WEEK 5 :

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facili es but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

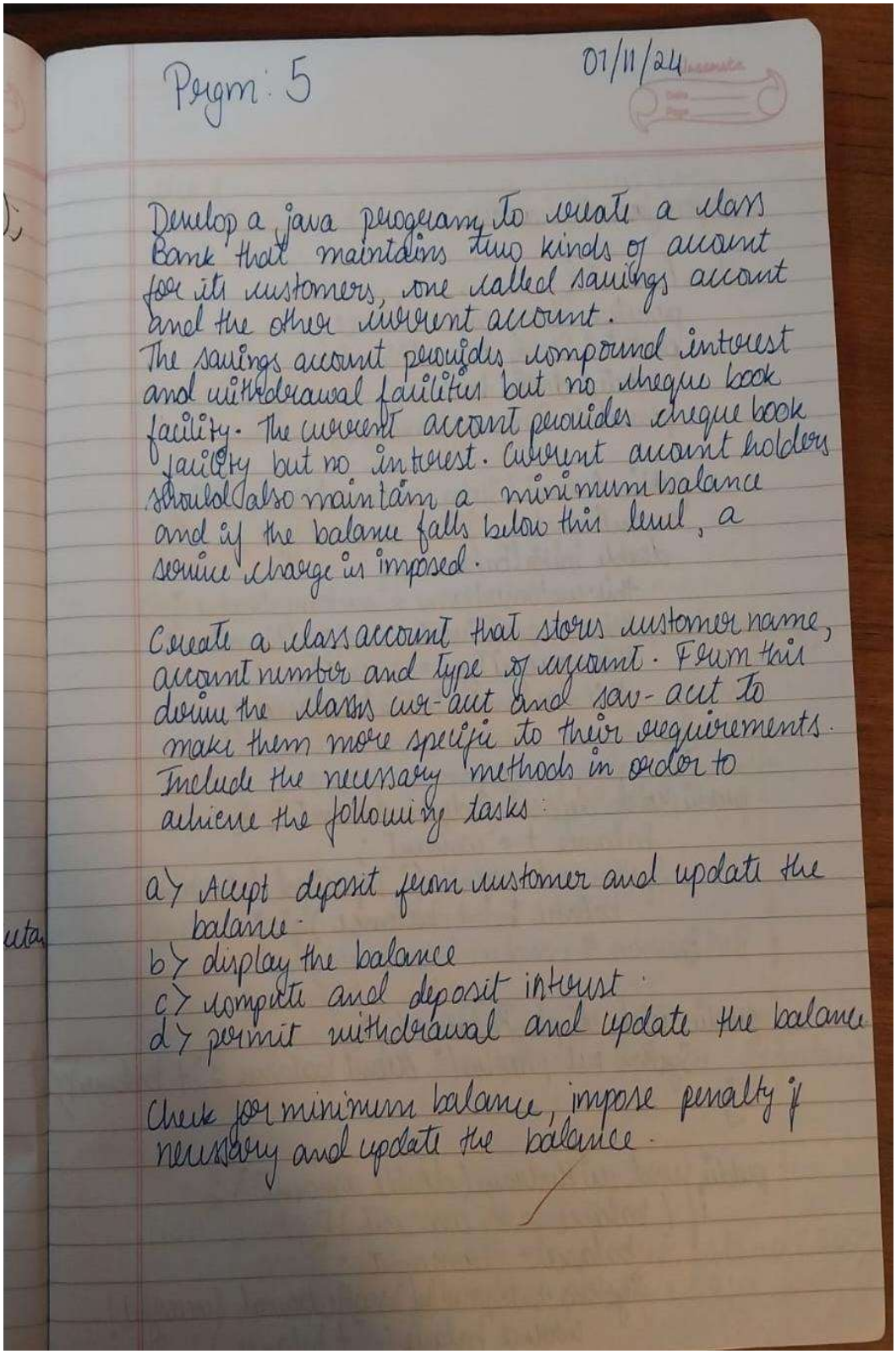
a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.



Pgm: 5

01/11/24

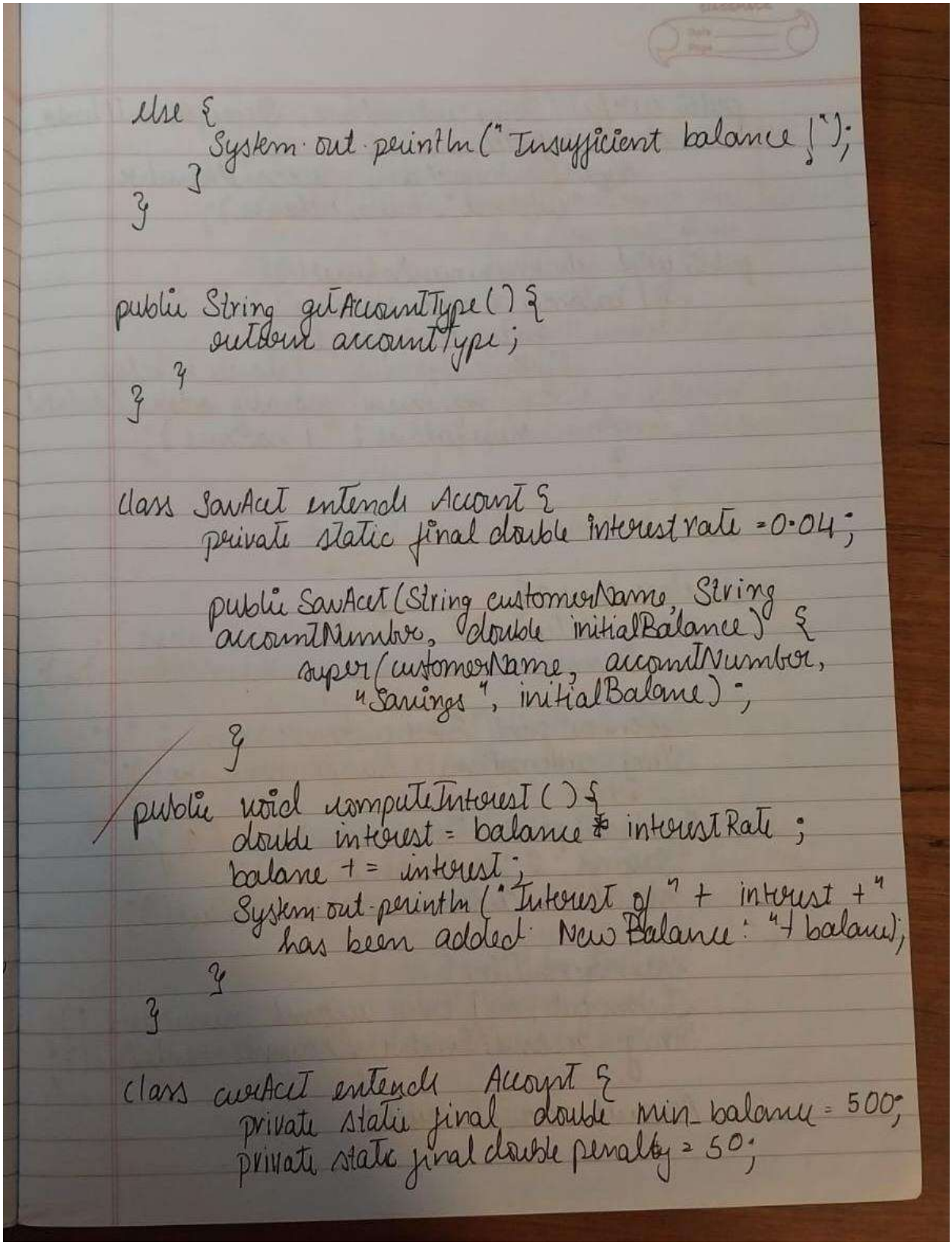
Develop a java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account.
The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class account that stores customer name, account number and type of account. From this derive the classes cur-acct and sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.
b) display the balance
c) compute and deposit interest
d) permit withdrawal and update the balance

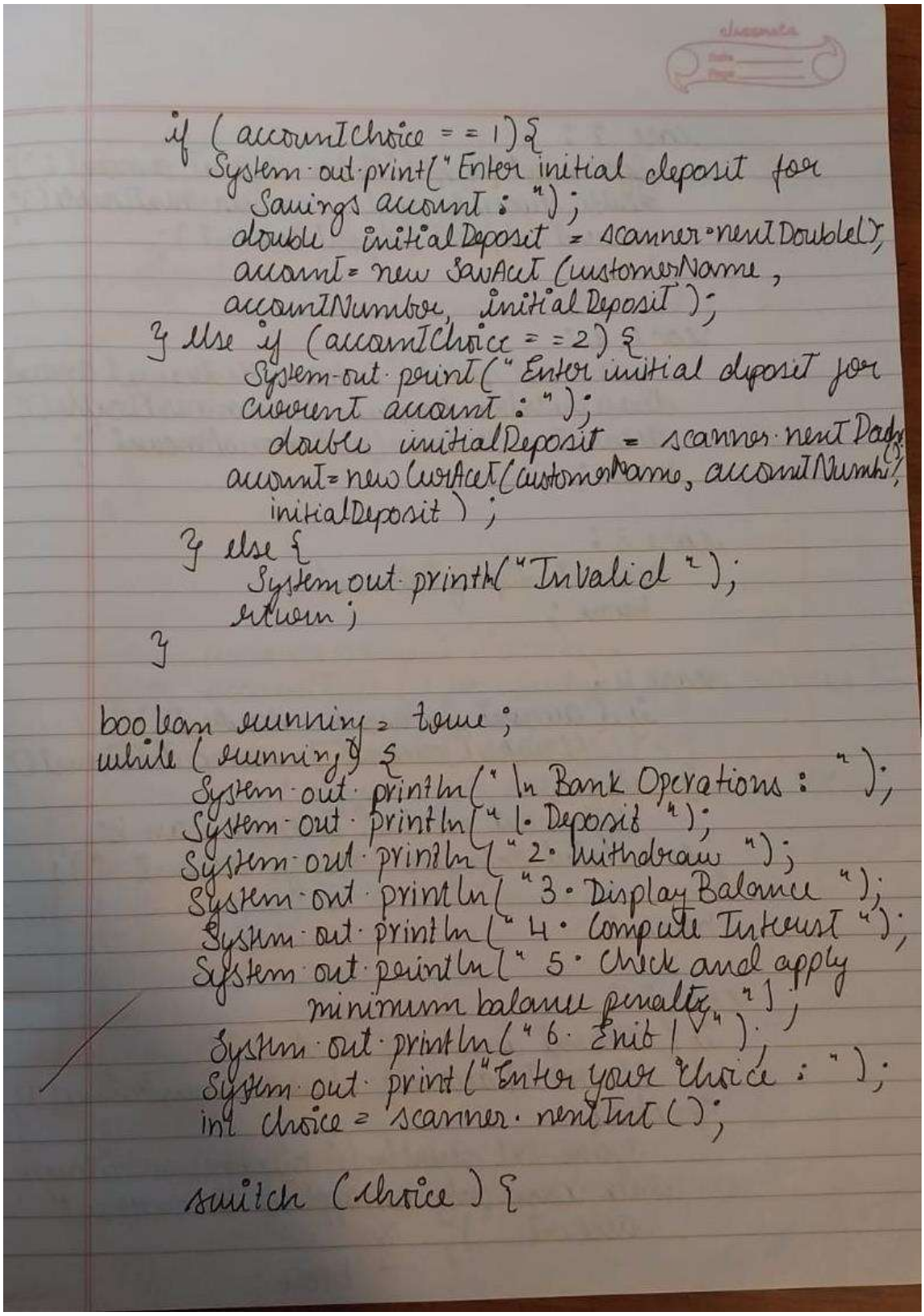
Check for minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;

class Account {
    protected String customerName;
    protected String accountNumber;
    protected double balance;
    protected String accountType;

    public Account(String customerName, String
    accountNumber, String accountType,
     double initialBalance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = initialBalance;
    }

public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited! Current
            balance: " + balance);
    }

public void displayBalance() {
    System.out.println("Account balance: " + balance
}

public void withdraw(double amount) {
    if (balance >= amount) {
        balance -= amount;
        System.out.println("withdrawal Successful!
            current balance: " + balance); }
```

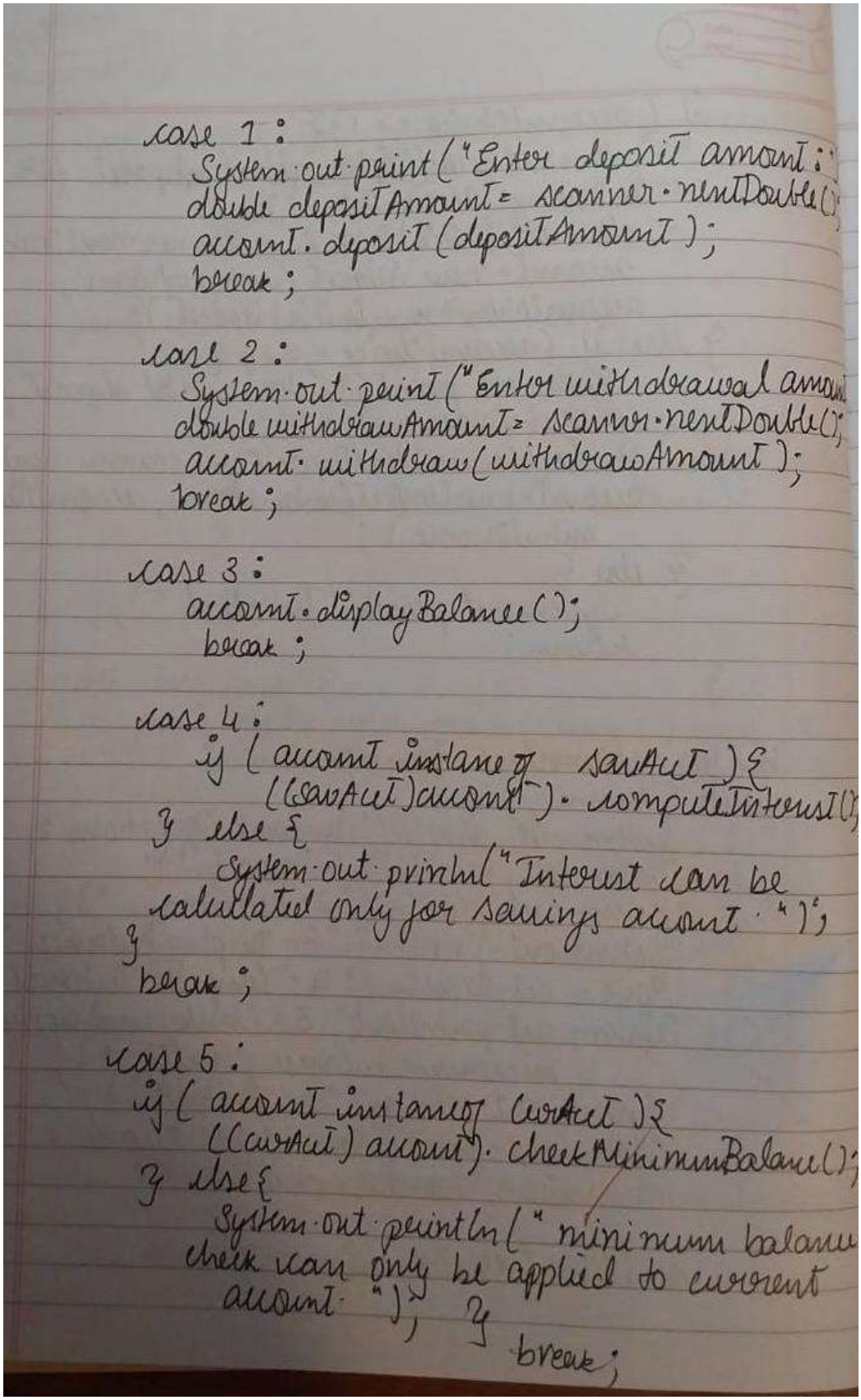


```
    else {
        System.out.println("Insufficient balance!");
    }
}

public String getAccountType() {
    return accountType;
}

class SavAct extends Account {
    private static final double interestRate = 0.04;

    public SavAct(String customerName, String
    accountNumber, double initialBalance) {
        super(customerName, accountNumber,
            "Savings", initialBalance);
    }

    public void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest of " + interest + "
            has been added. New Balance: " + balance);
    }
}

class curAct extends Account {
    private static final double min_balance = 500;
    private static final double penalty = 50;
```

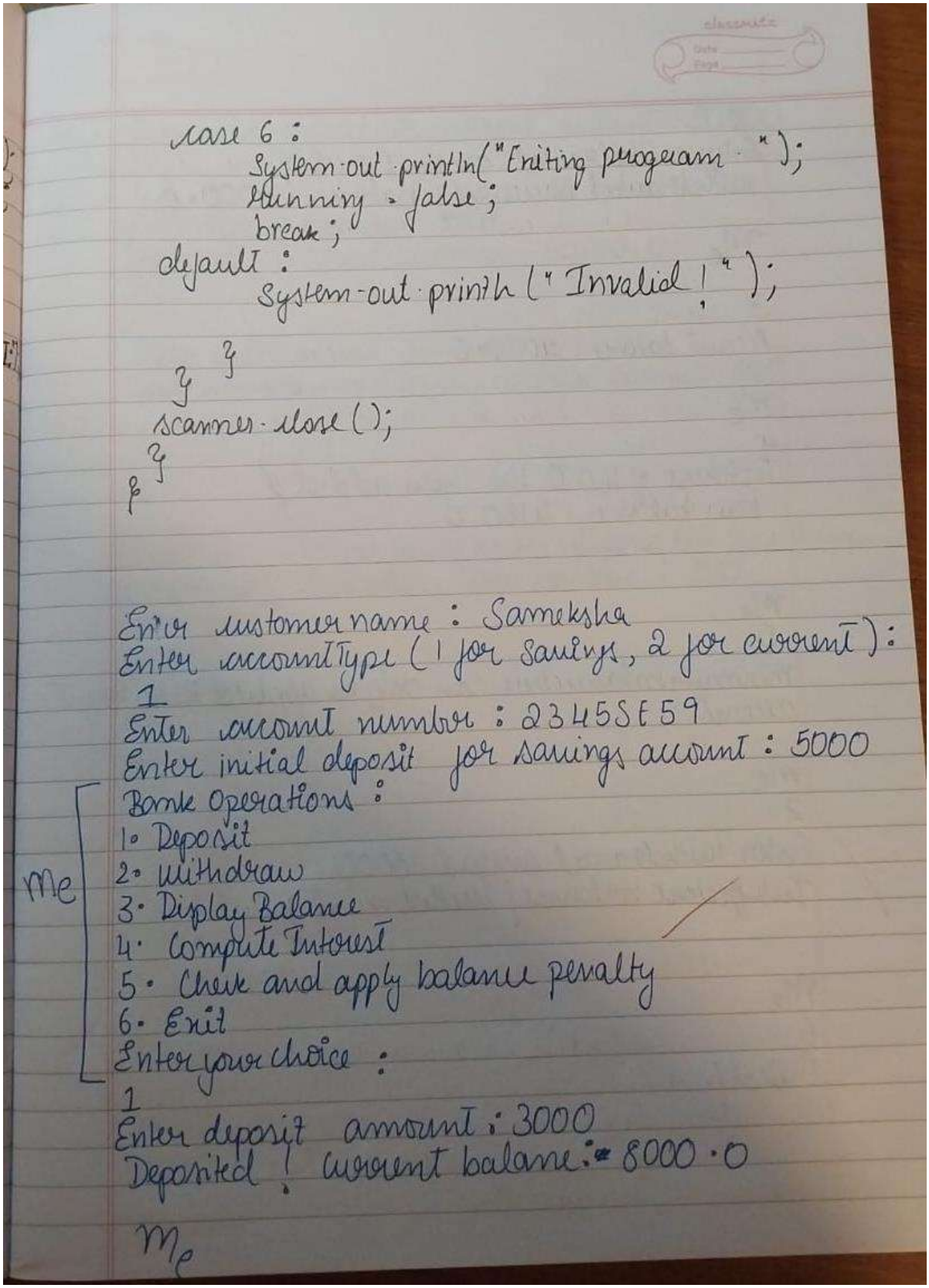
```
publi curAct (String customerNam, String accountNumb
    double initialBalance) {
        super(customerName, accountNumber,
        "current", initialBalance);
    }
publi void checkminimumBalance(){
    if (balance < min_balance) {
        balance -= penalty;
        System.out.println("Balance is below
            minimum. penalty added/deducte
            new balance: " + balance);
    }
}
}

publi class Bank1 {
    publi static void main(String[] args) {
        Scanner scanner=new scanner(System.in);

        System.out.print("Enter customer name: ");
        String customerName = scanner.nextLine();

        System.out.print("Enter account type (1 for
        Savings, 2 for Current): ");
        int accountChoice = scanner.nextInt();

        scanner.nextLine();
        System.out.print("Enter account number: ");
        String accountNumber = scanner.nextLine();

        Account account= null;
```



```
if (accountChoice == 1){
    System.out.print("Enter initial deposit for
        Savings account : ");
    double initialDeposit = scanner.nextDouble();
    account = new SavAct(customerName,
    accountNumber, initialDeposit);
} else if (accountChoice == 2) {
        System.out.print("Enter initial deposit for
        current account : ");
        double initialDeposit = scanner.nextDou
    account = new CurrAct(customerName, accountNumb
        initialDeposit);
} else {
        System.out.println("Invalid ");
        return;
}

boolean running = true;
while (running) {
        System.out.println("\n Bank Operations : ");
        System.out.println("1. Deposit ");
        System.out.println("2. Withdraw ");
        System.out.println("3. Display Balance ");
        System.out.println("4. Compute Interest ");
        System.out.println("5. Check and apply
            minimum balance penalty ");
        System.out.println("6. Exit \n");
        System.out.print("Enter your choice : ");
        int choice = scanner.nextInt();

        switch (choice) {
```



```
case 1:
    System.out.print("Enter deposit amount:");
    double depositAmount = scanner.nextDouble();
    account.deposit(depositAmount);
    break;

case 2:
    System.out.print("Enter withdrawal amount:");
    double withdrawAmount = scanner.nextDouble();
    account.withdraw(withdrawAmount);
    break;

case 3:
    account.displayBalance();
    break;

case 4:
    if (account instanceof SavAct) {
        ((SavAct)account).computeInterest();
    } else {
        System.out.println("Interest can be
calculated only for savings account.");
    }
    break;

case 5:
    if (account instanceof CurAct) {
        ((CurAct)account).checkMinimumBalance();
    } else {
        System.out.println("minimum balance
check can only be applied to current
account."); }
                    break;
```

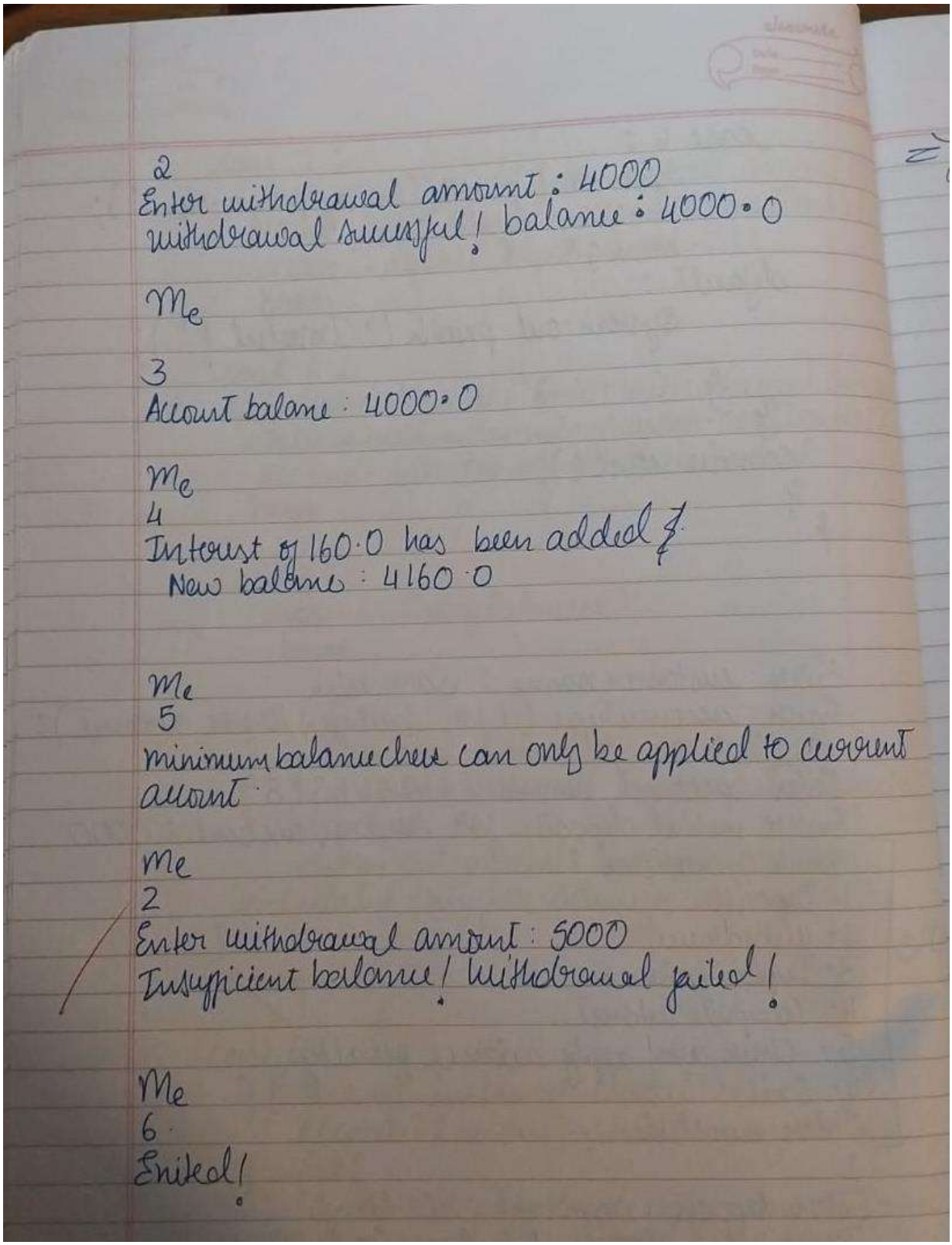


```
        case 6:
            System.out.println("Exiting program");
            running = false;
            break;
        default:
            System.out.println("Invalid!");

    }
    scanner.close();
}
```

```
Enter customer name : Sameksha
Enter accountType (1 for Savings, 2 for current):
1
Enter account number : 2345SE59
Enter initial deposit for savings account : 5000
Bank Operations :
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
5. Check and apply balance penalty
6. Exit
Enter your choice :
1
Enter deposit amount : 3000
Deposited ! current balance : 8000.0
```

Me

Me



2
Enter withdrawal amount : 4000
withdrawal successful! balance : 4000.0

Me

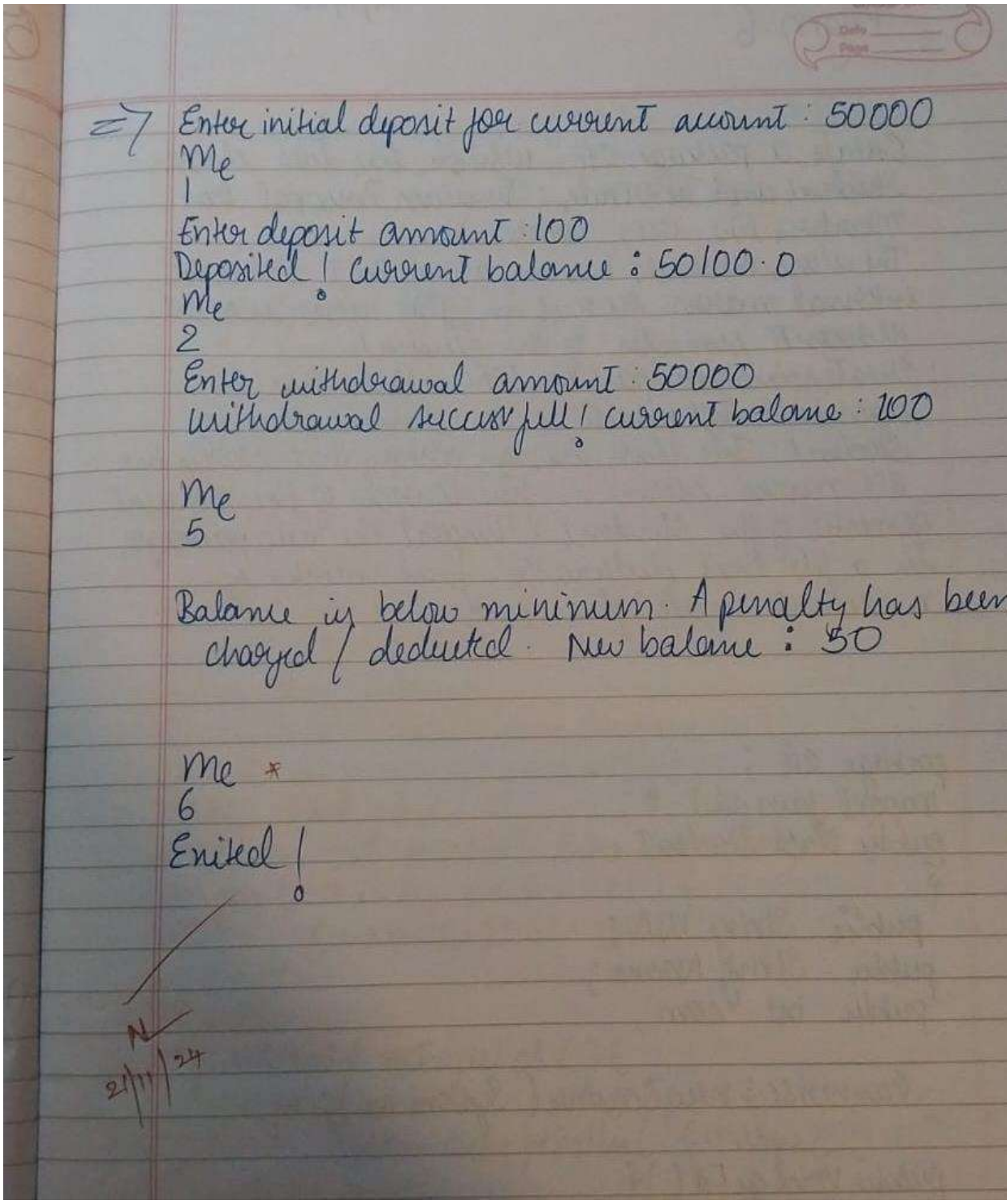
3
Account balance : 4000.0

Me
4
Interest of 160.0 has been added!
New balance : 4160.0

Me
5
minimum balance checks can only be applied to current account.

Me
2
Enter withdrawal amount : 5000
Insufficient balance! withdrawal failed!

Me
6
Exited!


=7 Enter initial deposit for current account : 50000
Me
1
Enter deposit amount : 100
Deposited ! current balance : 50100.0
Me
2
Enter withdrawal amount : 50000
withdrawal successfull ! current balance : 100

Me
5

Balance is below minimum. A penalty has been
charged / deducted. New balance : 50

Me *
6
Exited !

21/11/24

```
import java.util.Scanner;

class Account {

    protected String customerName;

    protected String accountNumber;

    protected double balance;

    protected String accountType;

  public Account(String customerName, String accountNumber, String accountType, double initialBalance) {

        this.customerName = customerName;

        this.accountNumber = accountNumber;

        this.accountType = accountType;

        this.balance = initialBalance;

    }

    public void deposit(double amount) {

        balance += amount;

        System.out.println("Deposit successful! Current balance: " + balance);

    }

    public void displayBalance() {

        System.out.println("Account balance: " + balance);

    }

  public void withdraw(double amount) {

        if (balance >= amount) {
```

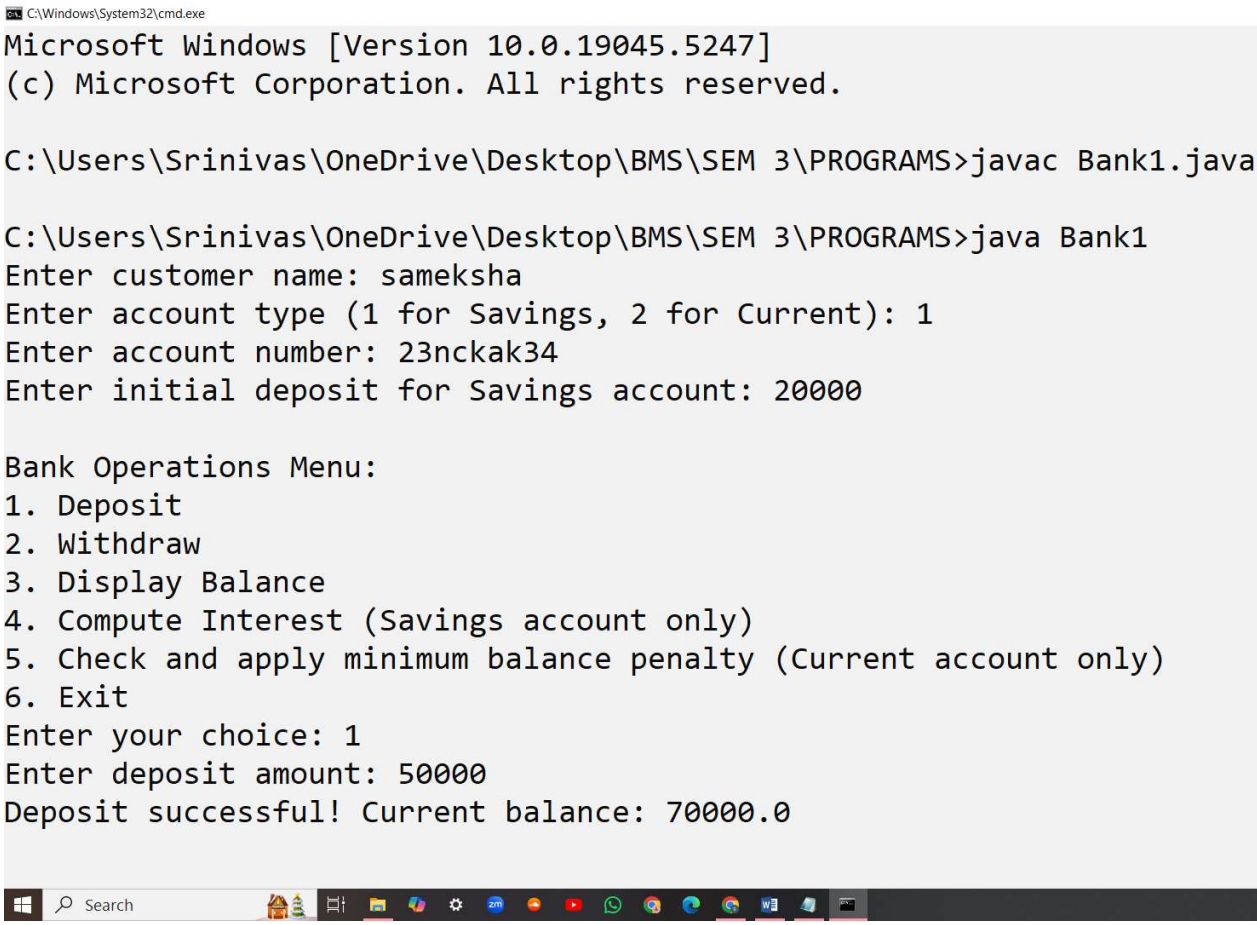
```
            balance -= amount;
            System.out.println("Withdrawal successful! Current balance: " + balance);
        } else {
            System.out.println("Insufficient balance! Withdrawal failed.");
        }
    }
    public String getAccountType() {
        return accountType;
    }
}
class SavAcct extends Account {
    private static final double interestRate = 0.04; // 4% interest rate for savings account
    public SavAcct(String customerName, String accountNumber, double initialBalance) {
        super(customerName, accountNumber, "Savings", initialBalance);
    }
public void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest of " + interest + " has been added to your account. New balance: " + balance);
    }
}
class CurAcct extends Account {
    private static final double MIN_BALANCE = 500; // Minimum balance required for Current Account
```

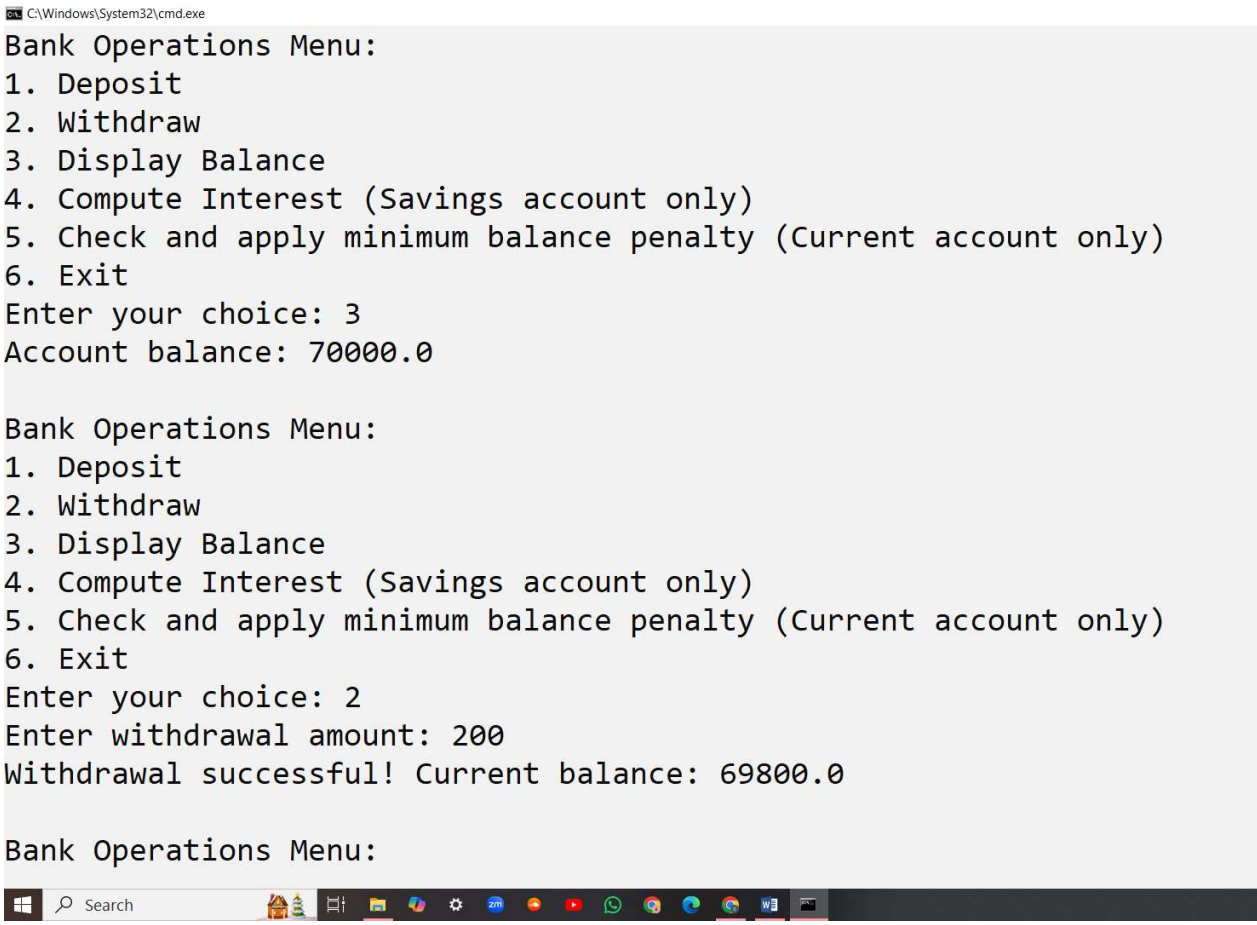
```
    private static final double PENALTY = 50; // Penalty fee for falling below minimum balance

    public CurAcct(String customerName, String accountNumber, double initialBalance) {
        super(customerName, accountNumber, "Current", initialBalance);
    }

    public void checkMinimumBalance() {
        if (balance < MIN_BALANCE) {
            balance -= PENALTY;
            System.out.println("Balance is below minimum. A penalty of " + PENALTY + " has been charged. New balance: " + balance);
        }
    }
}

public class Bank1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter customer name: ");
        String customerName = scanner.nextLine();
        System.out.print("Enter account type (1 for Savings, 2 for Current): ");
        int accountChoice = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter account number: ");
```

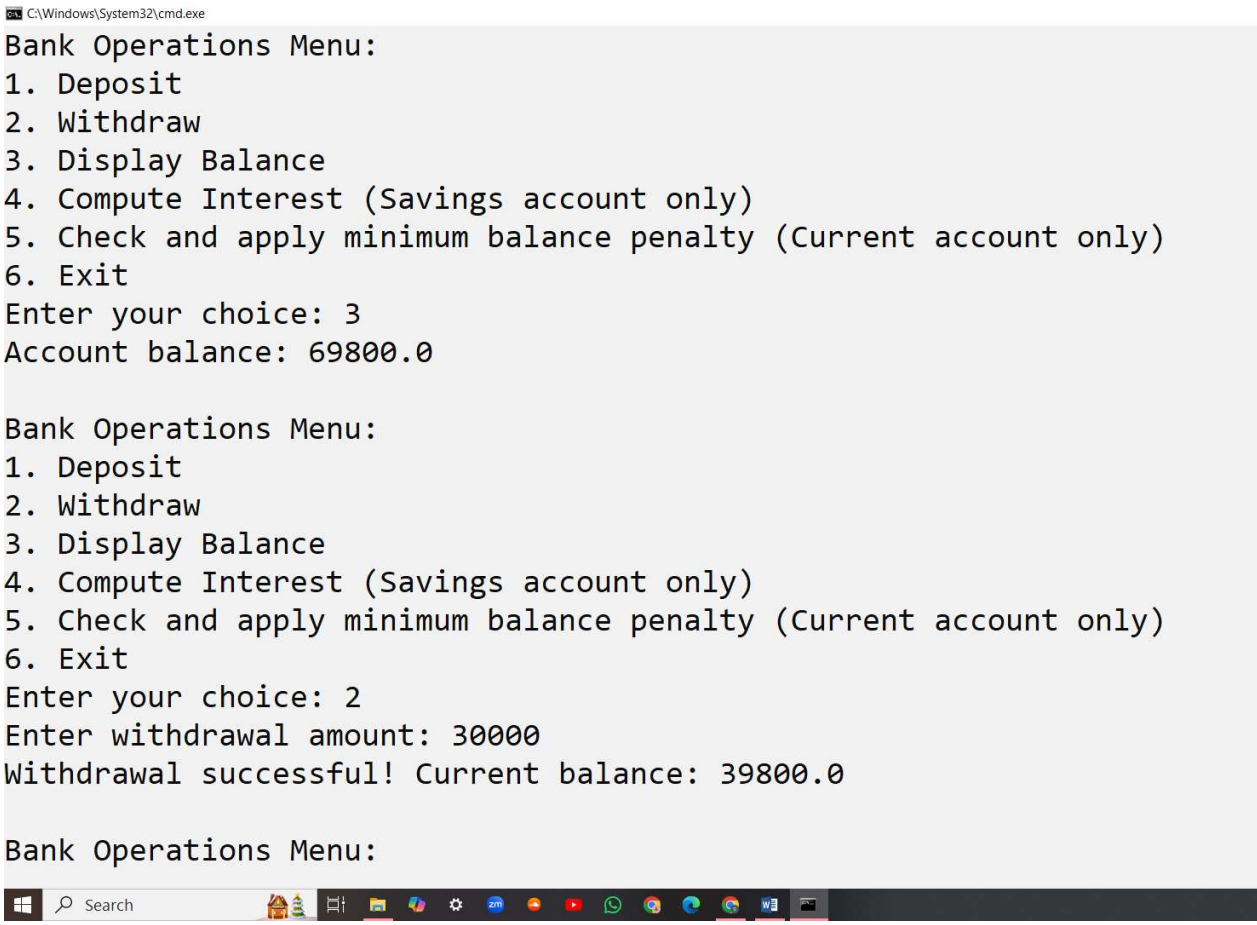
```
        String accountNumber = scanner.nextLine();
        Account account = null;
        if (accountChoice == 1) {
            System.out.print("Enter initial deposit for Savings account: ");
            double initialDeposit = scanner.nextDouble();
            account = new SavAcct(customerName, accountNumber, initialDeposit);
        } else if (accountChoice == 2) {
            System.out.print("Enter initial deposit for Current account: ");
            double initialDeposit = scanner.nextDouble();
            account = new CurAcct(customerName, accountNumber, initialDeposit);
        } else {
            System.out.println("Invalid choice! Exiting program.");
            return;
        }
        boolean running = true;
        while (running) {
            System.out.println("\nBank Operations Menu:");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display Balance");
            System.out.println("4. Compute Interest (Savings account only)");
            System.out.println("5. Check and apply minimum balance penalty (Current account only)");
            System.out.println("6. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
```

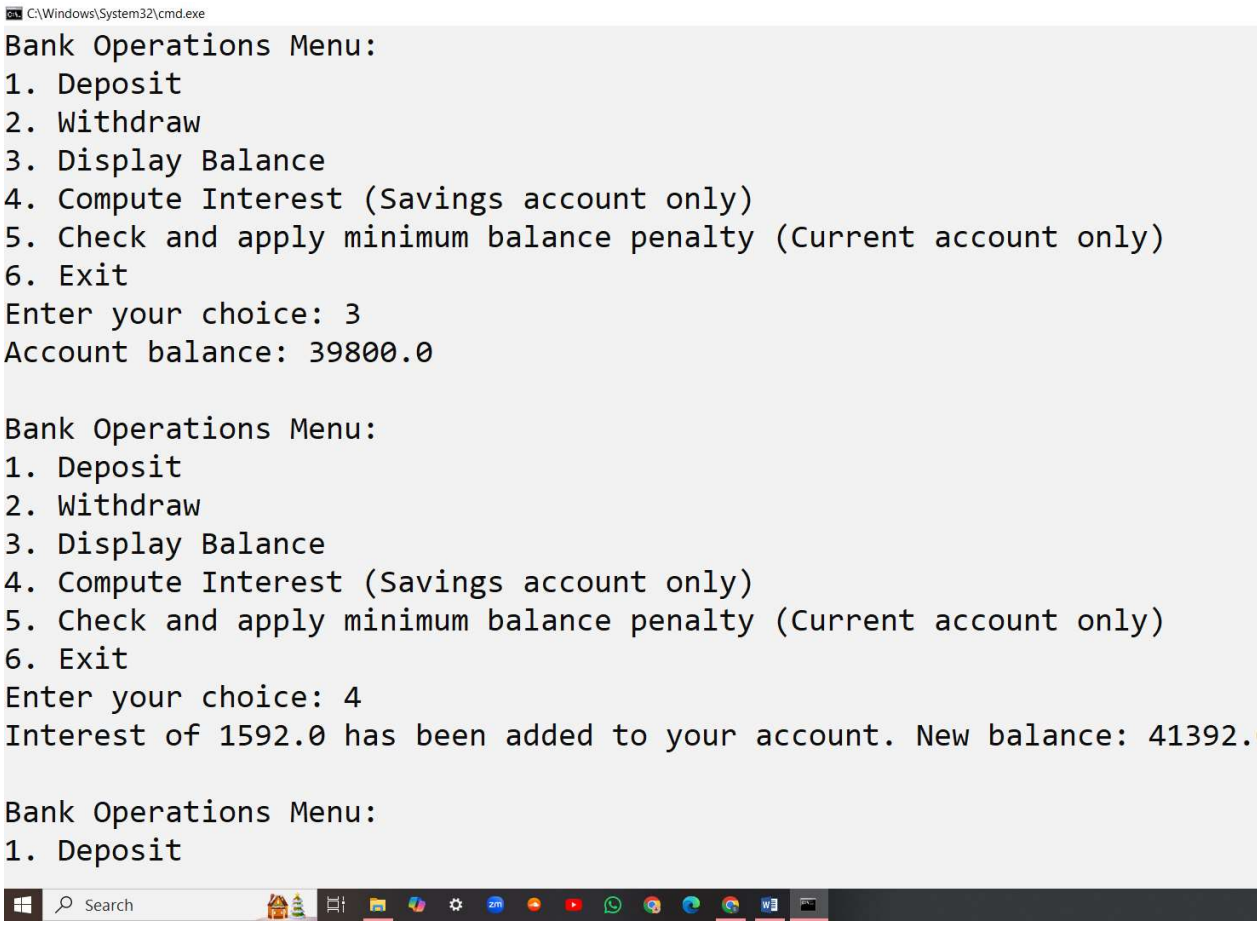
```
            switch (choice) {
                case 1:
                    System.out.print("Enter deposit amount: ");
                    double depositAmount = scanner.nextDouble();
                    account.deposit(depositAmount);
                    break;
                case 2:
                    System.out.print("Enter withdrawal amount: ");
                    double withdrawAmount = scanner.nextDouble();
                    account.withdraw(withdrawAmount);
                    break;
                case 3:
                    account.displayBalance();
                    break;
                case 4:
                    if (account instanceof SavAcct) {
                        ((SavAcct) account).computeInterest();
                    } else {
                        System.out.println("Interest can only be computed for Savings account.");
                    }
                    break;
                case 5:
                    if (account instanceof CurAcct) {
                        ((CurAcct) account).checkMinimumBalance();
                    } else {
```

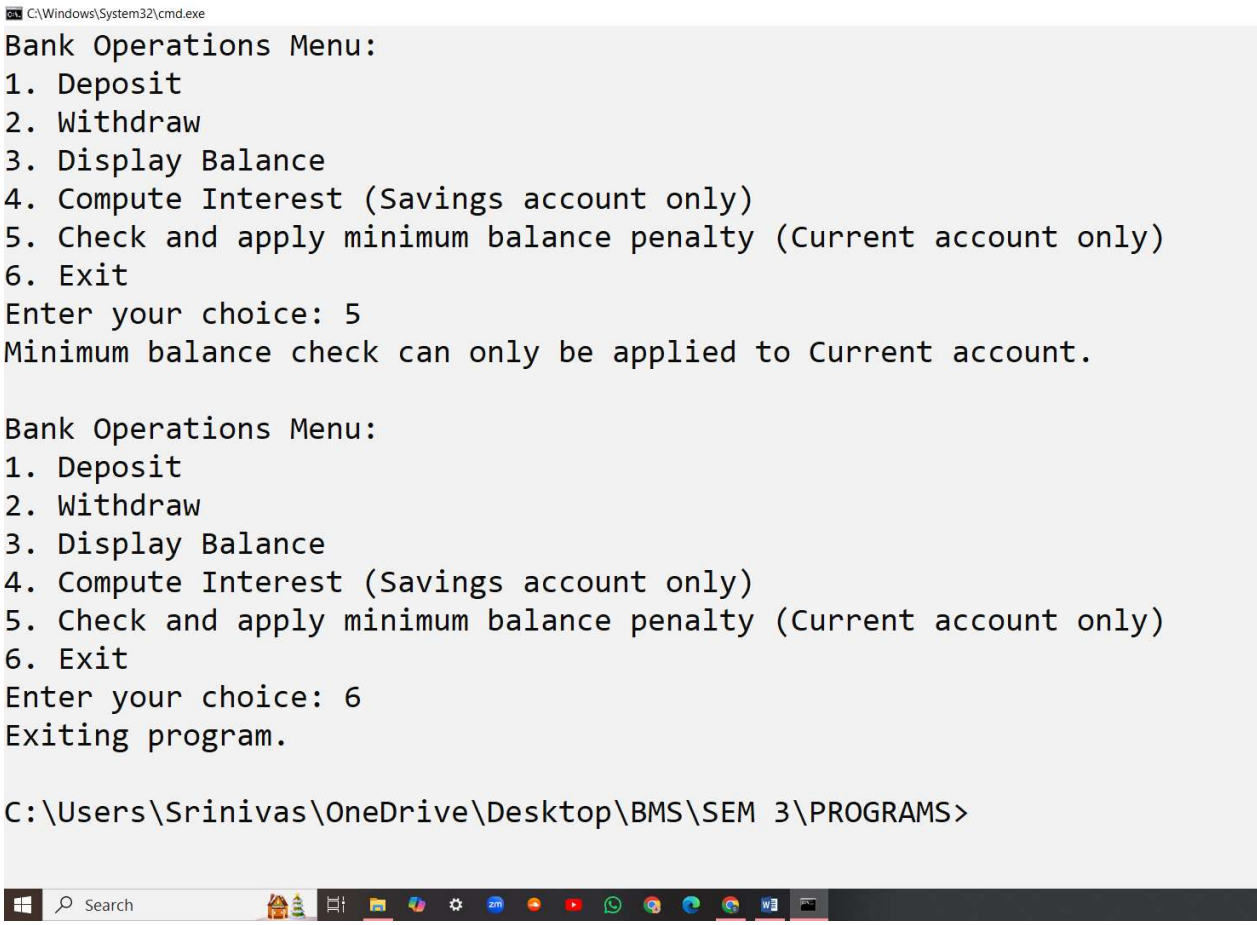
```
                    System.out.println("Minimum balance check can only be applied to Current account.");
                }
                break;
            case 6:
                System.out.println("Exiting program.");
                running = false;
                break;
            default:
                System.out.println("Invalid choice! Please select a valid option.");
        }
    }

    scanner.close();
  }
}
```


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Srinivas\OneDrive\Desktop\BMS\SEM 3\PROGRAMS>javac Bank1.java
C:\Users\Srinivas\OneDrive\Desktop\BMS\SEM 3\PROGRAMS>java Bank1
Enter customer name: sameksha
Enter account type (1 for Savings, 2 for Current): 1
Enter account number: 23nckak34
Enter initial deposit for Savings account: 20000
Bank Operations Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings account only)
5. Check and apply minimum balance penalty (Current account only)
6. Exit
Enter your choice: 1
Enter deposit amount: 50000
Deposit successful! Current balance: 70000.0
Search


C:\Windows\System32\cmd.exe
Bank Operations Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings account only)
5. Check and apply minimum balance penalty (Current account only)
6. Exit
Enter your choice: 3
Account balance: 70000.0
Bank Operations Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings account only)
5. Check and apply minimum balance penalty (Current account only)
6. Exit
Enter your choice: 2
Enter withdrawal amount: 200
Withdrawal successful! Current balance: 69800.0
Bank Operations Menu:
Search


C:\Windows\System32\cmd.exe
Bank Operations Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings account only)
5. Check and apply minimum balance penalty (Current account only)
6. Exit
Enter your choice: 3
Account balance: 69800.0
Bank Operations Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings account only)
5. Check and apply minimum balance penalty (Current account only)
6. Exit
Enter your choice: 2
Enter withdrawal amount: 30000
Withdrawal successful! Current balance: 39800.0
Bank Operations Menu:
Search


C:\Windows\System32\cmd.exe
Bank Operations Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings account only)
5. Check and apply minimum balance penalty (Current account only)
6. Exit
Enter your choice: 3
Account balance: 39800.0
Bank Operations Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings account only)
5. Check and apply minimum balance penalty (Current account only)
6. Exit
Enter your choice: 4
Interest of 1592.0 has been added to your account. New balance: 41392.
Bank Operations Menu:
1. Deposit
Search


C:\Windows\System32\cmd.exe
Bank Operations Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings account only)
5. Check and apply minimum balance penalty (Current account only)
6. Exit
Enter your choice: 5
Minimum balance check can only be applied to Current account.
Bank Operations Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings account only)
5. Check and apply minimum balance penalty (Current account only)
6. Exit
Enter your choice: 6
Exiting program.
C:\Users\Srinivas\OneDrive\Desktop\BMS\SEM 3\PROGRAMS>
Search