Java Day-3 Assignment

1. BankOperations Interface

```
package BankCaseStudy;

public interface BankOperations {
        void deposit(double amount);
   void withdraw(double amount);
   void transfer(Account target, double amount);
   double checkBalance();
   void showTransactionHistory();

}
```

2. Account Class

```
 package BankCaseStudy;

 import java.util.ArrayList;
 import java.util.List;

 public abstract class Account implements BankOperations {
            protected String accountNumber;
            protected double balance;
            protected List<String> transactionHistory = new ArrayList<>();

            public abstract void deposit(double amount);
            public abstract void withdraw(double amount);

            public void transfer(Account target, double amount) {
               this.withdraw(amount);
               target.deposit(amount);
               this.addTransaction("Transferred to Account " +
 target.accountNumber + ": " + amount);
                target.addTransaction("Received from Account " +
 this.accountNumber + ": " + amount);
            }
```

```
    public double checkBalance() {
        return balance;
    }
    protected void addTransaction(String info) {
        transactionHistory.add(info);
    }

    public void showTransactionHistory() {
        for (String tx : transactionHistory) {
            System.out.println("- " + tx);
        }
    }

}
```

3. SavingsAccount class

```
package BankCaseStudy;

    public class SavingsAccount extends Account{
      private static final double MIN_BALANCE = 1000.0;

   public SavingsAccount(String accountNumber, double initialBalance) {
      this.accountNumber = accountNumber;
      this.balance = initialBalance;
   }
   public void deposit(double amount) {
      balance += amount;
      addTransaction("Deposited: " + amount);
   }
   public void withdraw(double amount) {
      if (balance - amount >= MIN_BALANCE) {
         balance -= amount;
         addTransaction("Withdrawn: " + amount);
      } else {
         System.out.println("Insufficient balance. Minimum ₹1000 required.");
      }
   }

}
```

4. CurrentAccount Class

```
package BankCaseStudy;

public class CurrentAccount extends Account{
        private static final double OVERDRAFT_LIMIT = 1000.0;

    public CurrentAccount(String accountNumber, double initialBalance) {
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }

    public void deposit(double amount) {
        balance += amount;
        addTransaction("Deposited: " + amount);
    }

    public void withdraw(double amount) {
        if (balance - amount >= -OVERDRAFT_LIMIT) {
            balance -= amount;
            addTransaction("Withdrawn: " + amount);
        } else {
            System.out.println("Overdraft limit exceeded.");
        }
    }

}
```

5. Customer Class

```
package BankCaseStudy;

 import java.util.ArrayList;
 import java.util.List;

   public class Customer {
   private String customerId;
    private String name;
    private List<Account> accounts = new ArrayList<>();

       public Customer(String customerId, String name) {
        this.customerId = customerId;
        this.name = name;
System.out.println("Customer Created: " + name + " [Customer ID: " +
customerId + "]");
    }

    public void addAccount(Account acc) {
        accounts.add(acc);
    }

    public List<Account> getAccounts() {
        return accounts;
    }

    public String getCustomerId() {
        return customerId;
    }
    public String getName() {
        return name;
    }

}
```

6. BankBranch Class

```
package BankCaseStudy;

import java.util.ArrayList;
import java.util.List;

public class BankBranch {
private String branchId;
private String branchName;
private List<Customer> customers = new ArrayList<>();

    public BankBranch(String branchId, String branchName) {
        this.branchId = branchId;
        this.branchName = branchName;
        System.out.println("Branch Created: " + branchName + " [Branch ID:
" + branchId + "]");
    }

    public void addCustomer(Customer c) {
        customers.add(c);
        System.out.println("Customer added to branch.");
    }

    public Customer findCustomerById(String id) {
        for (Customer c : customers) {
            if (c.getCustomerId().equals(id)) {
                return c;
            }
        }
        return null;
    }

    public void listAllCustomers() {
    for (Customer c : customers) {
     System.out.println("Customer: " + c.getName() + " [ID: " +
c.getCustomerId() + "]");
        }
    }

}
```

7. Main Class

```
package BankCaseStudy;

public class Main {

        public static void main(String[] args) {
                BankBranch branch = new BankBranch("B001", "Main Branch");


        Customer alice = new Customer("C001", "Alice");
        branch.addCustomer(alice);


        SavingsAccount sAccount = new SavingsAccount("S001", 5000.0);
        alice.addAccount(sAccount);
        System.out.println("Savings Account [S001] opened with initial balance:
5000.0");

        CurrentAccount cAccount = new CurrentAccount("C001", 2000.0);
        alice.addAccount(cAccount);
        System.out.println("Current Account [C001] opened with initial balance:
2000.0 and overdraft limit 1000.0");

        sAccount.deposit(2000.0);
        System.out.println("Deposited 2000.0 to Savings Account [S001]");
        System.out.println("Current Balance: " + sAccount.checkBalance());

        cAccount.withdraw(2500.0);
        System.out.println("Withdrawn 2500.0 from Current Account [C001]");
        System.out.println("Current Balance: " + cAccount.checkBalance() + "
(Using Overdraft)");

        sAccount.transfer(cAccount, 1000.0);
        System.out.println("Transferred 1000.0 from Savings Account [S001] to
Current Account [C001]");
        System.out.println("Savings Balance: " + sAccount.checkBalance());
        System.out.println("Current Balance: " + cAccount.checkBalance());


        System.out.println("\nTransaction History:");
```

```
        System.out.println("Account: S001");
        sAccount.showTransactionHistory();

        System.out.println("\nAccount: C001");
        cAccount.showTransactionHistory();


          }

}
```