```
import java.util.Scanner;
abstract class Shape
{
   int l,b;
   double a;
   Scanner ss=new Scanner(System.in);
   Shape()
   {
   System.out.println("enter l:");
   l=ss.nextInt();
   System.out.println("enter b:");
   b=ss.nextInt();
   }
   void printArea()
   {

   }
}
class Rectangle extends Shape
{

   void printArea()
   {
      a=l*b;

      System.out.println("area is :"+this.a);
}
   }


class Triangle extends Shape
{
   void printArea()
   {
      a=0.5*l*b;
      System.out.println("area is :"+this.a);
   }

}
class Circle extends Shape
{
   void printArea()
   {
      a=3.14*l*l;
```

```
        System.out.println("area is :"+this.a);
    }

}
public class MainShape
{
    public static void main(String[] args)
    {
     //Shape s1=new Shape();
     Rectangle s2=new Rectangle();
     Triangle s3=new Triangle();
     Circle s4=new Circle();
     //s1.printArea();
     s2.printArea();
     s3.printArea();
     s4.printArea();
    }
}
```

```
D:\3 sem\java\java prgms>java MainShape
enter l:
10
enter b:
10
enter l:
10
enter b:
10
enter l:
10
enter b:
10
area is :100.0
area is :50.0
area is :314.0
```

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

// Base Account class
class Account {
    String customerName;
    String accountNumber;
    String accountType;
    double balance;

    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) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful. Updated balance: " + balance);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

    public void displayBalance() {
        System.out.println("Account Holder: " + customerName + ", Account Number: " +
accountNumber +
                ", Current Balance: " + balance);
    }
```

```
}

// Savings Account class
class SavAcct extends Account {
    private final double interestRate = 0.04; // 4% annual interest rate

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

    public void computeAndDepositInterest(int years) {
        if (years > 0) {
            double interest = balance * Math.pow(1 + interestRate, years) - balance;
            balance += interest;
            System.out.println("Interest of " + interest + " has been added. Updated balance: " + balance);
        } else {
            System.out.println("Invalid number of years.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient balance.");
        }
    }
}

// Current Account class
```

```
class CurAcct extends Account {
    private final double minBalance = 1000.0;
    private final double penalty = 50.0;

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

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            if (balance < minBalance) {
                balance -= penalty;
                System.out.println("Penalty of " + penalty + " imposed for falling below minimum
balance.");
            }
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient balance.");
        }
    }
}

// Main Bank class
public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        ArrayList<Account> accounts = new ArrayList<>();

        System.out.print("Enter the number of users: ");
        int n = sc.nextInt();
```

```
// Creating accounts for users
for (int i = 0; i < n; i++) {
    System.out.println("\nEnter details for user " + (i + 1) + ":");
    System.out.print("Enter customer name: ");
    sc.nextLine(); // Consume newline
    String name = sc.nextLine();
    System.out.print("Enter account number: ");
    String accNo = sc.nextLine();
    System.out.print("Enter initial balance: ");
    double initialBalance = sc.nextDouble();
    System.out.print("Enter account type (1 for Savings, 2 for Current): ");
    int accType = sc.nextInt();

    if (accType == 1) {
        accounts.add(new SavAcct(name, accNo, initialBalance));
    } else if (accType == 2) {
        accounts.add(new CurAcct(name, accNo, initialBalance));
    } else {
        System.out.println("Invalid account type. Skipping user.");
    }
}

// Menu for operations
int choice;
do {
    System.out.println("\n=== Bank Operations Menu ===");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Compute Interest (Savings only)");
    System.out.println("4. Display Balance");
```

```
System.out.println("5. Exit");
System.out.print("Enter your choice: ");
choice = sc.nextInt();

switch (choice) {
   case 1: // Deposit
      System.out.print("Enter account number: ");
      sc.nextLine(); // Consume newline
      String accNo = sc.nextLine();
      Account acc = findAccount(accounts, accNo);
      if (acc != null) {
         System.out.print("Enter deposit amount: ");
         double amount = sc.nextDouble();
         acc.deposit(amount);
      } else {
         System.out.println("Account not found.");
      }
      break;

   case 2: // Withdraw
      System.out.print("Enter account number: ");
      sc.nextLine();
      accNo = sc.nextLine();
      acc = findAccount(accounts, accNo);
      if (acc != null) {
         System.out.print("Enter withdrawal amount: ");
         double amount = sc.nextDouble();
         if (acc instanceof SavAcct) {
            ((SavAcct) acc).withdraw(amount);
         } else if (acc instanceof CurAcct) {
            ((CurAcct) acc).withdraw(amount);
```

```
            }
        } else {
            System.out.println("Account not found.");
        }
        break;

    case 3: // Compute Interest
        System.out.print("Enter account number: ");
        sc.nextLine();
        accNo = sc.nextLine();
        acc = findAccount(accounts, accNo);
        if (acc != null && acc instanceof SavAcct) {
            System.out.print("Enter number of years: ");
            int years = sc.nextInt();
            ((SavAcct) acc).computeAndDepositInterest(years);
        } else if (acc != null) {
            System.out.println("Interest calculation is not applicable for Current accounts.");
        } else {
            System.out.println("Account not found.");
        }
        break;

    case 4: // Display Balance
        for (Account account : accounts) {
            account.displayBalance();
        }
        break;

    case 5: // Exit
        System.out.println("Exiting program.");
        break;
```

```
                default:
                    System.out.println("Invalid choice.");
            }
        } while (choice != 5);

        sc.close();
    }

    private static Account findAccount(ArrayList<Account> accounts, String accountNumber) {
        for (Account account : accounts) {
            if (account.accountNumber.equals(accountNumber)) {
                return account;
            }
        }
        return null;
    }
}
```

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Shreyas\OneDrive\ドキュメント\Desktop\java lab>javac Bank.java

C:\Users\Shreyas\OneDrive\ドキュメント\Desktop\java lab>java Bank.java
Enter the number of users: 2

Enter details for user 1:
Enter customer name: shreyas
Enter account number: 789456
Enter initial balance: 10000
Enter account type (1 for Savings, 2 for Current): 1

Enter details for user 2:
Enter customer name: suhas
Enter account number: 654987
Enter initial balance: 20000
Enter account type (1 for Savings, 2 for Current): 2

=== Bank Operations Menu ===
1. Deposit
2. Withdraw
3. Compute Interest (Savings only)
4. Display Balance
5. Exit
Enter your choice: 1
Enter account number: 789456
Enter deposit amount: 50000
Deposit successful. Updated balance: 60000.0

=== Bank Operations Menu ===
1. Deposit
2. Withdraw
3. Compute Interest (Savings only)
4. Display Balance
5. Exit
Enter your choice: 2
Enter account number: 654987
Enter withdrawal amount: 1000
Withdrawal successful. Updated balance: 19000.0
```

```
=== Bank Operations Menu ===
1. Deposit
2. Withdraw
3. Compute Interest (Savings only)
4. Display Balance
5. Exit
Enter your choice: 3
Enter account number: 789456
Enter number of years: 9
Interest of 25398.7087452891 has been added. Updated balance: 85398.7087452891

=== Bank Operations Menu ===
1. Deposit
2. Withdraw
3. Compute Interest (Savings only)
4. Display Balance
5. Exit
Enter your choice: 4
Account Holder: shreyas, Account Number: 789456, Current Balance: 85398.7087452891
Account Holder: suhas, Account Number: 654987, Current Balance: 19000.0

=== Bank Operations Menu ===
1. Deposit
2. Withdraw
3. Compute Interest (Savings only)
4. Display Balance
5. Exit
Enter your choice: 5
Exiting program.

C:\Users\Shreyas\OneDrive\ドキュメント\Desktop\java lab>
```

```
public class Student {

    protected String usn;
    protected String name;
    protected String sem;

    public Student(String usn, String name, String sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }


    public void displayDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}

public class Internals extends Student {

    protected int[] internalMarks = new int[5];

    public Internals(String usn, String name, String sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }


    public void displayInternalMarks() {
        System.out.println("Internal Marks:");
        for (int i = 0; i < internalMarks.length; i++) {
            System.out.println("Course " + (i + 1) + ": " + internalMarks[i]);
        }
    }
}
```

```
package SEE;

import CIE.Internals;

public class External extends Internals {
 int[] externalMarks = new int[5];

    public External(String usn, String name, String sem, int[] internalMarks, int[]
externalMarks) {
        super(usn, name, sem, internalMarks);
        this.externalMarks = externalMarks;
    }

     public void displayExternalMarks() {
        System.out.println("External Marks:");
        for (int i = 0; i < externalMarks.length; i++) {
            System.out.println("Course " + (i + 1) + ": " + externalMarks[i]);
        }
    }



    public void displayFinalMarks() {
        System.out.println("Final Marks (Internal + External):");
        for (int i = 0; i < 5; i++) {
            int finalMarks = internalMarks[i] + externalMarks[i];
            System.out.println("Course " + (i + 1) + ": " + finalMarks);
        }
    }
}



import CIE.Internals;
import SEE.External;
import java.util.Scanner;

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

              System.out.print("Enter number of students: ");
```

```
int n = scanner.nextInt();
scanner.nextLine(); // Consume the newline character

        External[] students = new External[n];

    for (int i = 0; i < n; i++) {
    System.out.println("\nEnter details for student " + (i + 1));

    System.out.print("Enter USN: ");
    String usn = scanner.nextLine();

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

    System.out.print("Enter Semester: ");
    String sem = scanner.nextLine();

    int[] internalMarks = new int[5];
    System.out.println("Enter internal marks for 5 courses:");
    for (int j = 0; j < 5; j++) {
        System.out.print("Course " + (j + 1) + ": ");
        internalMarks[j] = scanner.nextInt();
    }

    int[] externalMarks = new int[5];
    System.out.println("Enter external marks for 5 courses:");
    for (int j = 0; j < 5; j++) {
        System.out.print("Course " + (j + 1) + ": ");
        externalMarks[j] = scanner.nextInt();
    }
    scanner.nextLine();

    students[i] = new External(usn, name, sem, internalMarks, externalMarks);
}

System.out.println("\nStudent Marks Information:");
for (int i = 0; i < n; i++) {
    students[i].displayDetails();
    students[i].displayInternalMarks();
    students[i].displayExternalMarks();
```

```
            students[i].displayFinalMarks();
            System.out.println();
        }

        scanner.close();
    }
}
```

```
Enter number of students: 2

Enter details for student 1
Enter USN: 001
Enter Name: Alice
Enter Semester: 5
Enter internal marks for 5 courses:
Course 1: 18
Course 2: 20
Course 3: 15
Course 4: 17
Course 5: 19
Enter external marks for 5 courses:
Course 1: 40
Course 2: 45
Course 3: 38
Course 4: 42
Course 5: 44

Enter details for student 2
Enter USN: 002
Enter Name: Bob
Enter Semester: 5
Enter internal marks for 5 courses:
Course 1: 19
Course 2: 17
Course 3: 20
Course 4: 16
Course 5: 18
Enter external marks for 5 courses:
Course 1: 36
Course 2: 40
Course 3: 39
Course 4: 41
Course 5: 43
```

```
Student Marks Information:
----------------------------------
Student 1:
USN: 001
Name: Alice
Semester: 5
Internal Marks: [18, 20, 15, 17, 19]
External Marks: [40, 45, 38, 42, 44]
Final Marks: [58, 65, 53, 59, 63]

----------------------------------
Student 2:
USN: 002
Name: Bob
Semester: 5
Internal Marks: [19, 17, 20, 16, 18]
External Marks: [36, 40, 39, 41, 43]
Final Marks: [55, 57, 59, 57, 61]

C:\Users\User\>
```