**UML**

etails(int) : void

# password : int

# acc\_id : string

# name : string

# phone : string

# balance : float

# userid : string

# \*a : ATM

+ write() : void

int

+ getid() : void

+ deduct(char c[],float ded) : void

+ credit(char c[], float add) : void

+ details() : void

**SAVINGS**

**ACCOUNT**

+message[]:string

**MyException**

+ check\_details(int a\_num, int pwd)) : int

**ATM\_TRANSACTION**

+ getid() : void

+ deduct(char c[],float ded) : void

+ credit(char c[], float add) : void

+ details(string) : void

**CURRENT**

+assigns(Savings &ss):void

+assignc(Current &cc):void

+ assigna(ATM\_transaction &aa):void

+ clerk\_screen():void

+ cust\_screen():void

# save: Savings

# current: Current

# car: CAR

# edu: EDUCATION

# atm: ATM\_transaction

# dd: Locker

# lock: Locker

**LO GIN**

+getatm():int

+getatmid():char

+getpass() :int

# atm\_num : int

# accd :char

# atm\_password : int

#

**ATM**

+ get\_id() : virtual string

+ deduct(char[] , float) : virtual void

+ credit(char[] , float) : virtual void

+ atm\_exist(): virtual int

+ details(int) : virtual void

+ update() :void

+ details() : void

+ create() : void

+ check\_salary() : int

+ getlid():string

**LOCKER**

# salary : float

**CAR**

# loan\_id : string

# fixed\_amt: float

# rate\_interest : float

# num\_of Installments : int

# installments\_paid : int

# principle : float

#amount\_paid : float

# date : string

**LOAN**

+ print() : void

+ create (): void

# date : string

# from : string

# to : string

+ update() :void

+ details() : void

+ check\_marks() : int

+ getlid() :string

# twelth\_marks : float

**EDUCATION**

+create():virtual void

+details():virtual void

+check(char []):virtual int

+check\_marks(float): virtual int

+check\_salary(float): virtual int

+update():virtual void

**DD**

+ create() : void

+ details() : void

+ update() : void

+ getlocknum():int

# locker\_num : int

# type : string

# rent : float

# date : string