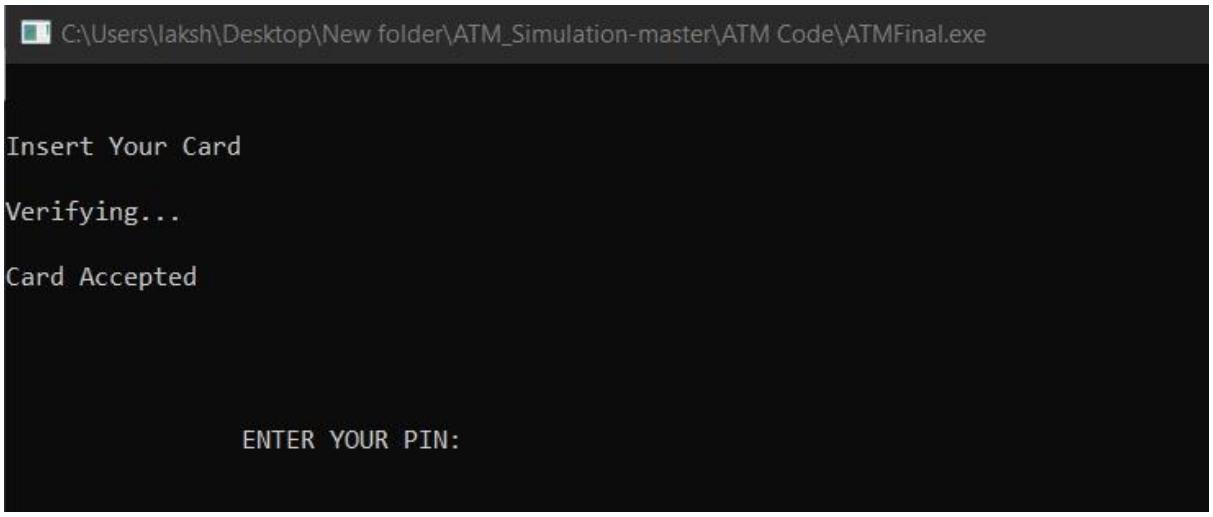
```
               default :
                       cout<<"\nInvalid Choice"<<endl;
                       for(i=1;i>0;i--)
                       {
                        wait(1);
                       }

               }
     }

getch();

}
```

OUTPUT SCREENS

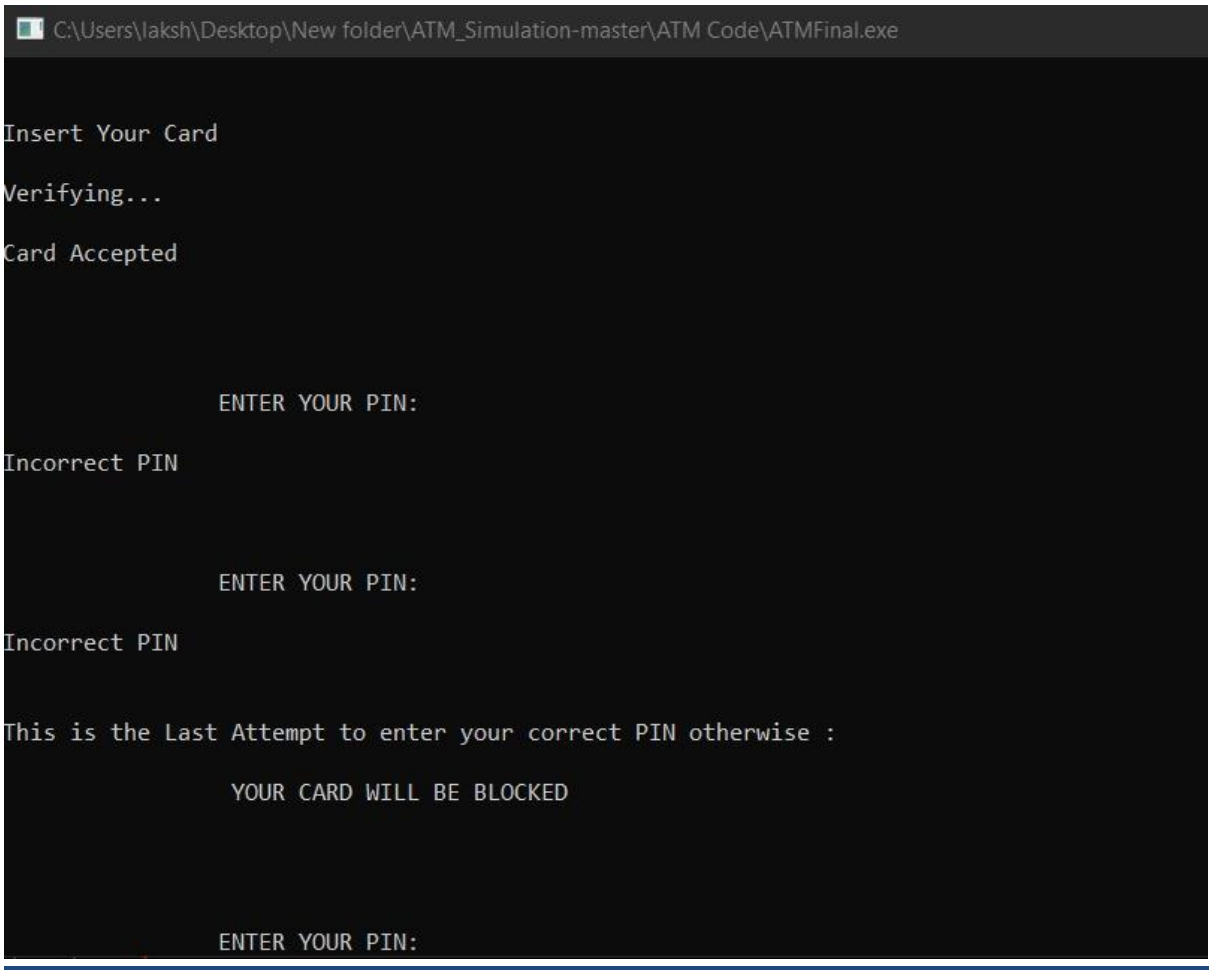
Intro screen



Accepting PIN

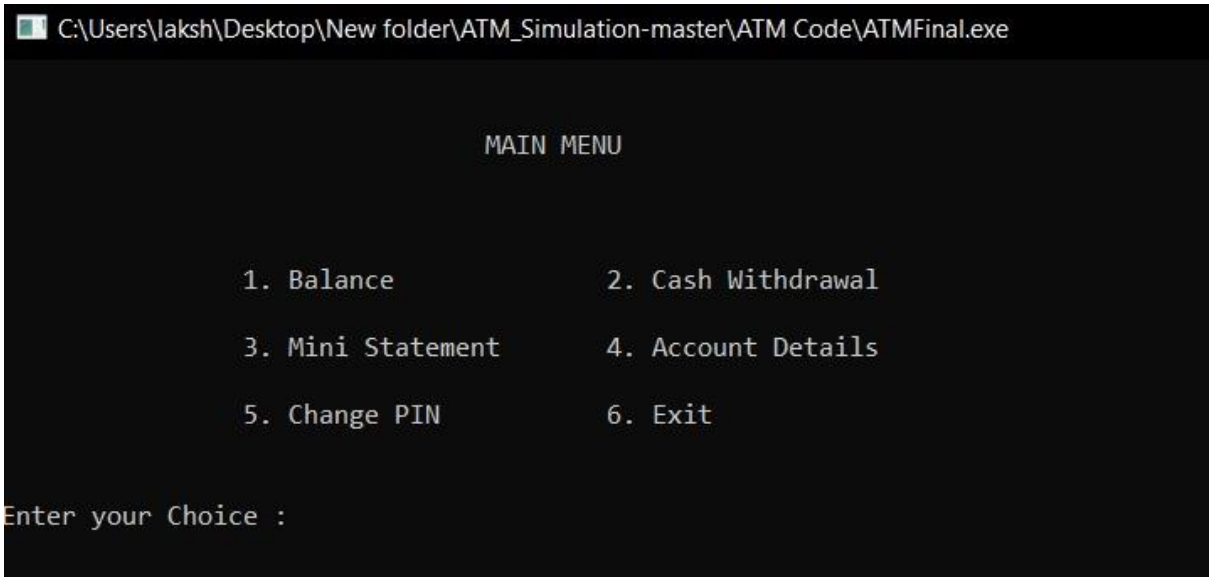


Wrong Attempt

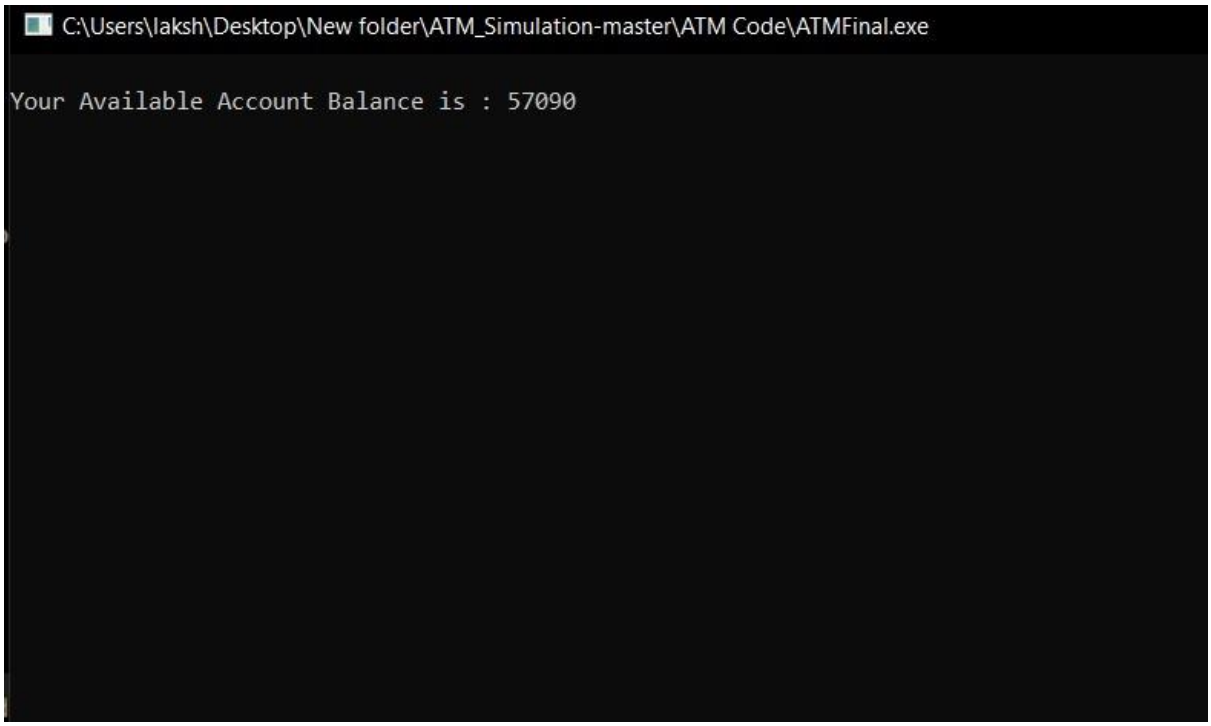


Card Blocked after 3 unsuccessful attempts

Main Menu

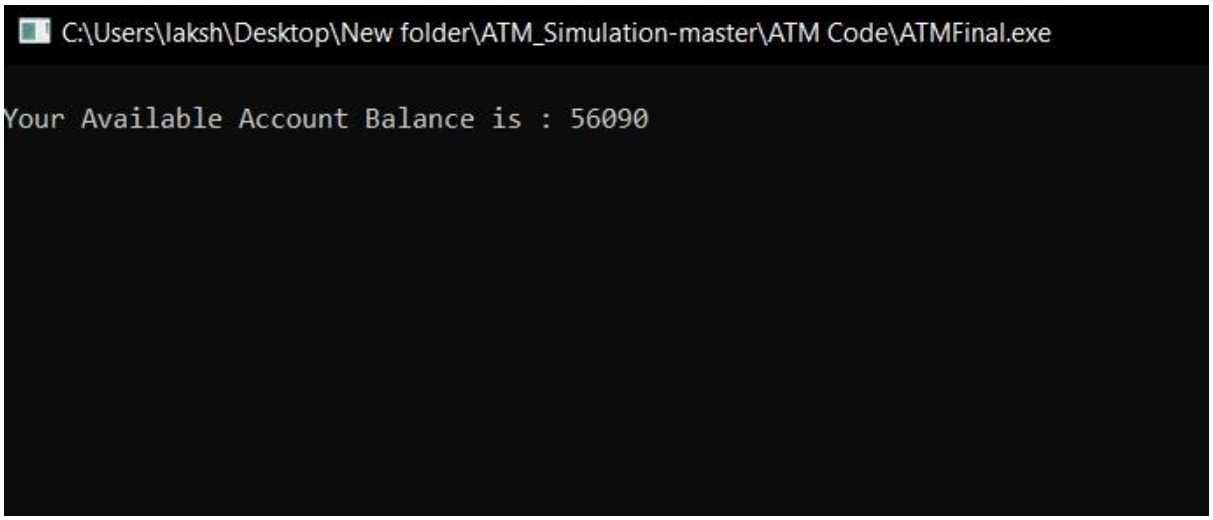


Balance

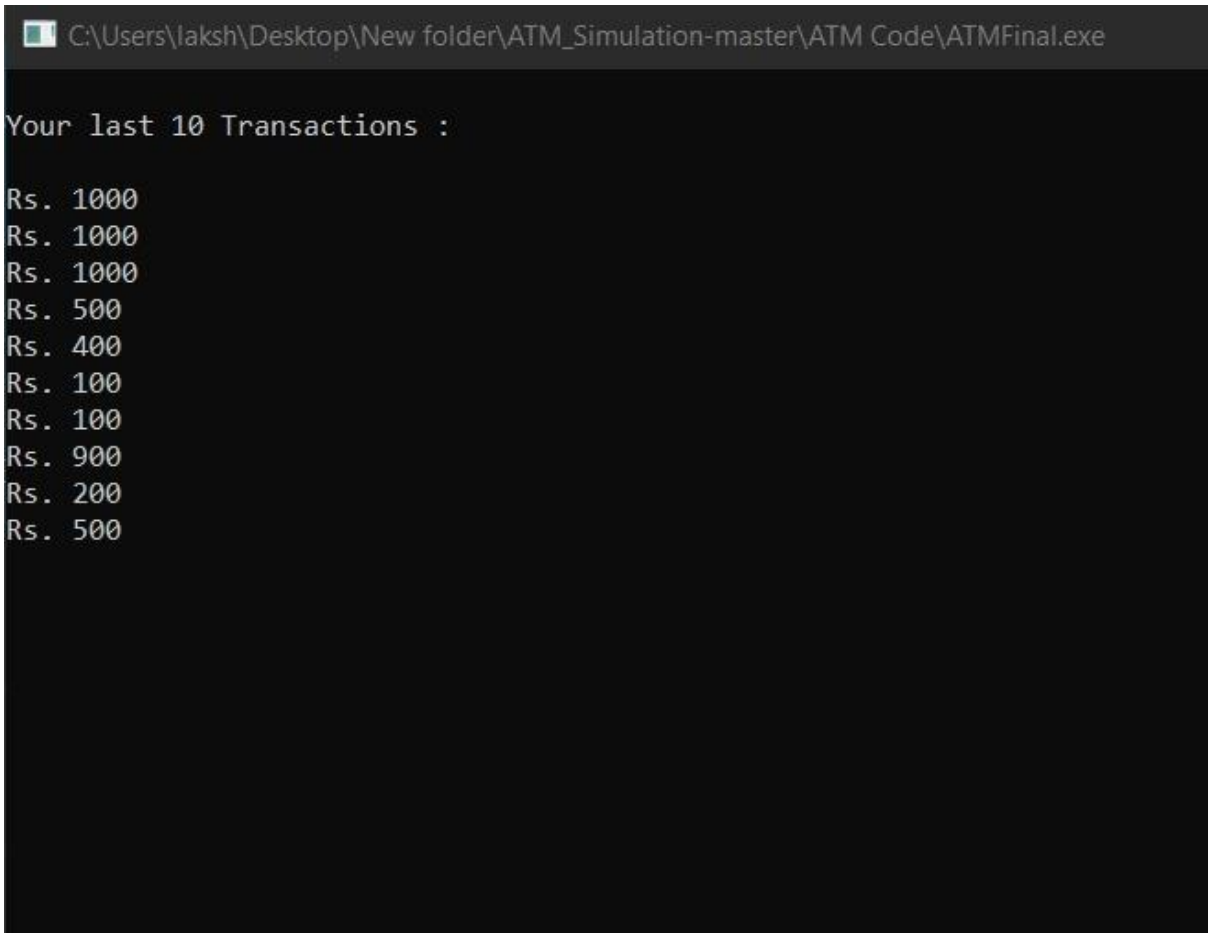


Cash Withdrawal

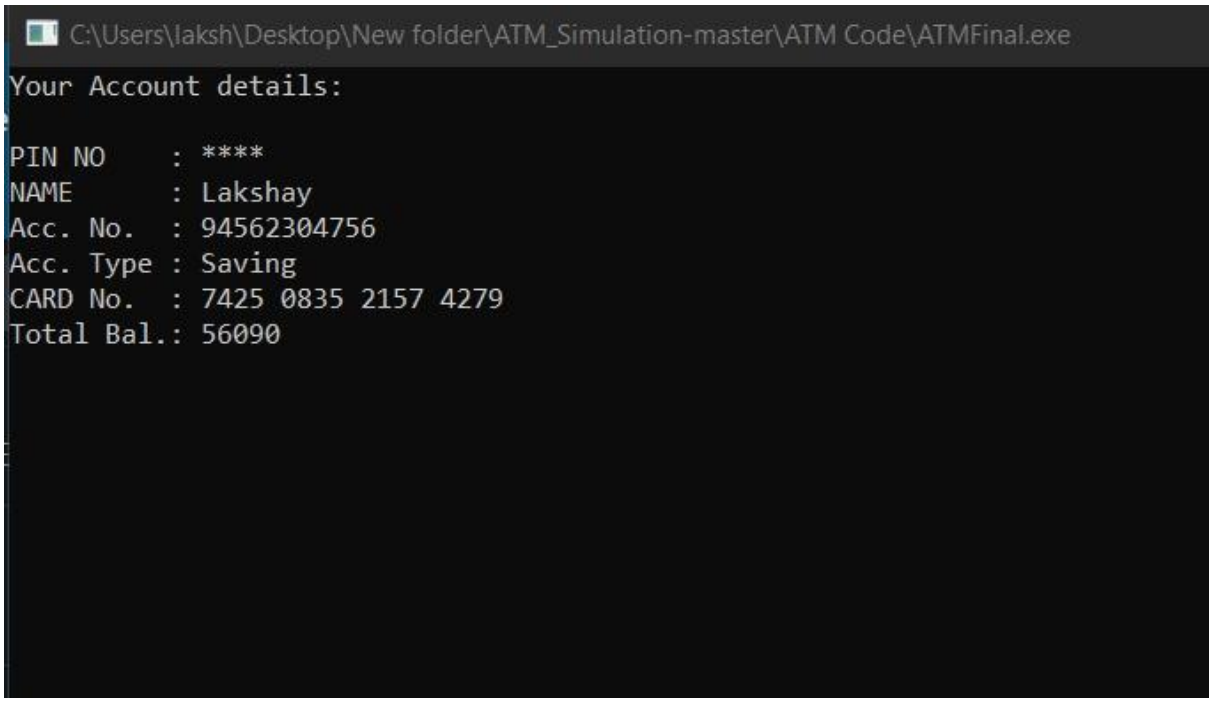
Balance after Cash Withdrawal



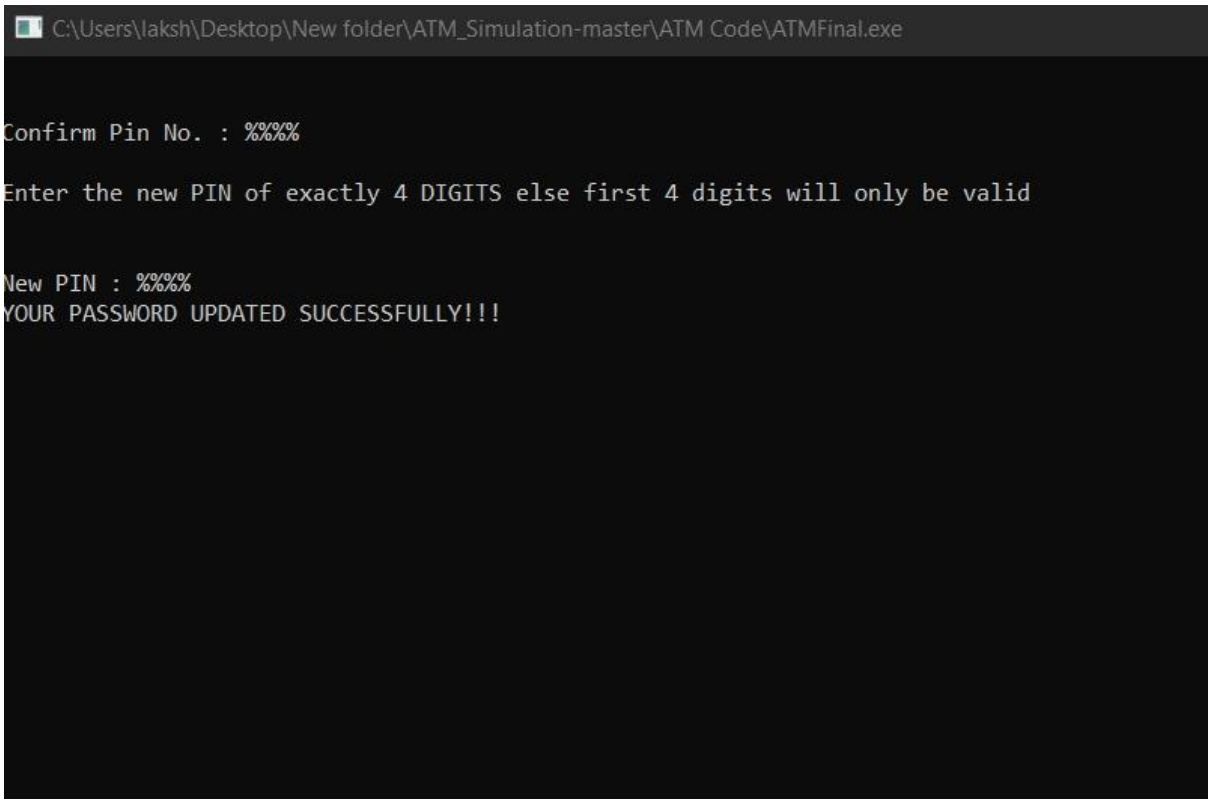
Mini Statement



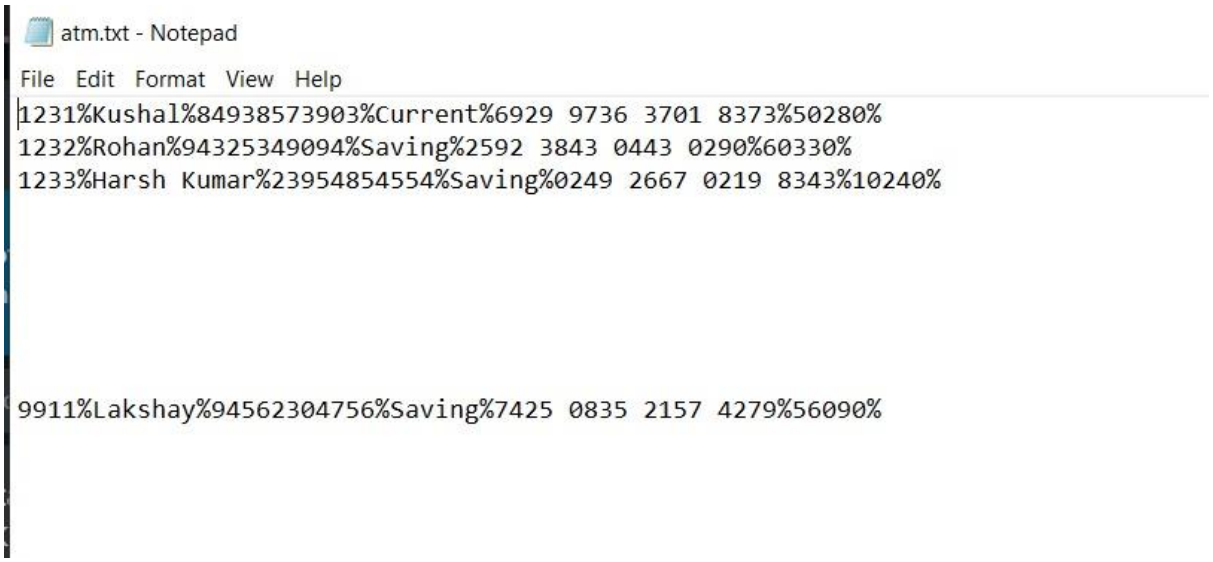
Account Details



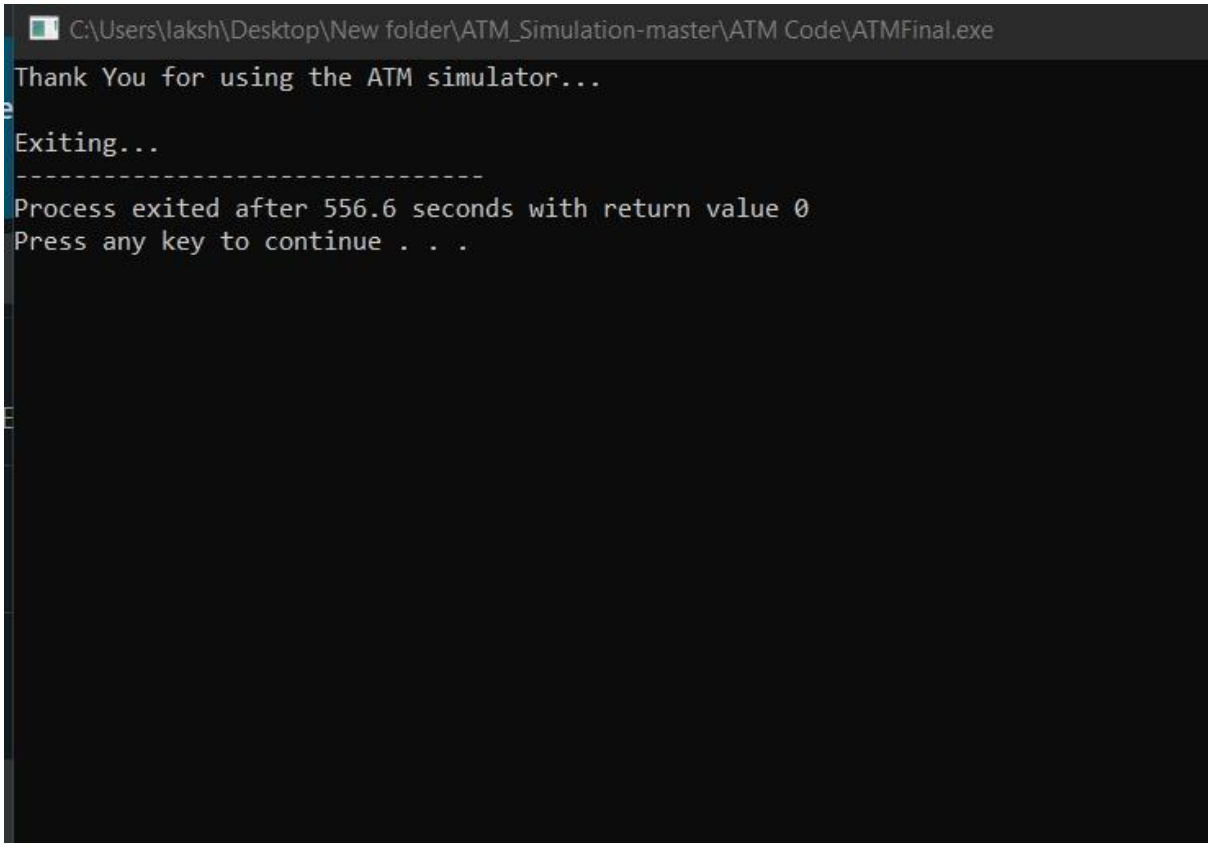
PIN Change



Changes in Text File After PIN Change



Exit Screen



REFERENCES

- https://stackoverflow.com
- https://ieeexplore.ieee.org/abstract/document/5607693
- https://www.motc.gov.qa/en/ditoolkit/migrant-workers/cash-machine-simulator-atm
- GeeksforGeeks | A computer science portal for geeks