

```

class account {
    } true to classes & know well
    String customerName; ; starts with "shubh"
    String accountNumber;
    String accountType; -> saving, current
    double balance;

```

2) Account(String customerName, String accountNumber,

String accountType, double balance){

Customer("Shubh"), this.customerName = customerName;

this.accountNumber = accountNumber;

this.accountType = accountType;

this.balance = balance;

}

} (referring to last block) block

3) void deposit(double amount) {

String customerName = "Shubh";

balance = balance + amount; true, matches

System.out.println("Successful Deposit. updated balance
: " + balance);

}

} (true to shubh) } (working fine

} (success - if match) if

void displayBalance() {

System.out.println("balance " + balance); matches

balance += amount; matches

System.out.println("successful deposit. updated balance
: " + balance);

}

class SavAcct & extends account {
double interestRate; // interestRate
// calculate
this.interestRate = interestRate; // interestRate
// calculate

Act 80

```
address SavAcct( string <customerName>, string <accountNumber>,
double <balance>, double <interestRate>) p{  
    emiSuper(<customerName>,<accountNumber>, "savings", <balance>);  
    emiThis.interestRate = <interestRate>;  
}  
  
q3dUT2400-1P = q3dUT2400-21P  
    ( used = used . 21st )
```

void ComputeAndDepositInterest();

double interest balance *interestRate / 100

balance + = interest + principle

system.out.println("Interest deposited. updated balance

fixed + the balance); restrict the number

: (wind + ":

void withdraw(~~t~~f (double amount) {

if (balance > amount) {

balance = amount right - debit b/w

system.out.println("successful update, updated
balance "+ctr.balance);

Concluded + "

2

Current

class Savings extends account {
 double minimumBalance; double serviceCharge;
 double savingCharge; }
 {
 double minBal = 1000.00;
 double serviceCharge = 10.00;
}

```
if (0.0, "Savings", "s101") balance = 2000.00  

  Savings (String customerName, String accountNumber,  

  double minimumBalance, double serviceCharge) {  

  super (customerName, accountNumber, "Current", balance);  

  this.minimumBalance = minimumBalance;  

  this.serviceCharge = serviceCharge;  

}  

  , (0.0, "Customer", "s101")  

  ; (0.0, "Customer", "s101")  

  ; (0.0, "Customer", "s101")
```

```
void withdraw (double amount) {  

  if (balance >= amount) {  

    balance -= amount;  

    System.out.println ("Insufficient balance.");
}
```

```
}  

  }
```

: withdraw
 ? withdraw

```
: (" --- withdraw --- ") + "Current" + "s101"  

  void checkMinimumBalance () {  

  if (balance < minimumBalance) {  

    balance += serviceCharge;  

    System.out.println ("Balance below minimum  

  (balance - serviceCharge) imposed. Updated balance  

  : " + balance);
}
```

```
} : (" --- withdraw --- ") + "Current" + "s101"  

  (" withdraw after ") + "Current" + "s101"
```

}

```
;" "Created by myself") catch. Examples  
int choice;  
do {  
    System.out.println("In--- Bank menu ---");  
    System.out.println("1. Deposit");  
    System.out.println("2. Withdraw");  
    System.out.println("3. Display balance");  
    if (account instanceof SavingsAccount) {  
        System.out.println("4. Compute and deposit interest");  
    }  
    System.out.println("D. Exit");  
    System.out.println("Enter your choice");
```

choice = Scanner.nextInt();
 switch (choice) {

case 1 : (0 = 1 to 3) *deposit*

System.out.println("Enter amount to deposit:");
 double depositAmount = scanner.nextDouble();
 account.deposit(depositAmount);
 break;

case 2 :

System.out.println("Enter amount to withdraw:");
 double withdrawAmount = scanner.nextDouble();
 account.withdraw(withdrawAmount);

case 3 :

account.displayBalance(); *initial value*

break;

; (break in case 1, 2 and 3) *if it comes then*

case 4 :

if(amount instanceof SavAct){

((SavAct) account).computeInterest();

} else{

System.out.println("Interest computation is
 not applicable for current accounts.");

} *final fix for two student*

fix 1

break;

1: final may not

00001: final at end after

a school is started with initial fix

3
3

: () bank transaction = main
? (option) choice

while (choice != 0) :

: 1 3.20

: (" : type("Scanner closed") ; type(" ")) at least BNF. matches
? () bank transaction = bank transaction oldlib
; (bank transaction) through function
: standard

output :

: 6 3.20

: (" : withdraw or transfer entry") at least two type

: () withdraw customer name & amount withdrawal oldlib
Display transaction withdraw transaction. ENDTRAN

Enter account number:

1234567

: \$ 9000

Enter initial balance enter full float amount

400

: 4000

Enter account type (1 for savings , 2 for current) :

1

: 1 9000

? (F1AVD 0 savings function) if

: () initial balance menu selection (F1AVD)

1. Deposit

? select

: (withdraw withdraw entry) at least two type

: (withdraw 3) display balance withdraw transaction

4. Compute and Deposit interest

D. Exit

: deposit

Enter your choice :)

Enter amount to deposit : 100000

Deposit successful. New balance : 100400. D

--- Bank Menu ---

1. Deposite
2. withdraw
3. Display Balance
4. Compute and deposit Interest
0. Exit

Enter your choice : 2

Enter amount to withdraw : 5000

withdrawal successful . New balance : 95400.0

--- Bank Menu ---

1. Deposit
2. withdraw
3. Display Balance
4. Compute and deposit Interest
0. Exit

Enter your choice : 3

current balance : 95400.0

To
Rt
99.10

```
C:\Users\STUDENT\Downloads>java Bank
Enter customer name:
divya
Enter account number:
1234567
Enter initial balance:
400
Enter account type (1 for Savings, 2 for Current):
1
```

--- Bank Menu ---

- 1. Deposit
- 2. Withdraw
- 3. Display Balance
- 4. Compute and Deposit Interest
- 0. Exit

Enter your choice: 1

Enter amount to deposit: 100000

Deposit successful. New balance: 100400.0

--- Bank Menu ---

- 1. Deposit
- 2. Withdraw
- 3. Display Balance
- 4. Compute and Deposit Interest
- 0. Exit

Enter your choice: 2

Enter amount to withdraw: 5000

Withdrawal successful. New balance: 95400.0

--- Bank Menu ---

- 1. Deposit
- 2. Withdraw
- 3. Display Balance
- 4. Compute and Deposit Interest
- 0. Exit

Enter your choice: 3

Current balance: 95400.0

--- Bank Menu ---

- 1. Deposit
- 2. Withdraw
- 3. Display Balance
- 4. Compute and Deposit Interest
- 0. Exit

Enter your choice: 4