

ATM CASE STUDY

ATM CASE STUDY

- What is ATM?
- Why we choose ATM CASE STUDY?
- Language?
- Is it really a OOP Concept?
- Concepts (OOP terms)?
- Concerning Strategies/Modules?
- Our Objective with UML Design?
- Unit Testing
- Data Testing

ATM CASE STUDY(Cntd..)

- Editing/Designing Goal->well structured?
- Support linking Accounts?
- Design elements?

What is ATM?

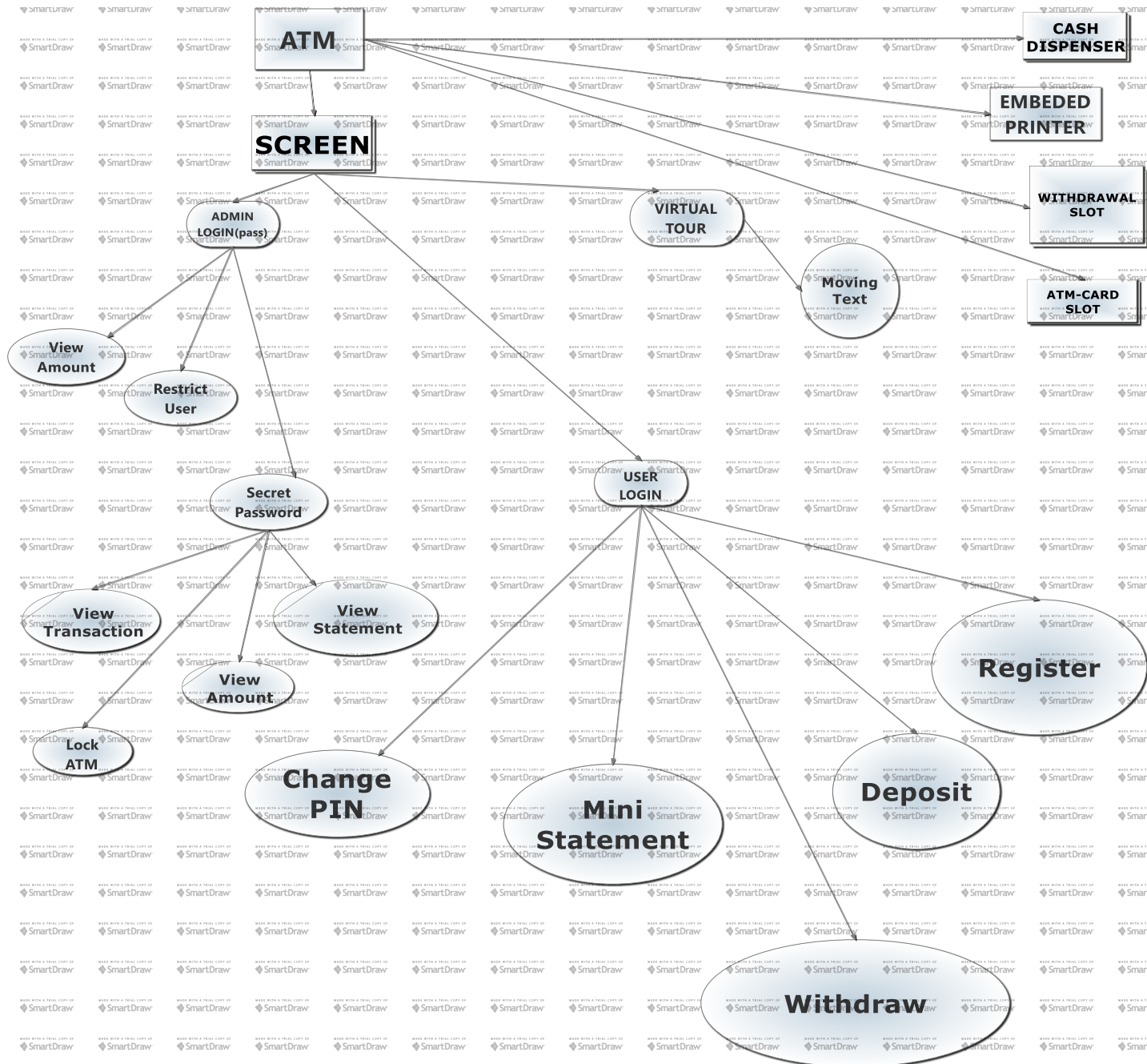
- Automated Teller Machine
- Programable H/W Memories
- We All know about it

Why We Choose ATM

- Complete H/W and S/W study
- Everyone is familiar
- A running Setup
- For Future Prospectus
 - For the time being if anyone of may have to work in Banks
 - So, It is useful to study ATM

Our Requirements in UML

Design of ATM



C++

- Complete Modular Programming Capability
- 33 Standard Header Files
- 18 Borrowed from 'C'
- About 51 header files

Language

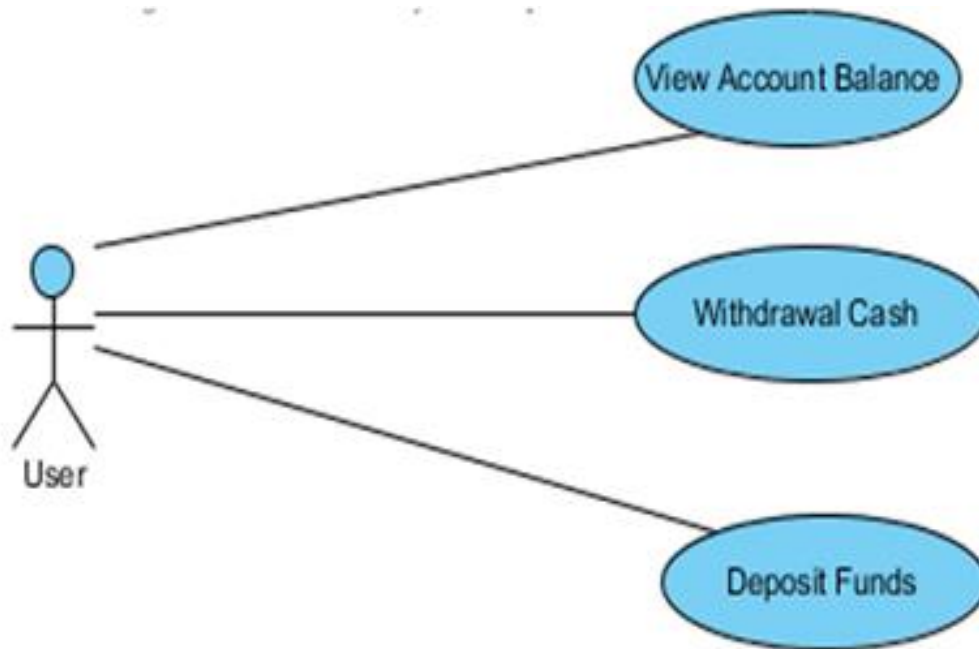
- Complete C++ is used
- Libraries:
 - `#include<iostream.h>`
 - `#include<stdio.h>`
 - `#include<conio.h>`
 - `#include<string.h>`
 - `#include<fstream.h>`
 - `#include<iomanip.h>`
 - `#include<math.h>`

Windows libraries

- `#include<Dos.h>`
- `#include<time.h>`
- `#include<stdlib.h>`

Is ATCS really OOP?

- ATCS is a complete oop concept
- We need to invoke little modules and routines again and again
- Protect confidential data elements -> oop



OOP Terms

- Classes
- Objects
- Private, public, protected access specifiers
- Static Data members
- Global Variables
- Constructors
- Member Functions Outside Classes
- Inheritance
- Containership
- Multilevel Inheritance
- Reusability/User Defined Header file

Inner Structures

Loop Structure

- While,
- do while,
- for

Selection Structure

- if else,
- switch,

Independent statements

- goto statement,
- kbhit() function
- Delay(); function
- getch();
- labels

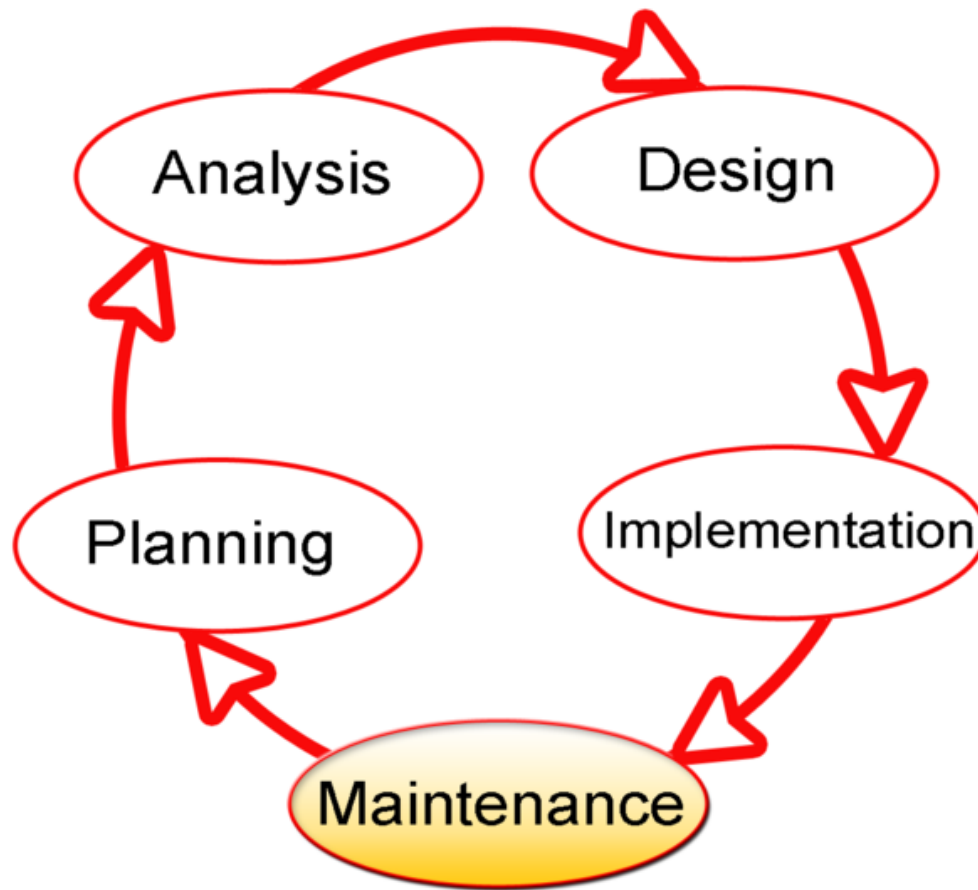
Concerning Strategies & Modules

- Explained In UML design
 - Embeded Printer
 - Withdrawal Slot
 - Cash Dispenser
 - Card Slot

Unit Testing & Data Testing

- Tested On
 - Intel(R) Pentium(R) CPU B950 @ 2.10 Ghz 2.10 Ghz
 - Further Testing and implementation ahead
 - Data testing
 - Teted Properly on given data types and parameters

Testing Phase



Phase Implementation (Compiler)

- Every Class is tested and compiled properly with certain testing parameters
- Then we embed classes to enhance our module

Accounts

- 4 billion Accounts maintained approximately, but requires large amount of functional Memory.

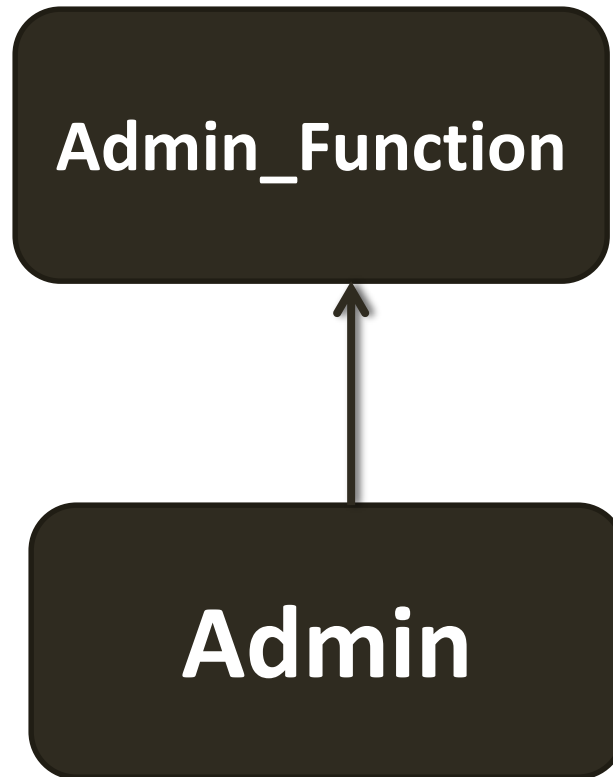
Design Elements

- Classes
 - Functions
 - Parameters

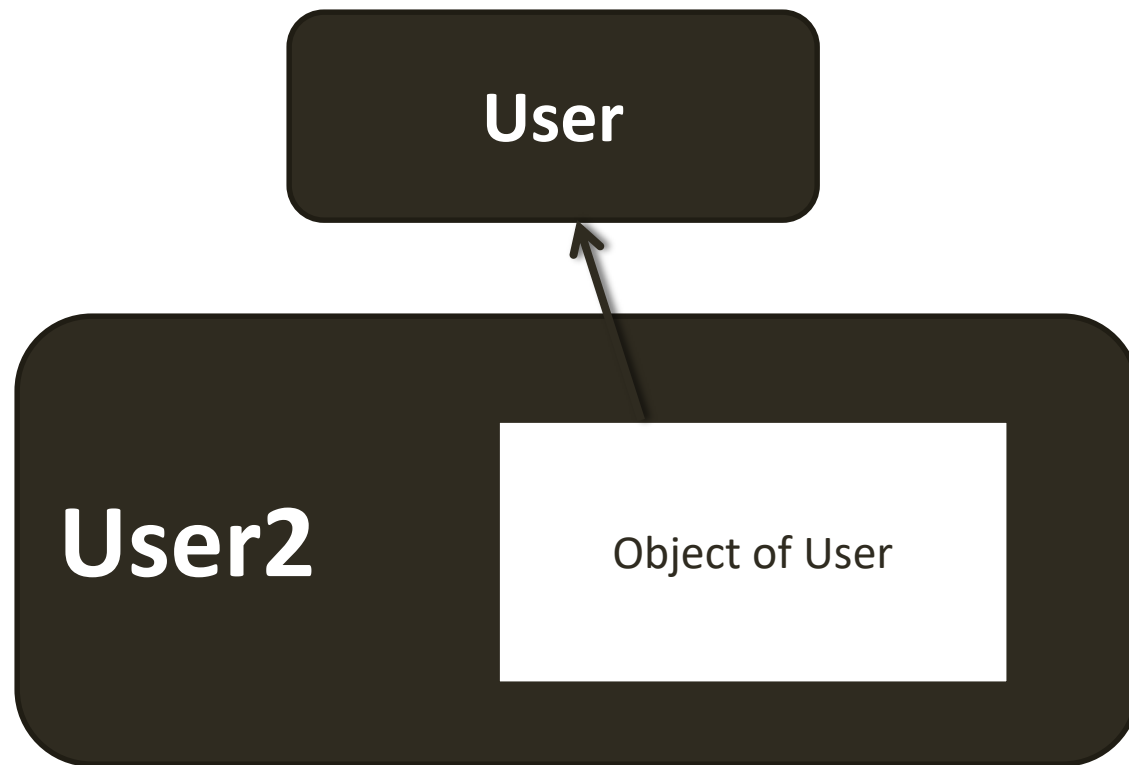
Classes

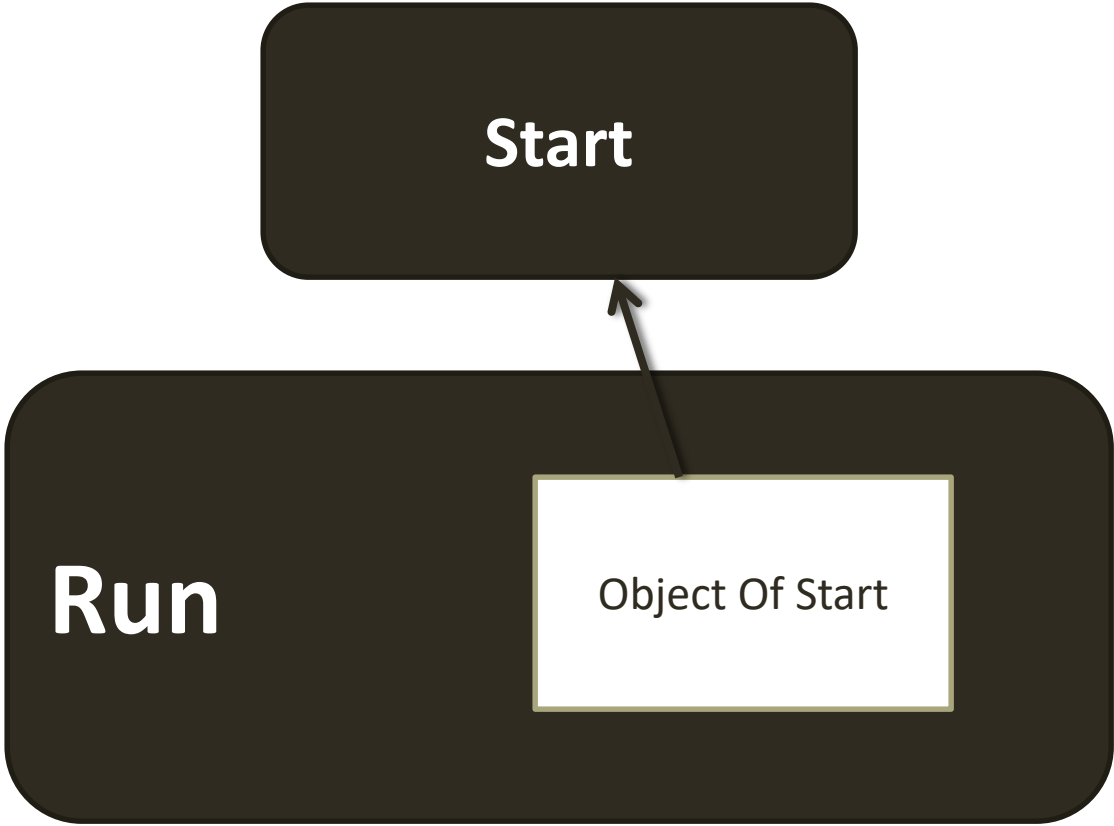
- class admin_function
- class admin:public admin_function
- class user
- class user2
- class start
- class run
- Class Thanks

Classes UML



Containership





Independent Class with thanks() constructor

Thanks

Program Modules

Built in Header Files

- `#include<iostream.h>`
- `#include<conio.h>`
- `#include<string.h>`
- `#include<fstream.h>`
- `#include<stdio.h>`
- `#include<iomanip.h>`
- `#include<dos.h>`
- `#include<time.h>`
- `#include<stdlib.h>`
- `#include<math.h>`

Main0

```
#include "atm.h"  
void main(void)  
{  
  run atm;  
}
```

Global Variables

```
long double deposited_amount=0;  
long double available_amount=500;  
long double transacted_amount=0;
```

Classes

```
class admin_function  
class admin:public admin_function  
class user  
class user2  
class start  
class run  
Class Thanks
```

Thanks

- class thanks
- {
- public:
- thanks()
- {
- for(int q=0;q<5;q++)
- {
- clrscr();
- delay(500);
- cout<<"\n\n\n\n\n\n\n\n\n";
- cout<<"\t\t\t*****";
- cout<<"\n\n\t\t\t Thanks For Using CSE-ATM";
- cout<<"\n\n\t\t\t*****";
- delay(500);
- }
- }
- };

Admin_function

```
• class admin_function
• {
• char pass2[20],pass3[20],r;
• public:
• void close_services()
• {
• cout<<"Enter Secret Password To Lock ATM:";
• cin>>pass2;
• char pass1[20]="atmcse";
• {
• if(strcmp(pass1,pass2)==0)
• {
• while(available_amount!=0)
• {
• stuck:
• while(!kbhit())
• {
• cout<<"\n\n\n\n\n\n";
• cout<<"\t\t\t-----";
• cout<<"\n\t\t\tSecret Password Enabled";
• cout<<"\n\n\t\t\tATM is Temporarily Down\n\n";
• cout<<"\t\t\t-----";
• delay(700);clrscr();delay(700);}
• cin>>pass3;
• if(strcmp(pass3,pass1)==0)
• {
```

- clrscr();
- while(!kbhit())
- {cout<<"\n\n\n\n\n\n\n\n\t\t\tSecret Password Disabled \\"ATM Is UP!\\"";
- delay(700);clrscr();delay(700);}
- getch();
- break;
- }
- else{goto stuck;}
- }
- }
- }
- }
- }
- void deposit_amount()
- {
- long double a;
- clrscr();
- cout<<"Least amount all time Availabe is : "<<500;
- cout<<"\n\nAmount Avsilable For Withdraw : "<<deposited_amount;
- cout<<"\n\nEnter Amount To Deposit:";
- cin>>a;
- deposited_amount+=a;
- cout<<"\n\nNew Amount Is="<<deposited_amount;
- cout<<"\nTotal amount in ATM="<<deposited_amount+500;
- getch();
- }

- void statement()
- {
- available_amount=deposited_amount-transacted_amount;
- clrscr();
- cout<<setw(50)<<"\nADMIN -> Statement\n";
- cout<<"\n\n\n\n";
- cout<<"\n\n\t\t\t*****";
- cout<<"\n\n\t\t\t Deposited Amount= "<<deposited_amount;
- cout<<"\n\n\t\t\t Transacted Amount= "<<transacted_amount;
- cout<<"\n\n\t\t\t Available Amount For Withdraw= "<<available_amount;
- cout<<"\n\n\t\t\t Total Amount in ATM= "<<available_amount+500;
- cout<<"\n\n\t\t\t*****\n";
- getch();
- }
- };

Admin:admin_function

- class admin:public admin_function
- {
- private:
- int choice;
- public:
- admin()
- {
- do
- {
- adm:
- clrscr();
- choice=0;
- cout<<setw(50)<<"\nWelcome To ATM Service-> Admin Block\n";
- cout<<"\n\n\n\n\n\n";
- cout<<"\n\t\t\t*****";
- cout<<"\n\t\t\t*-----Admin Block-----*";
- cout<<"\n\t\t\t* 1. Close Services *";
- cout<<"\n\t\t\t* 2. Deposit Amount *";
- cout<<"\n\t\t\t* 3. Check Amount History *";
- cout<<"\n\t\t\t* 4. Exit *";
- cout<<"\n\t\t\t*****\n";
- cin>>choice;

- `if(choice>0&&choice<5)`
- `{`
- `admin_function go;`
- `if(choice==1)`
- `go.close_services();`
- `else if(choice==2)`
- `go.deposit_amount();`
- `else if(choice==3)`
- `go.statement();`
- `}`
- `else`
- `{cout<<"\n\n\t\t\tInvalid Choice!";getch();goto adm;}`
- `}while(choice>0&&choice<4);`
- `}`
- `};`

user

```
class user
{
private:
int ac_no,pin_no,card_no;
long double withdraw_amount,deposit_amount,balance;
char name[15],cnic[15],ph_no[15],ac_type[15];
static unsigned long int ac,pin,card,index;
public:
int menu()
{
int menu=0;
clrscr();
cout<<setw(50)<<"\nWelcome To ATM Service-> Menue\n";
cout<<"\n\n\n";
cout<<"\n\t\t\t*****";
cout<<"\n\t\t\t*-----MENU-----*";
cout<<"\n\t\t\t* 1. Register Here *";
cout<<"\n\t\t\t* 2. Check Balance/Mini-Statement *";
cout<<"\n\t\t\t* 3. Withdraw *";
cout<<"\n\t\t\t* 4. Deposit *";
cout<<"\n\t\t\t* 5. Change Pin *";
cout<<"\n\t\t\t* 6. Exit *";
cout<<"\n\t\t\t*-----*";
cout<<"\n\t\t\t*****\n";
cin>>menu;
return menu;
}
```

- `int index_ac()`
- `{`
- `index++;`
- `card=index;`
- `return index;`
- `}`
- `void register_here(void);`
- `void store(user);`
- `void deposit(void);`
- `void withdraw(void);`
- `void check_balance(void);`
- `void change_pin(void);`
- `};`
- `unsigned long int user::index=0;`
- `unsigned long int user::ac=12345;`
- `unsigned long int user::pin=0;`
- `unsigned long int user::card=0;`

- `void user::register_here()`
- `{`
- `ac++;`
- `if(ac%2==0)`
- `pin=(ac%10*3)+9;`
- `else`
- `pin=(ac*3)/10+19;`
- `clrscr();`
- `cout<<"\n\nEnter The Following Information : ";`
- `cout<<"\n\nEnter Full Name:";`
- `cin>>name;`
- `cout<<"\n\nEnter CNIC#:";`
- `cin>>cnic;`
- `cout<<"\n\nEnter Phon No:";`
- `cin>>ph_no;`
- `cout<<"\n\nAccount Type:";`
- `cin>>ac_type;`
- `ac_no=ac;pin_no=pin;card_no=card;`
- `balance=1000;`

- withdraw_amount=0;
- deposit_amount=0;
- clrscr();
- cout<<"\n\t\t\tInformation Accepted!";
- cout<<"\n";
- cout<<setw(50)<<"\nYour Bank Keys Are Given Below:\n";
- cout<<"\n\n\n";
- cout<<"\n\t\t\t*****",
- cout<<"\n\n\t\t\t*-----Sign Up Sheet-----*";
- cout<<"\n\t\t\tName: "<<name;
- cout<<"\n\t\t\tCNIC : "<<cnic;
- cout<<"\n\t\t\tPhon Number: "<<ph_no;
- cout<<"\n\t\t\tAccount Type: "<<ac_type;
- cout<<"\n\t\t\tAccount No: "<<ac_no;
- cout<<"\n\t\t\tCard No: "<<card_no;
- cout<<"\n\t\t\tPin No: "<<pin_no;
- cout<<"\n\n\t\t\t*****\n";
- getch();
- }

- void user::store(user dummy_receive)
- {
- user inner_dummy=dummy_receive;
- ofstream out("c:\\account.text",ios::binary);
- out.write((char*)&inner_dummy,sizeof(inner_dummy));
- out.close();
- }
- void user::deposit()
- {
- int p;
- cout<<"\n\nEnter PIN# : ";
- cin>>p;
- if(p==pin_no)
- {
- cout<<"\n\nYour Current Balance is : "<<balance;
- cout<<"\n\nEnter Amount You Want To deposit : ";
- cin>>deposit_amount;
- balance=balance+deposit_amount;
- cout<<"\n\nYour New Balance is : "<<balance;
- }
- getch();
- thanks cse;
- }

- void user::withdraw()
- {
- int p;
- cout<<"\n\nEnter PIN# : ";
- cin>>p;
- if(p==pin_no)
- {
- cout<<"\n\nYour Current Balance is : "<<balance;
- cout<<"\n\nEnter Amount You want to withdraw :";
- cin>>withdraw_amount;
- if(withdraw_amount>(balance-500) || deposited_amount-withdraw_amount<=500)
- cout<<"\n\nSorry Not Available!!!";
- else
- {
- transacted_amount+=withdraw_amount;
- balance=balance-withdraw_amount;
- deposited_amount-=withdraw_amount;
- cout<<"\n\nYour New Balance is : "<<balance;
- }
- }
- else
- cout<<"\n\nInvalid PIN# !!!";
- getch();
- clrscr();
- thanks cse;
- }

- void user::change_pin()
- {
- int p,counter=0;
- while(counter<=3)
- {
- cout<<"\n\nEnter Previous PIN# : ";
- cin>>p;
- if(p==pin_no)
- {
- cout<<"\n\nEnter 4 digit numeric Pin#:";
- cin>>p;
- pin_no=p;
- cout<<"\nPin# is changed Successfully:";
- cout<<"\n\n";
- cout<<"-----\n";
- cout<<" | "<<pin_no<<" |\n";
- cout<<"-----";
- getch();
- break;
- }
- else
- {
- counter++;
- cout<<"\n\nWrong Pin#!!";
- cout<<"\n\nYou Can't Change It!";
- getch();
- if(counter==3)
- {cout<<"\nYou Have tried 3 times\nKnow We are reporting to Admin\nIf you are authorized\n";
- cout<<"Try after 24 Hours:";getch();break;}
- }
- }
- thanks cse;
- }

- void user::check_balance()
- {
- int p;
- cout<<"\n\nEnter PIN# : ";
- cin>>p;
- if(p==pin_no)
- {
- clrscr();
- cout<<"\n\n\t\t\t*****",
- cout<<"\n\n\t\t\t Your Account No : "<<ac_no;
- cout<<"\n\n\t\t\t Your Account Type : "<<ac_type;
- cout<<"\n\n\t\t\t Your available Balance is : "<<balance;
- cout<<"\n\n\t\t\t*****\n";
- getch();
- }
- else{cout<<"\n\n\nInvalid PIN:";}
- thanks cse;
- }

- class user2
- {
- public:
- static int temp;
- void user2_call()
- {
- move:
- clrscr();
- int choice=0;
- int i=0;
- int c=0;
- user obj[100],screen;

- do{
- choice=screen.menu();
- if(choice>1&&choice<6)
- {cout<<"Card No:(Automated Scanner H/W):";
- cin>>c;
- if(c<=temp)
- {
- if(choice==2)
- obj[c].check_balance();
- else if(choice==3)
- obj[c].withdraw();
- else if(choice==4)
- obj[c].deposit();
- else if(choice==5)
- obj[c].change_pin();
- }else{cout<<"\n\nAccount Doesn't Exit!!";getch();}
- }

- else
- {
- if(choice==1)
- {user dummy; //storing in file
- i=screen.index_ac();
- temp=i;
- obj[i].register_here();
- dummy=obj[i];
- obj[i].store(dummy);}
- }
- if(choice<1&&choice>6)
- {cout<<"\n\Invalid Choice:";getche();}
- }while(choice>0&&choice<6);
- }
- };
- int user2::temp=0;

```

• class start
• {
• public:
• int top_menu;
• start()
• {
• top_menu=0;
• clrscr();
• while(!kbhit())
• {
• cout<<"\t\tProcedural versus object oriented programming languages,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tObject oriented design strategy and problem solving,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tObjects & classes, member functions,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tPublic and private members,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tDynamic memory management,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tConstructors and destructors,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tTemplates,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tObject encapsulation,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tDerived classes,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tClass hierarchies,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tInheritance and polymorphism,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tOperator overloading,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tStream class,\n\n";
• delay(500);cout<<"ATM STUDY";
• cout<<"\t\tPractical design through OOP\n\n";
• delay(500);
• cout<<"\nLAB: Laboratory work will be based on the contents of the course.\n";
• delay(500);
• cout<<"\nSuggested Text:\nObject Oriented Programming, C++ by Robert Lafore\n";
• delay(500);
• }
• }

```

- int atm_menu()
- {
- char admin_pass1[20];
- char admin_pass2[20]="admincse";
- do
- {
- clrscr();
- cout<<"Welcome To ATM Service:";
- cout<<"\n\n\n\n\n\n\n\n";
- cout<<"\n\t\t\t\t*****";
- cout<<"\n\t\t\t\t*-----ATM Service-----*";
- cout<<"\n\t\t\t\t* 1. User Login *";
- cout<<"\n\t\t\t\t* 2. Admin Login *";
- cout<<"\n\t\t\t\t* 3. Virtual Tour *";
- cout<<"\n\t\t\t\t* 4. Terminate ATM Program *";
-
- cout<<"\n\t\t\t\t*****\n";
- cin>>top_menu;

- if(top_menu==1)
- {
- user2 obj;
- obj.user2_call();
- }
- if(top_menu==2)
- {
- cout<<"Administrator Password:";
- cin>>admin_pass1;
- if(strcmp(admin_pass1,admin_pass2)==0)
- {admin obj;}
- else
- {cout<<"Invalid";getch();}
- }
- if(top_menu==3)
- {start constructor;}
- if(top_menu>4&&top_menu<0)
- {cout<<"\n\nInvalid:";getch();}
- }while(top_menu>0&&top_menu<4);
- return top_menu;
- }
- };

- class run
- {
- public:
- run()
- {
- textcolor(114);
- again:
- int repeat=0;
- start object;
- repeat=object.atm_menu();
- if(repeat!=4)
- {
- repeat=0;
- goto again;
- }
- }
- };

Questions