

</> Computer Project

ATM



Submitted By:

Name: Rishi Tiku
Class: XII-C
Roll no:

Under The Supervision of:

Mrs. Poonam Makhija
PGT, Computer Science

</> Certificate

This is to certify that **Rishi Tiku** of class **XII – C** bearing Roll No. _____ of **GREENWAY MODERN SCHOOL** has completed the Computer Science project on ‘**ATM**’ successfully under the guidance of **Mrs. Poonam Makhija** during the academic year 2019 – 20.

I wish him all successes for the board.

Mrs. Poonam Makhija
(Department of
Computer Science)

</> Acknowledgements

I would like to begin my project in a prayer to all those who acted like Gods and helped me make this a project a success.

I thank my Computer Science teacher and mentor, Mrs. Poonam Makhija, for her exemplary guidance that has helped me to program the way I can today. I believe she is one of the best teachers I've come across. I sincerely wish for the incoming batches to be blessed by her guidance. I also thank our Lab Assistant, Mrs. Suman Sahni, who cared for me like a mother at my most difficult moments.

I would like to thank my parents, for their hardwork, that makes every moment of my life a celebration. I shall always remain indebted to what you've done for me. I would also thank my friends for their useful insights and advice on this project.

I also thank the reader who spent their time reading the page that isn't generally read by most, if not all.

At last I would like to thank Almighty God for giving me the strength to complete this project.

-Rishi Tikun

</> Index

S No.	Topic	Page No.
1	Certificate	2
2	Acknowledgements	3
3	Index	4
4	C++ and Its Advantages	5
5	About Turbo C++	6
6	About ATMs	7
7	Flow Chart	8
8	Header Files Used	9
9	Program	10
10	Output	47
11	Conclusion	61
12	Sources of Error	62
13	Bibliography	63

</> C++ and Its Advantages

C++ is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or "C with Classes". The language has expanded significantly over time, and modern C++ has object-oriented, generic, and functional features in addition to facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers, including the Free Software Foundation, LLVM, Microsoft, Intel, Oracle, and IBM, so it is available on many platforms.

C++ was designed with a bias toward system programming and embedded, resource-constrained software and large systems, with performance, efficiency, and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce, Web search, or SQL servers), and performance-critical applications (e.g. telephone switches or space probes).

C++ is standardized by the International Organization for Standardization (ISO), with the latest standard version ratified and published by ISO in December 2017 as *ISO/IEC 14882:2017* (informally known as C++17). The C++ programming language was initially standardized in 1998 as *ISO/IEC 14882:1998*, which was then amended by the C++03, C++11 and C++14 standards. The current C++17 standard supersedes these with new features and an enlarged standard library. Before the initial standardization in 1998, C++ was developed by Danish computer scientist Bjarne Stroustrup at Bell Labs since 1979 as an extension of the C language; he wanted an efficient and flexible language similar to C that also provided high-level features for program organization. C++20 is the next planned standard, keeping with the current trend of a new version every three years.

</> About Turbo C++

Turbo C++ is a discontinued C++ compiler and integrated development environment originally from Borland. It was designed as a home and hobbyist counterpart for Borland C++. As the developer focused more on professional programming tools, later Turbo C++ products were made as scaled down versions of its professional compilers.

The first release of Turbo C++ was made available during the MS-DOS era on personal computers. Version 1.0, running on MS-DOS, was released in May 1990. An OS/2 version was produced as well. Version 1.01 was released on February 28, 1991, running on MS-DOS. The latter was able to generate both COM and EXE programs and was shipped with Borland's Turbo Assembler compiler for Intel x86 processors. The initial version of the Turbo C++ compiler was based on a front end developed by TauMetric (TauMetric was later acquired by Sun Microsystems and their front end was incorporated in Sun C++ 4.0, which shipped in 1994). This compiler supported the AT&T 2.0 release of C++.

Turbo C++ 3.0 was released in 1991 (shipping on November 20), and came in amidst expectations of the coming release of Turbo C++ for Microsoft Windows. Initially released as an MS-DOS compiler, 3.0 supported C++ templates, Borland's inline assembler, and generation of MS-DOS mode executables for both 8086 real mode and 286 protected mode (as well as the Intel 80186.) 3.0 implemented AT&T C++ 2.1, the most recent at the time. The separate Turbo Assembler product was no longer included, but the inline-assembler could stand in as a reduced functionality version.

</> About ATMs

An **automated teller machine (ATM)** is an electronic telecommunications device that enables customers of financial institutions to perform financial transactions, such as cash withdrawals, deposits, transfer funds, or obtaining account information, at any time and without the need for direct interaction with bank staff.

ATMs are known by a variety of names, including automatic teller machine (ATM) in the United States (sometimes redundantly as "ATM machine"). In Canada, the term *automated banking machine* (ABM) is used, although ATM is also very commonly used in Canada, and many Canadian organizations use ATM over ABM. In British English, the terms *cashpoint*, *cash machine* and *hole in the wall* are most widely used. Other terms include *any time money*, *cashline*, *nibank*, *tyme machine*, *cash dispenser*, *cash corner*, *bankomat*, or *bancomat*. Many ATMs have a sign above them indicating the name of the bank or organisation that owns the ATM, and possibly including the networks to which it can connect. ATMs that are not operated by a financial institution are known as "white-label" ATMs.

Using an ATM, customers can access their bank deposit or credit accounts in order to make a variety of financial transactions, most notably cash withdrawals and balance checking, as well as transferring credit to and from mobile phones. ATMs can also be used to withdraw cash in a foreign country. If the currency being withdrawn from the ATM is different from that in which the bank account is denominated, the money will be converted at the financial institution's exchange rate. Customers are typically identified by inserting a plastic ATM card (or some other acceptable payment card) into the ATM, with authentication being by the customer entering a personal identification number (PIN), which must match the PIN stored in the chip on the card (if the card is so equipped), or in the issuing financial institution's database.

</> Flow Chart

</>Header Files Used

- <iostream.h>
- <conio.h>
- <stdio.h>
- <string.h>
- <fstream.h>
- <process.h>
- <dos.h>
- <time.h>
- <math.h>
- <ctype.h>

</> Program

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <fstream.h>
```

```
#include <process.h>
```

```
#include <dos.h>
```

```
#include <time.h>
```

```
#include <math.h>
```

```
#include <ctype.h>
```

```
char * A[3] = {"Where were you Born?",  
              "What is your Favorite Dish?",  
              "What Book did you first read?"};
```

```
int mIndex = 1;
```

```
class ATM
```

```
{
```

```
    float money;
```

```
    public:
```

```
        friend int main();
```

```
        ATM()
```

```
{
    Update(1);
}
void Deposit()
{
    float add;
    cout<<"Enter Money to be added!\n";
    cin>>add;
    money = money + add;
    Update(2);
}
void Withdraw(float sub)
{
    money = money - sub;
}
int WithdrawCheck(float sub)
{
    if(sub>money)
    {
        cout<<"Not Enough Money in the"
        <<" Machine! Try a different branch.\n";
        return 0;
    }
    else
        return 1;
}
```

```
}  
int MoneyCheck()  
{  
    if(money<=20000)  
        return 0;  
    else  
        return 1;  
}  
void Update(int a)  
{  
    if(a==1)  
    {  
        file:  
        ifstream fin("Money.txt",ios::in);  
        if(!fin)  
        {  
            ofstream fout("Money.txt",ios::out);  
            fout<<5000000;  
            fout.close();  
            goto file;  
        }  
        fin>>money;  
        fin.close();  
    }  
    else if(a==2)
```

```
        {
            ofstream fout("Money.txt",ios::out);
            fout<<money;
            fout.close();
        }
    }

    float Money()
    {return money;}

}atm;
```

```
class Person
{
    float amount;
    long accountNumber;
    long PIN;
    char address[40];
    float rate;
    int priority;
    int securityQuestion;
    char securityAnswer[50];
    public:
        char name[30];
        int seniorCitizen;
        friend int main();
        friend Person Login(int);
```

```
Person(long a = 0)
{
    accountNumber = a;
    securityQuestion = -1;
}
float Amount()
{return amount;}
int Priority()
{
    if(amount>=1000000 || seniorCitizen==1)
        priority = 1;
    else
        priority = 0;
    return priority;
}
float Rate()
{
    if(Priority()==1)
        rate = 7;
    else
        rate = 6.5;
    return rate;
}
char * Address()
{return address;}
```

```
long Pin()
{return PIN;}
void Withdraw(float sub)
{
    amount = amount - sub;
}
void Deposit(float sub)
{
    amount = amount + sub;
}
int WithdrawCheck(float sub)
{
    if(sub>amount)
    {
        cout<<"Not Enough Money in your Account!
Try adding some money first!\n";
        return 0;
    }
    else
        return 1;
}
void ChangeAddress()
{
    cout<<"Enter a new Address!\n";
    cin.getline(address,40);
```

```
    }  
    void Input()  
    {  
        char ch;  
        cout<<"Enter Name!\n";  
        cin.getline(name,30);  
        cout<<"Enter Amount!\n";  
        cin>>amount;  
        cin.ignore();  
        cout<<"Enter Home Address!\n";  
        cin.getline(address,40);  
        cout<<"Senior Citizen?(y/n)\n";  
        cin>>ch;  
        if(ch=='Y' || ch=='y')  
            seniorCitizen=1;  
        else  
            seniorCitizen=0;  
        if(Priority()==1 || Rate()==7)  
            cout<<"Eligible for Priority Service!\n";  
        cin.ignore();  
        cout<<"Enter a PIN. (Numbers Only)\n";  
        cin>>PIN;  
        cout<<"Account Number:  
"<<accountNumber<<"\nPIN: ";  
        cout<<PIN;
```



```
        cout<<"\nDo you wish to have your Security
Question Backup?(y/n)\n";
        cin>>ch;
        if(ch=='Y' || ch=='y')
            SecurityQuestion();
    }
    void ChangePIN()
    {
        long p1,p2;
        cout<<"Enter your old PIN.\n";
        cin>>p1;
        if(p1==PIN)
        {
            cout<<"Enter a New PIN.\n";
            cin>>p1;
            cout<<"Confirm PIN.\n";
            cin>>p2;
            if(p1==p2)
                PIN = p1;
            else
                cout<<"PINs do not match! Aborting!";
        }
        else
            cout<<"Incorrect PIN entered.\n";
    }
```

```
void SecurityQuestion()
{
    int choice;
    char ch, answer[40];
    if(securityQuestion!=-1)
    {
        cout<<"You already have answered a Previous
Security Question!"
        <<" Do you wish to continue?(y/n)\n";
        cin>>ch;
        cin.ignore();
        if(ch!='Y'&&ch!='y')
        {
            return;
        }
    }
    start:
    cout<<"Choose a Preference.(1-3)\n";
    for(int i = 0;i<=2;i++)
    {
        cout<<i+1<<" ";
        puts(A[i]);
    }
    cin>>choice;
    cin.ignore();
```

```
if(choice>=1&&choice<=3)
{
    Question:
    securityQuestion = choice-1;
    cout<<"You chose Question "<<choice<<", ";
    puts(A[choice-1]);
    cout<<"Enter an Answer!\n";
    cin.getline(answer,50);
    cout<<"You entered ";
    puts(answer);
    cout<<"Do you wish to change?(y/n)\n";
    cin>>ch;
    cin.ignore();
    if(ch=='y' || ch=='Y')
        goto Question;
    else
        strcpy(securityAnswer,answer);
}
else
{
    cout<<"Invalid choice entered. Try Again!\n";
    goto start;
}
}

int RecoverPIN()
```

```
    {  
        char answer[40];  
        if(securityQuestion== -1)  
        {  
            cout<<"Uh Oh! You haven't set up a recovery  
option yet."  
            <<" Go to Customer Service for a solution!\n";  
            return 0;  
        }  
        puts(A[securityQuestion]);  
        cin.getline(answer,40);  
        if(strcmpi(securityAnswer,answer)==0)  
        {  
            cout<<"Success!\n";  
            cout<<"Name:\t";  
            puts(name);  
            cout<<"Account Number:\t"<<accountNumber;  
            cout<<"\nPIN: ";  
            cout<<PIN;  
            cout<<"\nPress any key to continue.\n";  
            getch();  
            return 1;  
        }  
        else  
        {
```

```
        cout<<"Please Try Again in a different session"
        <<"! Quitting!\n";
        return 0;
    }
}

void Display()
{
    cout<<"Account Number:\t"<<accountNumber;
    cout<<"\nName:\t";
    puts(name);
    cout<<"Rate:\t"<<Rate()<<endl;
    cout<<"Amount: "<<amount<<" Rupees!\n";
    if(Priority()==1)
        cout<<"Priority Account!\n";
    else
        cout<<"Regular Account!\n";
}

int AccountNumber()
{return accountNumber;}

~Person()
{
}

};

int getPIN(Person P)
{
```

```
long pwd;
cout<<"Enter PIN!\n";
for(int i = 2;i>=0;i--)
{
    cin>>pwd;
    cin.ignore();
    if(P.Pin()==pwd)
    {
        cout<<"Success!\n";
        return 1;
    }
    if(i!=0)
    {
        cout<<"Incorrect PIN! Try Again! "
        <<i<<" attempts remaining!\n";
    }
}

return 0;
} //*****

void init()
{
    ifstream fin("Number.txt",ios::in);
    if(fin)
    {
        fin.close();
    }
}
```

```
        return;
    }
    else
    {
        fin.close();
        ofstream fout("Number.txt",ios::out);
        if(fout)
        {
            fout<<0;
            fout.close();
        }
    }
}

// *****

int Number(int f=0)
{
    int a;
    init();
    ifstream fin("Number.txt",ios::in);
    fin>>a;
    fin.close();
    if(f==0)
        return a;
    a++;
    ofstream fout("Number.txt",ios::out);
    fout<<a;
```

```
fout.close();

return a;

} // *****

int Menu()
{
    Person ScanRecords(int);
    int aNum[3] = {8,6,3};
    char * menu[][10] = {"Withdraw Money.",
                        "Account Mini Statement.",
                        "Account Transaction Log.",
                        "Transfer Money.",
                        "Setup a Recovery Question.",
                        "Change PIN.",
                        "Other Options.",
                        "Exit."},
        {"Request ChequeBook.",
        "Request Credit Card.",
        "Request PassBook.",
        "Apply for a Loan.",
        "Change Address.",
        "Go Back."},
        {"Add Records.",
        "Add Money.",
        "Exit."};

    int choice;
```



```
start:
clrscr();
if(mIndex==3)
{
    cout<<"\tDeveloper mode.\n";
}
cout<<"Select from a list of options.\n";
for(int i = 0;i<aNum[mIndex-1];i++)
{
    cout<<i+1<<". ";
    puts(menu[mIndex-1][i]);
}
cout<<"Enter your choice. (1-<<aNum[mIndex-1]<<")\n";
cin>>choice;
if(choice==0)
    return 0;
else if(choice>0&&choice<=aNum[mIndex-1])
    return choice;
cout<<"Wrong Choice! Try Again.\n";
delay(1000);
goto start;
} // *****
char * Current()
{
    time_t tt;
```

```
    struct tm * ti;
    time(&tt);
    ti = localtime(&tt);
    return asctime(ti);
} // *****

void AccountInfo(Person P)
{
    char ch;

    cout<<"Would you like to have a reciept for the
transaction?(y/n)\n";
    cin>>ch;
    void LogCheck(int,int);
    if(ch=='y' || ch=='Y')
    {
        P.Display();
        cout<<"Time of Transaction: "<<Current();
        cout<<"Statement: ";
        cprintf(P.name);
        LogCheck(P.AccountNumber(),1);
    }
} // *****

void Log(int ac, char c, float money=0,int ac2=0)
{
    ofstream fout("Log.txt",ios::app);
    ofstream fout1("LogHolde.txt",ios::app);
    Person ScanRecords(int);
```

```
char * str,*str2;
int len;
switch(c)
{
    case 'w': strcpy(str," withdrew ");
                len = 10;
                break;
    case 't': strcpy(str," transferred ");
                len = 13;
                break;
    case 'r': strcpy(str," recieved ");
                len = 10;
                break;
    case 'd': strcpy(str,"Credit Card ");
                len = 12;
                break;
    case 'p': strcpy(str,"PassBook ");
                len = 9;
                break;
    case 'l': strcpy(str,"Loan ");
                len = 5;
                break;
    case 'c': strcpy(str,"ChequeBook ");
                len = 11;
                break;
```

```
        case 'a': strcpy(str,"change in Address ");
                    len = 18;
                    break;
    }
    if(money>0)
    {
        fout.write(str,len);
        fout<<money<<" Rupees";
        if(c=='t')
        {
            char* name2 = ScanRecords(ac2).name;
            fout<<" to "<<name2;
        }
        else if(c=='r')
        {
            char* name2 = ScanRecords(ac2).name;
            fout<<" from "<<name2;
        }
        fout1<<ac<<'\n';
        fout<<" on "<<Current()<<'\n';
    }
    else if(money==0)
    {
        fout<<" has requested for a ";
        fout.write(str,len);
    }
}
```

```
        fout<<"on "<<Current();
        fout<<'\\n';
        fout1<<ac<<'\\n';
    }
    fout.close();
    fout1.close();
} // *****
void LogCheck(int accn,int rev = 0)
{
    Person ScanRecords(int);
    ifstream fin("Log.txt",ios::in);
    ifstream fin1("LogHolde.txt",ios::in);
    if(!fin || !fin1)
    {
        cout<<"Error! Log File Missing. Press any Key to
continue.\\n";
        fin.close();
        fin1.close();
        return;
    }
    char str[80], ch,*log,* name = ScanRecords(accn).name;
    int x = 0, ctr = 0, num = 0, * N;
    do
    {
        ctr = 0;
```

```
num = 0;
fin1.getline(str,10);
for(ctr=0;isdigit(str[ctr]);ctr++)
{
    ch = str[ctr];
    x = ch - '0';
    N[ctr] = x;
}
ctr--;
for(int i = ctr;i>=0;i--)
{
    num = num+(N[i]*pow(10,ctr-i));
}
fin.getline(str,80);
if(num==accn)
{
    strcpy(log,str);
    if(rev==0)
    {
        cprintf(name);
        cprintf(str);
        cout<<endl;
    }
}
fin.getline(str,18);
```

```
        continue;
    }while(!fin1.eof());
    if(rev==1)
    {
        cprintf(log);
        cout<<endl;
    }
    fin.close();
    fin1.close();
    return;
} // *****

int Update(Person P)
{
    Person P1;
    int flag = 0;
    if(P.AccountNumber()<=0 || P.AccountNumber()>Number())
    {
        cout<<"Bad Account Number!";
        return 0;
    }
    ifstream fin("Records.dat",ios::in | ios::binary);
    ofstream fout("temp.dat",ios::out | ios::binary);
    if(!fin || !fout)
    {
        cout<<"Couldn't Update!\n";
    }
}
```

```
        return 0;
    }
    while(!fin.eof())
    {
        fin.read((char*)&P1,sizeof(P1));
        if(P1.AccountNumber()==P.AccountNumber()&&flag==0)
        {
            fout.write((char*)&P,sizeof(P));
            flag = 1;
            continue;
        }
        else
            if(P1.AccountNumber()==P.AccountNumber()&&flag==1)
            {
                continue;
            }
            else
                fout.write((char*)&P1,sizeof(P1));
    }
    if(flag==0)
    {
        cout<<"Record Not Found!\n";
    }
    fout.close();
    fin.close();
```



```
        remove("Records.dat");
        rename("temp.dat","Records.dat");
        return 1;
} // *****

Person Withdraw(Person P)
{
    float sub;
    cout<<"Enter Money to be Withdrawn!\n";
    cin>>sub;
    if(P.WithdrawCheck(sub))
        if(atm.WithdrawCheck(sub))
        {
            cout<<"Here is the Cash! "<<sub<<" Rupees!\n";
            atm.Withdraw(sub);
            P.Withdraw(sub);
            atm.Update(2);
            Update(P);
            Log(P.AccountNumber(),'w',sub);
            return P;
        }
    Person P1;
    return P1;
} // *****

Person Transfer(Person P1)
{
    Person ScanRecords(int);
```

```
    Person P;
    float money;
    int accn;
    char ch;

    cout<<"Enter the Account Number in which you wish to
transfer.\n";
    cin>>accn;
    if(accn==P1.AccountNumber())
    {
        cout<<"Can't transfer to self. Aborting!\n";
        return P;
    }
    Person P2 = ScanRecords(accn);
    if(P2.AccountNumber()==0)
    {
        cout<<"The Requested Account Number Doesn't
Exist!\n";
        return P;
    }
    cout<<"Enter Amount of money to be transferred.\n";
    cin>>money;
    if(P1.WithdrawCheck(money))
    {
        cout<<"You have "<<P1.Amount()<<" Rupees in your
account."
```

```
<<"\nAre you sure you want to transfer "<<money<<"
Rupees to ";
    cprintf(P2.name);
    cout<<"\n's account?(y/n)\n";
    cin>>ch;
    cin.ignore();
    if(ch=='Y' || ch=='y')
    {
        P1.Withdraw(money);
        P2.Deposit(money);
        cout<<money<<" Rupees transferred
successfully.\n";
        Update(P1);
        Update(P2);

        Log(P1.AccountNumber(),'t',money,P2.AccountNumber());

        Log(P2.AccountNumber(),'r',money,P1.AccountNumber());
        return P1;
    }
    else
        return P;
}
else
{
    cout<<"Not Enough Money in your Account. Try a lower
amount!\n";
```

```
        return P;
    }
} // *****
int Execute(Person P,int choice)
{
    Person ScanRecords(int);
    int i = 0;
    char ch;
    if(mIndex==1)
    {
        switch(choice)
        {
            case 1: P = Withdraw(P);
                    if(P.AccountNumber())
                        AccountInfo(P);
                    break;
            case 2: P.Display();
                    break;
            case 3:  LogCheck(P.AccountNumber());
                    break;
            case 4: P = Transfer(P);
                    if(P.AccountNumber())
                        AccountInfo(P);
                    break;
            case 5: P.SecurityQuestion();
```

```
        Update(P);
        break;
    case 6: P.ChangePIN();
        Update(P);
        break;
    case 7: cout<<"Accessing!\n";
        delay(500);
        break;
    case 8: cout<<"Quitting Session! Have a Nice Day!\n";
        break;
}
}
else if(mIndex==2)
{
    switch(choice)
    {
        case 1: cout<<"Your Selected Address is ";
            cprintf(P.Address());
            cout<<".\nDo you wish to Procure a
ChequeBook?(y/n)\n";
            cin>>ch;
            if(ch=='y' || ch=='Y')
            {cout<<"Request Processed Succesfully! "
<<"It will Arrive in the Mail in about 48 hours.";
            Log(P.AccountNumber(),'c');}
```

```
        break;
    case 2: cout<<"Your Selected Address is ";
            cprintf(P.Address());
            cout<<".\nDo you wish to Request a Credit
Card?(y/n)\n";
            cin>>ch;
            if(ch=='y' || ch=='Y')
            {cout<<"Request Processed Succesfully! "
            <<"It will Arrive in the Mail in about 48 hours.";
            Log(P.AccountNumber(),'d');}
            break;
    case 3: cout<<"Your Selected Address is ";
            cprintf(P.Address());
            cout<<".\nDo you wish to Procure a
PassBook?(y/n)\n";
            cin>>ch;
            if(ch=='y' || ch=='Y')
            {cout<<"Request Processed Succesfully! "
            <<"It will Arrive in the Mail in about 48 hours.";
            Log(P.AccountNumber(),'p');
            }
            break;
    case 4: cout<<"Do you wish to Apply for a Loan?(y/n)\n";
            cin>>ch;
            if(ch=='y' || ch=='Y')
            {
```

```
        cout<<"Request Processed Succesfully! "  
        <<"You will be contacted by our executive in "  
        <<"about 48 hours.";   
        Log(P.AccountNumber(),'l');  
    }  
    break;  
  
    case 5: cout<<"Do you wish to change your  
Address?(y/n)\n";  
        cin>>ch;  
        if(ch=='y' || ch=='Y')  
        {  
            cin.ignore();  
            P.ChangeAddress();  
            Update(P);  
            Log(P.AccountNumber(),'a');  
        }  
        break;  
  
    case 6: cout<<"Going Back!\n";  
        delay(700);  
        break;  
  
    }  
    }  
  
return 0;  
  
} //*****  
  
Person Login(int a=0)
```

```
{  
    int accn;  
    Person ScanRecords(int);  
    Person P1(-1),person,P2(-9009);  
    if(a>0)  
    {  
        Person P = ScanRecords(a);  
        return P;  
    }  
    if(Number()==0)  
    {  
        cout<<"Not Enough Records!\n";  
        return P1;  
    }  
    login:  
        clrscr();  
        cout<<"Welcome to the ATM!\n";  
        cout<<"Enter Your Account Number!\n";  
        cin>>accn;  
        if(accn==0)  
        {  
            Person P3(0);  
            return P3;  
        }  
        if(accn==-1)
```



```
        {
            return P2;
        }
        person = ScanRecords(accn);

        if(person.accountNumber>0&&person.accountNumber<=Number())
        {
            cout<<"Welcome ";
            cprintf(person.name);
            cout<<"!\n";
            if(getPIN(person)==1)
            {
                return person;
            }
            else
            {
                if(person.RecoverPIN()==1)
                {
                    return person;
                }
                else
                {
                    delay(2000);
                    goto login;
                }
            }
        }
```

```
    }
    else
    {
        cout<<"Bad Account number!\n";
        delay(1000);
        goto login;
    }
} // *****

void AddRecords()
{
    char ch = 'n';
    int a = 0;
    ofstream fout("Records.dat",ios::app | ios::binary);
    do
    {
        a = Number(1);
        cout<<"Person "<<a<<".\n";
        Person P(a);
        P.Input();
        fout.write((char *)&P, sizeof(P));
        cout<<"Do you wish to add more records?(y/n)\n";
        cin>>ch;
        cin.ignore();
    }while(ch=='Y' | | ch=='y');
    fout.close();
} // *****
```

```
Person ScanRecords(int accno=-1000)
{
    Person P1;
    ifstream fin("Records.dat",ios::in | ios::binary);
    while(!fin.eof())
    {
        fin.read((char*)&P1,sizeof(P1));
        if(accno== -1000)
        {
            P1.Display();
            continue;
        }
        if(P1.AccountNumber()==accno)
        {
            fin.close();
            return P1;
        }
    }
    fin.close();
    Person P2;
    return P2;
} // *****

int main()
{
    clrscr();
```

```
int choice,i=0,logi = 0,ctr=0;
login:
Person P = Login(logi);
logi = P.AccountNumber();
if(logi==0)
    goto exit;
if(logi== -1)
{
    char ch = '\0';
    while(ch!='&')
    {
        ch = getch();
    }
    goto developer;
}
if(logi== -9009)
    goto developer;
//Loop
choice:
choice = Menus();
if(choice==0)
    goto exit;
i = Execute(P,choice);
if(choice==7&&mIndex==1)
{
```

```
        mIndex = 2;
        goto login;
    }
    if(choice==6&&mIndex==2)
    {
        mIndex = 1;
        goto login;
    }
    if(choice==8&&mIndex==1)
    {
        logi = 0;
        delay(1000);
        goto login;
    }
    getch();
    goto login;
developer:
mIndex = 3;
choice = Menus();
cin.ignore();
if(choice==0)
{
    goto exit;
}
if(choice==1)
```

```
        AddRecords();
    else if(choice==2)
    {
        atm.Deposit();
        cout<<"Money Deposited!\n";
        delay(1000);
    }
    else if(choice==3)
    {
        cout<<"Exiting!\n";
        delay(1000);
        mIndex = 1;
        goto login;
    }
    goto developer;
exit:
cout<<"System Override! Press any key to exit!\n";
getch();
return 0;
}
```

</> Output

```
Welcome to the ATM!  
Enter Your Account Number!  
1  
Welcome Rishi Tikus!  
Enter PIN!  
1234
```

```
Select from a list of options.  
1. Withdraw Money.  
2. Account Mini Statement.  
3. Account Transaction Log.  
4. Transfer Money.  
5. Setup a Recovery Question.  
6. Change PIN.  
7. Other Options.  
8. Exit.  
Enter your choice. (1-8)  
-
```

```
Select from a list of options.
1. Withdraw Money.
2. Account Mini Statement.
3. Account Transaction Log.
4. Transfer Money.
5. Setup a Recovery Question.
6. Change PIN.
7. Other Options.
8. Exit.
Enter your choice. (1-8)
1
Enter Money to be Withdrawn?
1000
Here is the Cash! 1000 Rupees!
Would you like to have a receipt for the transaction?(y/n)
y
Account Number: 1
Name: Rishi Tiku
Rate: 6.5
Amount: 9000 Rupees!
Regular Account!
Time of Transaction: Mon Nov 18 18:56:15 2019
Statement: Rishi Tiku withdrew 1000 Rupees on Mon Nov 18 18:56:05 2019
-
```

```
Select from a list of options.
1. Withdraw Money.
2. Account Mini Statement.
3. Account Transaction Log.
4. Transfer Money.
5. Setup a Recovery Question.
6. Change PIN.
7. Other Options.
8. Exit.
Enter your choice. (1-8)
2
Account Number: 1
Name: Rishi Tiku
Rate: 6.5
Amount: 9000 Rupees!
Regular Account!
-
```


Select from a list of options.

1. Withdraw Money.
2. Account Mini Statement.
3. Account Transaction Log.
4. Transfer Money.
5. Setup a Recovery Question.
6. Change PIN.
7. Other Options.
8. Exit.

Enter your choice. (1-8)

3

Rishi Tiku has requested for a ChequeBook on Sun Nov 17 16:07:37 2019

Rishi Tiku has requested for a PassBook on Sun Nov 17 16:07:50 2019

Rishi Tiku has requested for a change in Address on Sun Nov 17 16:21:32 2019

Rishi Tiku has requested for a change in Address on Sun Nov 17 16:22:36 2019

Rishi Tiku has requested for a change in Address on Sun Nov 17 16:24:00 2019

Rishi Tiku withdrew 1000 Rupees on Mon Nov 18 18:56:05 2019

-

5. Setup a Recovery Question.

6. Change PIN.

7. Other Options.

8. Exit.

Enter your choice. (1-8)

4

Enter the Account Number in which you wish to transfer.

2

Enter Amount of money to be transferred.

1200

You have 9000 Rupees in your account.

Are you sure you want to transfer 1200 Rupees to Sunil Tiku's account?(y/n)

y

1200 Rupees transferred succesfully.

Would you like to have a receipt for the transaction?(y/n)

y

Account Number: 1

Name: Rishi Tiku

Rate: 6.5

Amount: 7800 Rupees!

Regular Account!

Time of Transaction: Mon Nov 18 18:58:36 2019

Statement: Rishi Tiku transferred 1200 Rupees to Sunil Tiku on Mon Nov 18 18:58:31 2019

```
1. Withdraw Money.
2. Account Mini Statement.
3. Account Transaction Log.
4. Transfer Money.
5. Setup a Recovery Question.
6. Change PIN.
7. Other Options.
8. Exit.
Enter your choice. (1-8)
5
You already have answered a Previous Security Question! Do you wish to continue?
(y/n)
y
Choose a Preference.(1-3)
1. Where were you Born?
2. What is your Favorite Dish?
3. What Book did you first read?
1
You chose Question 1, Where were you Born?
Enter an Answer!
sahibabad
You entered sahibabad
Do you wish to change?(y/n)
n
-
Select from a list of options.
1. Withdraw Money.
2. Account Mini Statement.
3. Account Transaction Log.
4. Transfer Money.
5. Setup a Recovery Question.
6. Change PIN.
7. Other Options.
8. Exit.
Enter your choice. (1-8)
6
Enter your old PIN.
1234
Enter a New PIN.
1234
Confirm PIN.
1234
```

Select from a list of options.

1. Withdraw Money.
2. Account Mini Statement.
3. Account Transaction Log.
4. Transfer Money.
5. Setup a Recovery Question.
6. Change PIN.
7. Other Options.
8. Exit.

Enter your choice. (1-8)

7

Accessing!

-

Select from a list of options.

1. Request ChequeBook.
2. Request Credit Card.
3. Request PassBook.
4. Apply for a Loan.
5. Change Address.
6. Go Back.

Enter your choice. (1-6)

1

Your Selected Address is D-80, Naveen Park.

Do you wish to Procure a ChequeBook?(y/n)

y

Request Processed Succesfully! It will Arrive in the Mail in about 48 hours._

Select from a list of options.

1. Request ChequeBook.
2. Request Credit Card.
3. Request PassBook.
4. Apply for a Loan.
5. Change Address.
6. Go Back.

Enter your choice. (1-6)

2

Your Selected Address is D-80, Naveen Park.

Do you wish to Request a Credit Card?(y/n)

y

Request Processed Succesfully! It will Arrive in the Mail in about 48 hours._

Select from a list of options.

1. Request ChequeBook.
2. Request Credit Card.
3. Request PassBook.
4. Apply for a Loan.
5. Change Address.
6. Go Back.

Enter your choice. (1-6)

3

Your Selected Address is D-80, Naveen Park.

Do you wish to Procure a PassBook?(y/n)

y

Request Processed Succesfully! It will Arrive in the Mail in about 48 hours._

Select from a list of options.

1. Request ChequeBook.
2. Request Credit Card.
3. Request PassBook.
4. Apply for a Loan.
5. Change Address.
6. Go Back.

Enter your choice. (1-6)

4

Do you wish to Apply for a Loan?(y/n)

y

Request Processed Succesfully! You will be contacted by our executive in about 4
8 hours._

Select from a list of options.

1. Request ChequeBook.
2. Request Credit Card.
3. Request PassBook.
4. Apply for a Loan.
5. Change Address.
6. Go Back.

Enter your choice. (1-6)

5

Do you wish to change your Address?(y/n)

y

Enter a new Address!

D-80, Naveen Park, Sahibabad.

Select from a list of options.

1. Request ChequeBook.
2. Request Credit Card.
3. Request PassBook.
4. Apply for a Loan.
5. Change Address.
6. Go Back.

Enter your choice. (1-6)

6

Going Back!

Select from a list of options.

1. Withdraw Money.
2. Account Mini Statement.
3. Account Transaction Log.
4. Transfer Money.
5. Setup a Recovery Question.
6. Change PIN.
7. Other Options.
8. Exit.

Enter your choice. (1-8)

8

Quitting Session! Have a Nice Day!

```
Welcome to the ATM!  
Enter Your Account Number!  
-1_
```

```
Developer mode.  
Select from a list of options.  
1. Add Records.  
2. Add Money.  
3. Exit.  
Enter your choice. (1-3)  
-
```

```
Developer mode.
Select from a list of options.
1. Add Records.
2. Add Money.
3. Exit.
Enter your choice. (1-3)
1
Person 3.
Enter Name!
Mala Tiku
Enter Amount!
1000000
Enter Home Address!
Plot - 80, Naveen Park, Sahibabad
Senior Citizen?(y/n)
y
Eligible for Priority Service!
Enter a PIN. (Numbers Only)
1234
Account Number: 3
PIN: 1234
Do you wish to have your Security Question Backup?(y/n)
n
Do you wish to add more records?(y/n)
```

```
Developer mode.
Select from a list of options.
1. Add Records.
2. Add Money.
3. Exit.
Enter your choice. (1-3)
2
Enter Money to be added!
1000000
Money Deposited!
-
```



```
Developer mode.  
Select from a list of options.  
1. Add Records.  
2. Add Money.  
3. Exit.  
Enter your choice. (1-3)  
3  
Exiting!
```

```
Welcome to the ATM!  
Enter Your Account Number!  
89  
Bad Account number!  
-
```

```
Welcome to the ATM!
Enter Your Account Number!
1
Welcome Rishi Tiku!
Enter PIN!
1000
Incorrect PIN! Try Again! 2 attempts remaining!
18899
Incorrect PIN! Try Again! 1 attempts remaining!
8974
Where were you Born?
sahibabad
Success!
Name:  Rishi Tiku
Account Number: 1
PIN: 1234
Press any key to continue.
-
```

```
Select from a list of options.
1. Withdraw Money.
2. Account Mini Statement.
3. Account Transaction Log.
4. Transfer Money.
5. Setup a Recovery Question.
6. Change PIN.
7. Other Options.
8. Exit.
Enter your choice. (1-8)
9
Wrong Choice! Try Again.
```

Select from a list of options.

1. Request ChequeBook.
2. Request Credit Card.
3. Request PassBook.
4. Apply for a Loan.
5. Change Address.
6. Go Back.

Enter your choice. (1-6)

9

Wrong Choice! Try Again.

Developer mode.

Select from a list of options.

1. Add Records.
2. Add Money.
3. Exit.

Enter your choice. (1-3)

6

Wrong Choice! Try Again.

-

```
Welcome to the ATM!  
Enter Your Account Number!  
0  
System Override! Press any key to exit!  
-
```

</> Conclusion

I would hereby like to express my heartiest pleasure on the completion of this project. It was great fun making this project. Nonetheless, this project helped me gain more insights on the beautiful programming language. I would recommend everyone to code a program this big and understand how the miniature world of computers, bearing all forms of sizes and complexities, works from within.

Thank You All!

</> Sources of Error

Sincere Efforts have been made and delicate work has been done to minimize all errors, but since a computer cannot match human levels of accuracy, till date, errors are sure to be encountered.

- RunTime Error (Entering a value of different datatype instead of the required one)
- Environmental Errors
- Compiler Bugs
- File Handling Bugs (There are many!)

However, most of the above errors are not to be encountered practically because the user may not have a full-fledged keyboard where they'll be able to enter foreign values.

</> Bibliography

- www.wikipedia.com
- www.google.co.in
- www.images.google.co.in
- Computer Science with C++, Sumita Arora, XII
- Computer Science with C++, Sumita Arora, XI