

## LAB PROGRAM 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 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 the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;

class Account {

    String customerName;

    int accountNumber;

    String accountType;

    double balance;

    Account(String name, int accNo, String accType, double initialBalance) {

        customerName = name;

        accountNumber = accNo;

        accountType = accType;

        balance = initialBalance;

    }

    void deposit(double amount) {

        balance += amount;

        System.out.println("Deposit of $" + amount + " successful.");

    }

}
```

```
void displayBalance() {  
    System.out.println("Account Balance: $" + balance);  
}  
}
```

```
class SavAcct extends Account {  
    double interestRate;
```

```
SavAcct(String name, int accNo, String accType, double initialBalance, double interest) {  
    super(name, accNo, accType, initialBalance);  
    interestRate = interest;  
}
```

```
void computeInterest() {  
    double interestAmount = balance * (interestRate / 100);  
    deposit(interestAmount);  
}
```

```
void withdraw(double amount) {  
    if (balance >= amount) {  
        balance -= amount;  
        System.out.println("Withdrawal of $" + amount + " successful.");  
    } else {  
        System.out.println("Insufficient funds.");  
    }  
}  
}
```

```
class CurAcct extends Account {
```

```
double minimumBalance;
```

```
double serviceCharge;
```

```
CurAcct(String name, int accNo, String accType, double initialBalance, double  
minBalance, double charge) {
```

```
    super(name, accNo, accType, initialBalance);
```

```
    minimumBalance = minBalance;
```

```
    serviceCharge = charge;
```

```
}
```

```
void checkMinimumBalance() {
```

```
    if (balance < minimumBalance) {
```

```
        balance -= serviceCharge;
```

```
        System.out.println("Service charge of $" + serviceCharge + " applied due to balance  
below minimum.");
```

```
    }
```

```
}
```

```
void withdraw(double amount) {
```

```
    if (balance >= amount) {
```

```
        balance -= amount;
```

```
        System.out.println("Withdrawal of $" + amount + " successful.");
```

```
        checkMinimumBalance();
```

```
    } else {
```

```
        System.out.println("Insufficient funds.");
```

```
    }
```

```
}
```

```
}
```

```
public class Bank {
```

```
    public static void main(String[] args) {
```

```

Scanner scanner = new Scanner(System.in);

// Create a savings account
System.out.println("Enter name for savings account:");
String savName = scanner.nextLine();
System.out.println("Enter initial balance for savings account:");
double savBalance = scanner.nextDouble();
SavAcct savingsAccount = new SavAcct(savName, 1001, "Savings", savBalance, 5.0);

// Create a current account
System.out.println("Enter name for current account:");
scanner.nextLine(); // Consume newline
String curName = scanner.nextLine();
System.out.println("Enter initial balance for current account:");
double curBalance = scanner.nextDouble();
CurAcct currentAccount = new CurAcct(curName, 2001, "Current", curBalance, 1000.0,
20.0);

// Perform operations on savings account
System.out.println("\nOperations for Savings Account:");
savingsAccount.displayBalance();
System.out.println("Enter amount to deposit in savings account:");
double savDeposit = scanner.nextDouble();
savingsAccount.deposit(savDeposit);
savingsAccount.displayBalance();
savingsAccount.computeInterest();
savingsAccount.displayBalance();
System.out.println("Enter amount to withdraw from savings account:");
double savWithdraw = scanner.nextDouble();
savingsAccount.withdraw(savWithdraw);
savingsAccount.displayBalance();

```

```
// Perform operations on current account
System.out.println("\nOperations for Current Account:");
currentAccount.displayBalance();
System.out.println("Enter amount to deposit in current account:");
double curDeposit = scanner.nextDouble();
currentAccount.deposit(curDeposit);
currentAccount.displayBalance();
System.out.println("Enter amount to withdraw from current account:");
double curWithdraw = scanner.nextDouble();
currentAccount.withdraw(curWithdraw);
currentAccount.displayBalance();

scanner.close();
}
}
```

OUTPUT:

```
C:\Users\Admin\Desktop\CS_140>javac Bank.java

C:\Users\Admin\Desktop\CS_140>java Bank
Enter name for savings account:
Madhu
Enter initial balance for savings account:
5000
Enter name for current account:
Sarika
Enter initial balance for current account:
6000

Operations for Savings Account:
Account Balance: $5000.0
Enter amount to deposit in savings account:
2000
Deposit of $2000.0 successful.
Account Balance: $7000.0
Deposit of $350.0 successful.
Account Balance: $7350.0
Enter amount to withdraw from savings account:
4000
Withdrawal of $4000.0 successful.
Account Balance: $3350.0

Operations for Current Account:
Account Balance: $6000.0
Enter amount to deposit in current account:
3000
Deposit of $3000.0 successful.
Account Balance: $9000.0
Enter amount to withdraw from current account:
8000
Withdrawal of $8000.0 successful.
Account Balance: $1000.0

C:\Users\Admin\Desktop\CS_140>
```