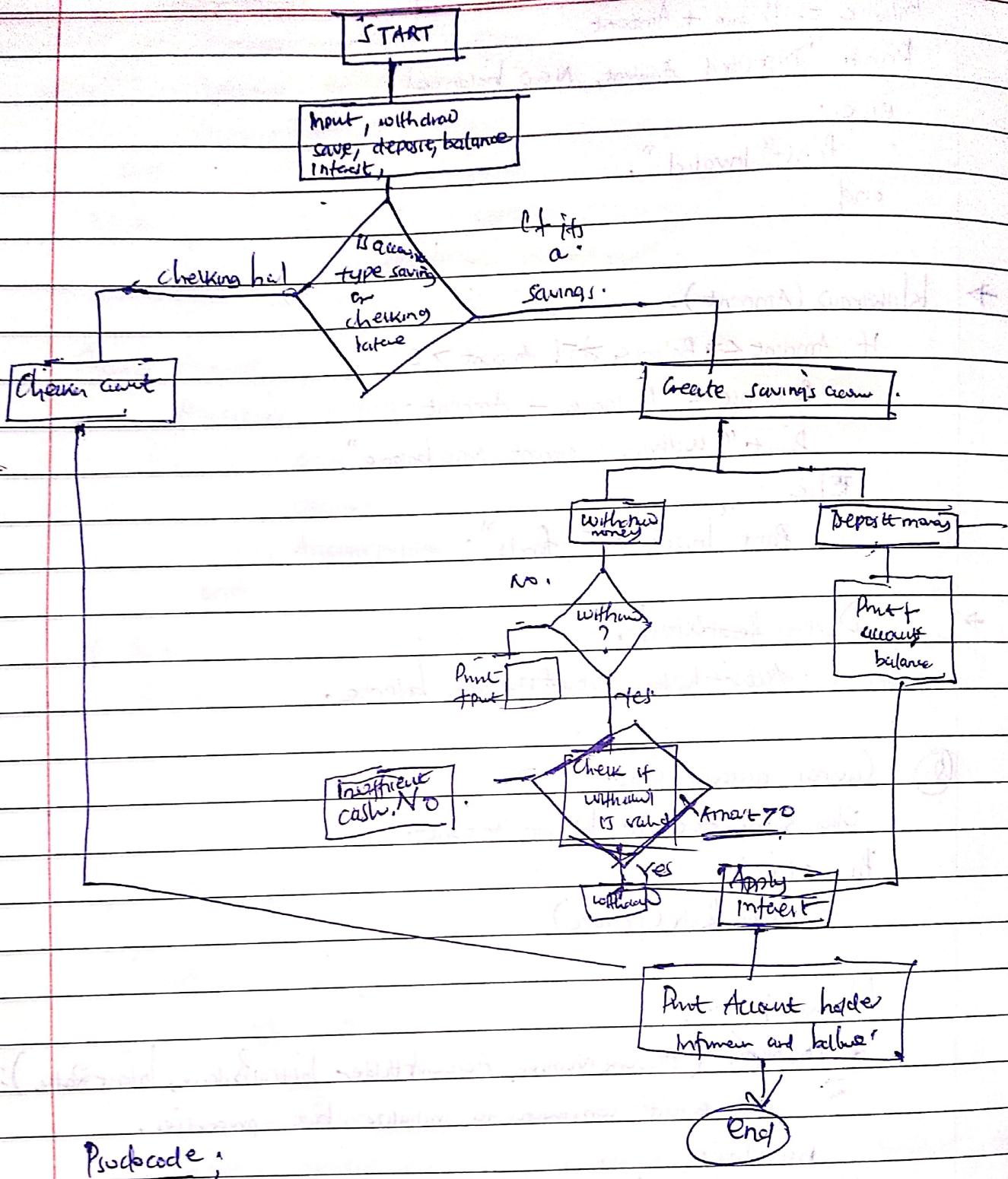


# Banking systems

PAGE NO. 1  
DATE: / /



Pseudocode:

base class: Account

Class Account

Properties

- AccountNumber (protected)

- Balance (protected)

- Account Holder (protected)

Methods:

- Constructor (Account Number, Account Holder, Initial Balance)

- void. account number, Account holder, balance

Code

## Classdef Account

Properties:

Account Number

Balance

Account Holder

end.

Methods

function obj = Account

## Classdef Account

Properties (Access = protected).

Account Number

Balance

Account Holder

end.

Methods.

function obj = Account (accountNumber, accountHolder, initialBalance)

obj.AccountNumber = accountNumber

obj.AccountHolder = accountHolder

obj.Balance = initialBalance

end

% method to deposit money.

function obj = deposit (obj, amount)

If amount > 0:

obj.Balance = obj.Balance + Am amount

fprintf (' Deposited Shs. %.2f, New Balance : Shs %.2f/n ', amount, obj.Bal

else

error

end

end.

% Method to withdraw money.

function obj = withdraw (obj, amount)

If amount <= obj.Balance && amount > 0

obj.Balance = obj.Balance - Amount

fprintf (' Withdraw Shs %.2f, New Balance Shs %.2f/n ', amount, obj.Bal

else

error (' Insufficient funds')

end.

function displayDetails ( Obj ).

```
printf ('Account Holder : %s \n', AccountHolder, obj.AccountNumber);
Balance shs %f \n', obj.AccountHolder, obj.AccountNumber,
obj.Balance );
end.
```

① Protected getter of Balance.

function balance = getBalance ( Obj ).

```
balance = obj.Balance;
```

end

end

end.

② Derived Class of ChekingAccount and SavingsAccount.

ClassDef SavingsAccount < Account.

Properties ( Access = private )

InterestRate.

Methods.

function Obj = SavingsAccount ( accountNumber, accountHolder, initialBalance ).

Obj @ Account ( \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_ ).

Obj. InterestRate = InterestRate;

end.

③ methods to apply interest rate.

function Objet = applyInterest ( Obj ).

Interest = Obj.Balance \* Obj. InterestRate / 100;

Obj. Balance = Obj. Balance + Interest;

printf ('Applied \$ shs. of Interest, New balance ; \$ / m,
Interest, Obj. Balance );

end

end

end.

? methods for overdraft

Main program

5. Create a saving account.

Savings = savings Account ('SA123', 'Ishant', 1000, 5);

6. Display Initial Details,

Savings, displayDetails();

7. Perform Operations

Savings = savings, deposit(500); // Deposit money.

Savings = savings, withdraw(300);

Savings = savings, applyInterest(); . . .

8. Display final Details,

Savings, displayDetails();