

Institute of Computer Technology
B. Tech. Computer Science and Engineering

Semester: III

Sub: Object-Oriented Programming

Course Code: 2CSE303

Practical Number: 10

Objective:

To learn about inheritance concepts in java.

Q1. Make an appropriate simple “Account management” project based practical by using inheritance concepts like (Single L. Inheritance, Multi L. Inheritance , Hierarchical L. Inheritance and Hybrid L. Inheritance)

In project, you have to include the following modules:

1. Create customer accounts in the bank.
 2. Deposit amount in Account.
 3. Withdraw amount from Account.
 4. Transfer amount from one account to another account.
 5. Display balance by Account ID or Name
 6. Modify customer account information.
 7. Delete Account
 8. Display monthly transaction Report.
- **Note:1:** You have to create minimum five customer account in bank and then have to perform all others module accordingly.
 - **Note:2:** After using inheritance concept, you can keep all your modules in switch case condition or in a do-while loop condition for a menu driven or choice based program.

Code :

```
import java.util.ArrayList;
```

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

```
// Base class
```

```
class Account {
```

```
    protected int accountId;
```

```
protected String accountHolderName;

protected double balance;

public Account(int accountId, String accountHolderName, double balance) {
    this.accountId = accountId;
    this.accountHolderName = accountHolderName;
    this.balance = balance;
}

public void deposit(double amount) {
    balance += amount;
    System.out.println("Deposit Successful! New Balance: " + balance);
}

public void withdraw(double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println("Withdrawal Successful! New Balance: " + balance);
    } else {
        System.out.println("Insufficient Balance!");
    }
}

public double getBalance() {
    return balance;
}

public String getAccountHolderName() {
    return accountHolderName;
}
```

```
    public int getAccountId() {  
        return accountId;  
    }  
}
```

// SavingsAccount and CurrentAccount (Hierarchical Inheritance)

```
class SavingsAccount extends Account {  
    public SavingsAccount(int accountId, String accountHolderName, double balance) {  
        super(accountId, accountHolderName, balance);  
    }  
}
```

```
class CurrentAccount extends Account {  
    public CurrentAccount(int accountId, String accountHolderName, double balance) {  
        super(accountId, accountHolderName, balance);  
    }  
}
```

// Transaction class (Multilevel Inheritance)

```
class Transaction extends SavingsAccount {  
    public Transaction(int accountId, String accountHolderName, double balance) {  
        super(accountId, accountHolderName, balance);  
    }  
}
```

```
    public void transfer(Account recipient, double amount) {  
        if (amount <= balance) {  
            this.withdraw(amount);  
            recipient.deposit(amount);  
            System.out.println("Transfer Successful!");  
        } else {  
            System.out.println("Insufficient Balance for Transfer!");  
        }  
    }
```

```
    }  
    }  
}  
  
// AccountManager (Hybrid Inheritance)  
class AccountManager {  
    private ArrayList<Account> accounts = new ArrayList<>();  
  
    public void createAccount(String type, int accountId, String name, double initialBalance) {  
        Account account = type.equals("Savings") ? new SavingsAccount(accountId, name,  
initialBalance)  
                : new CurrentAccount(accountId, name, initialBalance);  
        accounts.add(account);  
        System.out.println(type + " Account Created Successfully!");  
    }  
  
    public Account findAccount(int accountId) {  
        for (Account account : accounts) {  
            if (account.getAccountId() == accountId) {  
                return account;  
            }  
        }  
        return null;  
    }  
  
    public void deleteAccount(int accountId) {  
        Account account = findAccount(accountId);  
        if (account != null) {  
            accounts.remove(account);  
            System.out.println("Account Deleted Successfully!");  
        } else {
```

```
        System.out.println("Account Not Found!");
    }
}

public void displayAccount(int accountId) {
    Account account = findAccount(accountId);
    if (account != null) {
        System.out.println("Account ID: " + account.getAccountId() +
            ", Name: " + account.getAccountHolderName() +
            ", Balance: " + account.getBalance());
    } else {
        System.out.println("Account Not Found!");
    }
}

public void modifyAccount(int accountId, String newName) {
    Account account = findAccount(accountId);
    if (account != null) {
        account.accountHolderName = newName;
        System.out.println("Account Modified Successfully!");
    } else {
        System.out.println("Account Not Found!");
    }
}

public void displayMonthlyReport() {
    System.out.println("Monthly Transaction Report:");
    for (Account account : accounts) {
        System.out.println("Account ID: " + account.getAccountId() +
            ", Name: " + account.getAccountHolderName() +
            ", Balance: " + account.getBalance());
    }
}
```

```
    }  
    }  
}
```

// Main Class for Menu-driven Approach

```
public class BankManagementSystem {  
    public static void main(String[] args) {  
        AccountManager accountManager = new AccountManager();  
        Scanner scanner = new Scanner(System.in);  
        int choice;  
  
        do {  
            System.out.println("\n1. Create Account\n2. Deposit\n3. Withdraw\n4. Transfer\n5.  
Display Account\n6. Modify Account\n7. Delete Account\n8. Monthly Report\n9. Exit");  
            System.out.print("Enter your choice: ");  
            choice = scanner.nextInt();  
  
            switch (choice) {  
                case 1:  
                    System.out.print("Enter Account Type (Savings/Current): ");  
                    String type = scanner.next();  
                    System.out.print("Enter Account ID: ");  
                    int id = scanner.nextInt();  
                    System.out.print("Enter Account Holder Name: ");  
                    String name = scanner.next();  
                    System.out.print("Enter Initial Balance: ");  
                    double balance = scanner.nextDouble();  
                    accountManager.createAccount(type, id, name, balance);  
                    break;  
                case 2:  
                    System.out.print("Enter Account ID: ");
```

```
id = scanner.nextInt();  
System.out.print("Enter Amount to Deposit: ");  
double depositAmount = scanner.nextDouble();  
Account account = accountManager.findAccount(id);  
if (account != null) {  
    account.deposit(depositAmount);  
} else {  
    System.out.println("Account Not Found!");  
}  
break;
```

case 3:

```
System.out.print("Enter Account ID: ");  
id = scanner.nextInt();  
System.out.print("Enter Amount to Withdraw: ");  
double withdrawAmount = scanner.nextDouble();  
account = accountManager.findAccount(id);  
if (account != null) {  
    account.withdraw(withdrawAmount);  
} else {  
    System.out.println("Account Not Found!");  
}  
break;
```

case 4:

```
System.out.print("Enter Sender Account ID: ");  
int senderId = scanner.nextInt();  
System.out.print("Enter Recipient Account ID: ");  
int recipientId = scanner.nextInt();  
System.out.print("Enter Transfer Amount: ");  
double transferAmount = scanner.nextDouble();  
Account sender = accountManager.findAccount(senderId);  
Account recipient = accountManager.findAccount(recipientId);
```



```
        if (sender != null && recipient != null) {  
            ((Transaction) sender).transfer(recipient, transferAmount);  
        } else {  
            System.out.println("Account Not Found!");  
        }  
        break;  
case 5:  
    System.out.print("Enter Account ID: ");  
    id = scanner.nextInt();  
    accountManager.displayAccount(id);  
    break;  
case 6:  
    System.out.print("Enter Account ID: ");  
    id = scanner.nextInt();  
    System.out.print("Enter New Account Holder Name: ");  
    name = scanner.next();  
    accountManager.modifyAccount(id, name);  
    break;  
case 7:  
    System.out.print("Enter Account ID to Delete: ");  
    id = scanner.nextInt();  
    accountManager.deleteAccount(id);  
    break;  
case 8:  
    accountManager.displayMonthlyReport();  
    break;  
case 9:  
    System.out.println("Exiting System...");  
    break;  
default:  
    System.out.println("Invalid Choice! Please try again.");
```

```
    }  
    } while (choice != 9);  
  
    scanner.close();  
}  
}
```

Output :

1. Create Account
2. Deposit
3. Withdraw
4. Transfer
5. Display Account
6. Modify Account
7. Delete Account
8. Monthly Report
9. Exit

Enter your choice: 1

Enter Account Type (Savings/Current): Savings

Enter Account ID: 1

Enter Account Holder Name: vishva

Enter Initial Balance: 50000

Savings Account Created Successfully!

1. Create Account
2. Deposit
3. Withdraw
4. Transfer
5. Display Account
6. Modify Account
7. Delete Account
8. Monthly Report

9. Exit

Enter your choice: 1

Enter Account Type (Savings/Current): Current

Enter Account ID: 2

Enter Account Holder Name: Aryan

Enter Initial Balance: 100000

Current Account Created Successfully!

1. Create Account

2. Deposit

3. Withdraw

4. Transfer

5. Display Account

6. Modify Account

7. Delete Account

8. Monthly Report

9. Exit

Enter your choice: 1

Enter Account Type (Savings/Current): savings

Enter Account ID: 3

Enter Account Holder Name: jeal

Enter Initial Balance: 50000

savings Account Created Successfully!

1. Create Account

2. Deposit

3. Withdraw

4. Transfer

5. Display Account

6. Modify Account

7. Delete Account

8. Monthly Report

9. Exit

Enter your choice: 1

Enter Account Type (Savings/Current): savings

Enter Account ID: 4

Enter Account Holder Name: rishi

Enter Initial Balance: 50000

savings Account Created Successfully!

1. Create Account

2. Deposit

3. Withdraw

4. Transfer

5. Display Account

6. Modify Account

7. Delete Account

8. Monthly Report

9. Exit

Enter your choice: 1

Enter Account Type (Savings/Current): savings

Enter Account ID: 5

Enter Account Holder Name: Nisarg

Enter Initial Balance: 50000

savings Account Created Successfully!

1. Create Account

2. Deposit

3. Withdraw

4. Transfer

5. Display Account

6. Modify Account

7. Delete Account
8. Monthly Report
9. Exit

Enter your choice: 2

Enter Account ID: 1

Enter Amount to Deposit: 50000

Deposit Successful! New Balance: 100000.0

1. Create Account
2. Deposit
3. Withdraw
4. Transfer
5. Display Account
6. Modify Account
7. Delete Account
8. Monthly Report
9. Exit

Enter your choice: 3

Enter Account ID: 2

Enter Amount to Withdraw: 50000

Withdrawal Successful! New Balance: 50000.0

1. Create Account
2. Deposit
3. Withdraw
4. Transfer
5. Display Account
6. Modify Account
7. Delete Account
8. Monthly Report
9. Exit

Enter your choice: 4

Enter Sender Account ID: 1

Enter Recipient Account ID: 3

Enter Transfer Amount: 25000

Withdrawal Successful! New Balance: 75000.0

Deposit Successful! New Balance: 75000.0

Transfer Successful!

1. Create Account
2. Deposit
3. Withdraw
4. Transfer
5. Display Account
6. Modify Account
7. Delete Account
8. Monthly Report
9. Exit

Enter your choice: 5

Enter Account ID: 1

Account ID: 1, Name: vishva, Balance: 75000.0

1. Create Account
2. Deposit
3. Withdraw
4. Transfer
5. Display Account
6. Modify Account
7. Delete Account
8. Monthly Report
9. Exit

Enter your choice: 6

Enter Account ID: 1

Enter New Account Holder Name: Vishva_Patel

Account Modified Successfully!

1. Create Account
2. Deposit
3. Withdraw
4. Transfer
5. Display Account
6. Modify Account
7. Delete Account
8. Monthly Report
9. Exit

Enter your choice: 7

Enter Account ID to Delete: 5

Account Deleted Successfully!

1. Create Account
2. Deposit
3. Withdraw
4. Transfer
5. Display Account
6. Modify Account
7. Delete Account
8. Monthly Report
9. Exit

Enter your choice: 8

Monthly Transaction Report:

Account ID: 1, Name: Vishva_Patel, Balance: 75000.0

Account ID: 2, Name: Aryan, Balance: 50000.0

Account ID: 3, Name: jeal, Balance: 75000.0

Account ID: 4, Name: rishi, Balance: 50000.0

1. Create Account
2. Deposit
3. Withdraw
4. Transfer
5. Display Account
6. Modify Account
7. Delete Account
8. Monthly Report
9. Exit

Enter your choice: 9