

Lab Program 5

```
import java.util.Scanner;
```

```
class Account{
```

```
    String customername;
```

```
    int accountnumber;
```

```
    String accounttype;
```

```
    double balance;
```

```
    Scanner scanner=new Scanner(System.in);
```

```
    void getAccountDetails(){
```

```
        System.out.println("Enter Customer name");
```

```
        customername=scanner.nextLine();
```

```
        System.out.println("Enter account number");
```

```
        accountnumber=scanner.nextInt();
```

```
        System.out.println("Enter Account Type");
```

```
        accounttype=scanner.nextLine();
```

```
        System.out.println("Enter Account Balance");
```

```
        balance=scanner.nextInt();
```

```
    }
```

```
    void deposit(){
```

```
        System.out.println("Enter amount to deposit");
```

```
        double amount=scanner.nextDouble();
```

```
        balance += amount;
```

```
        System.out.println("Amount deposited");
```

```
    }
```

```
    void displaybalance(){
```

```
        System.out.println("Account Holder:"+customername);
```

```
        System.out.println("Account Number:"+accountnumber);
```

```
        System.out.println("Account Type:"+accounttype);
```

```

        System.out.println("Current Balance"+balance);
    }
}

class Savacct extends Account{

    final double interestrate=0.05;

    void computeInterest(){

        System.out.println("Enter the time period in years");

        double time=scanner.nextDouble();

        double interest= (balance*interestrate*time)/100;

        balance+=interest;

    }

    void withdraw(){

        System.out.println("Enter amount to withdraw");

        double amount=scanner.nextDouble();

        if(amount<=balance){

            balance-=amount;

            System.out.println("Withdrawn");

        }

        else{

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

        }

    }

}

class Curacct extends Account{

    final double minimumbalance=1000;

    final double servicecharge=100;

    void checkminimumbalance(){

        if(balance<minimumbalance){

```

```

        balance-=servicecharge;

        System.out.println("Service charge imposed");
    }
    else{

        System.out.println("Minimum balance is there");
    }
}

void withdraw(){

    System.out.println("Enter amount to withdraw");

    double amount=scanner.nextDouble();

    if(amount<=balance){

        balance-=amount;

        System.out.println("Withdrawn");

    }

    else{

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

    }

}

}

public class BankDemo {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Welcome to Bank System ");

        System.out.print("Enter account type (savings/current): ");

        String type = scanner.nextLine().toLowerCase();

        if (type.equals("savings")) {

            Savacct s = new Savacct();

            s.getAccountDetails();

```

```
while (true) {  
    System.out.println("\n--- Savings Account Menu ---");  
    System.out.println("1. Deposit");  
    System.out.println("2. Withdraw");  
    System.out.println("3. Compute Interest");  
    System.out.println("4. Display Balance");  
    System.out.println("5. Exit");  
    System.out.print("Enter your choice: ");  
    int ch = scanner.nextInt();  
    switch (ch) {  
        case 1:  
            s.deposit();  
            break;  
        case 2:  
            s.withdraw();  
            break;  
        case 3:  
            s.computeInterest();  
            break;  
        case 4:  
            s.displaybalance();  
            break;  
        case 5:  
            System.out.println("Thank you for using our bank!");  
            return;  
        default:  
            System.out.println("Invalid choice!");  
    }  
}
```

```
}
```

```
} else if (type.equals("current")) {  
    Curacct c = new Curacct();  
    c.getAccountDetails();  
    while (true) {  
        System.out.println("\n--- Current Account Menu ---");  
        System.out.println("1. Deposit");  
        System.out.println("2. Withdraw");  
        System.out.println("3. Check Minimum Balance");  
        System.out.println("4. Display Balance");  
        System.out.println("5. Exit");  
        System.out.print("Enter your choice: ");  
        int ch = scanner.nextInt();  
        switch (ch) {  
            case 1:  
                c.deposit();  
                break;  
            case 2:  
                c.withdraw();  
                break;  
            case 3:  
                c.checkminimumbalance();  
                break;  
            case 4:  
                c.displaybalance();  
                break;  
            case 5:
```

```

        System.out.println("Thank you for using our bank!");

        return;

    default:

        System.out.println("Invalid choice!");

    }

}

} else {

    System.out.println("Invalid account type entered!");

}

}

}

```

The screenshot shows a Java IDE with a dark theme. The main editor displays a Java program for a bank system. The code includes a welcome message, prompts for account type, customer name, account number, and account balance. It then presents a menu for the Savings Account with options: 1. Deposit, 2. Withdraw, 3. Compute Interest, 4. Display Balance, and 5. Exit. The user has entered 'Savings' for account type, 'Shivam' for customer name, '456' for account number, and '5000' for account balance. The program has processed these inputs and displayed the Savings Account Menu. The user has selected option 1 (Deposit) and entered '4000' for the amount to deposit. The program has confirmed the deposit. The user has then selected option 4 (Display Balance) and the program has displayed the account details: Account Holder: Shivam, Account Number: 456, Account Type: Current, and Current Balance: 9000.0. The user has then selected option 2 (Withdraw) and entered an amount to withdraw.

On the right side of the IDE, there is a sidebar for an AI assistant. The sidebar has a header 'Build with Agent' and a sub-header 'SUGGESTED ACTIONS'. Below the header, there is a button 'Build Workspace' and a button 'Show Config'. Below the suggested actions, there is a section for 'J BankDemo.java' with a plus sign. Below this, there is a text input field 'Describe what to build next' and a button 'Agent' with a dropdown menu, a button 'Auto' with a dropdown menu, and a button 'Run' with a play icon.

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + - - - x CHAT + - - - x
```

```
4. Display Balance
5. Exit
Enter your choice: 4
Account Holder:Shivam
Account Number:456
Account Type:
Current Balance9000.0

--- Savings Account Menu ---
1. Deposit
2. Withdraw
3. Compute Interest
4. Display Balance
5. Exit
Enter your choice: 2
Enter amount to withdraw
6000
Withdrawn

--- Savings Account Menu ---
1. Deposit
2. Withdraw
3. Compute Interest
4. Display Balance
5. Exit
Enter your choice: 3
Enter the time period in years
4

--- Savings Account Menu ---
1. Deposit
2. Withdraw
3. Compute Interest
4. Display Balance
5. Exit
Enter your choice: 4
Account Holder:Shivam
Account Number:456
Account Type:
Current Balance3006.0
```

Build with Agent

AI responses may be inaccurate.
[Generate Agent Instructions](#) to onboard AI onto your codebase.

SUGGESTED ACTIONS

Build Workspace Show Config

BankDemo.java +

Describe what to build next

Agent Auto

Ln 161, Col 1 Spaces: 4 UTF-8 CRLF {} Java

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS java + - - - x CHAT + - - - x
```

```
PS C:\Users\Saanvi\OneDrive\Desktop\oops_1BF424CS260> java BankDemo
Welcome to Bank System
Enter account type (savings/current): current
Enter Customer name
Divya
Enter account number
963
Enter Account Type
Enter Account Balance
7000

--- Current Account Menu ---
1. Deposit
2. Withdraw
3. Check Minimum Balance
4. Display Balance
5. Exit
Enter your choice: 2
Enter amount to withdraw
5000
Withdrawn

--- Current Account Menu ---
1. Deposit
2. Withdraw
3. Check Minimum Balance
4. Display Balance
5. Exit
Enter your choice: 3
Minimum balance is there

--- Current Account Menu ---
1. Deposit
2. Withdraw
3. Check Minimum Balance
4. Display Balance
5. Exit
Enter your choice: 2
Enter amount to withdraw
900
Withdrawn
```

Build with Agent

AI responses may be inaccurate.
[Generate Agent Instructions](#) to onboard AI onto your codebase.

SUGGESTED ACTIONS

Build Workspace Show Config

BankDemo.java +

Describe what to build next

Agent Auto

Ln 161, Col 1 Spaces: 4 UTF-8 CRLF {} Java

java + - CHAT + - ...

```

--- Current Account Menu ---
1. Deposit
2. Withdraw
3. Check Minimum Balance
4. Display Balance
5. Exit
Enter your choice: 4
Account Holder:Divya
Account Number:963
Account Type:
Current Balance0.0

```