

Good early morning Sir Gayo time check 4:am right now.File 1: CreditCard.java

```
1 public class CreditCard {
2     private String customerName;
3     private String bankName;
4     private String accountNumber;
5     private double creditLimit;
6     private double balance;
7
8     // Constructor
9     public CreditCard(String customerName, String bankName, String accountNumber, double creditLimit, double balance) {
10         this.customerName = customerName;
11         this.bankName = bankName;
12         this.accountNumber = accountNumber;
13         this.creditLimit = creditLimit;
14         this.balance = balance;
15     }
16
17     // Getter and Setter methods
18     public String getCustomerName() {
19         return customerName;
20     }
21
22     public String getBankName() {
23         return bankName;
24     }
```

```
25
26     public String getAccountNumber() {
27         return accountNumber;
28     }
29
30     public double getCreditLimit() {
31         return creditLimit;
32     }
33
34     public double getBalance() {
35         return balance;
36     }
37
38     public void charge(double amount) {
39         if (balance + amount <= creditLimit) {
40             balance += amount;
41         } else {
42             System.out.println("Charge exceeds credit limit!");
43         }
44     }
```

```
J CreditCard.java 1 x  J CreditCardTest.java 1
C: > Users > CCV YOUTH > OneDrive > Desktop > CreditCard > J CreditCard.java > ...
1  public class CreditCard {
44 }
45
46 public void payment(double amount) {
47     if (balance - amount >= 0) {
48         balance -= amount;
49     } else {
50         System.out.println(x:"Payment exceeds balance!");
51     }
52 }
53
54 // Display Card Info
55 public void displayCardInfo() {
56     System.out.println("Customer = " + customerName);
57     System.out.println("Bank = " + bankName);
58     System.out.println("Account = " + accountNumber);
59     System.out.println("Balance = " + balance);
60     System.out.println("Limit = " + creditLimit);
61 }
62 }
63
```

This file defines a class called `CreditCard`, which represents a credit card. The class has several attributes and methods that describe and manipulate credit card information.

Key components of `CreditCard.java`:

1. Attributes (Variables):

- `customerName`: The name of the cardholder.
- `bankName`: The name of the bank that issued the card.
- `accountNumber`: The unique number assigned to the cardholder's account.
- `creditLimit`: The maximum amount that can be charged to the card.
- `balance`: The current balance of the card (how much is owed).

2. Constructor:

- This is used to create a new `CreditCard` object with the cardholder's name, bank name, account number, credit limit, and initial balance.

3. Methods:

- `charge(amount)`: Adds a specified amount to the balance (this simulates making a purchase with the credit card).
- `payment(amount)`: Reduces the balance by a specified amount (this simulates making a payment to pay off part of the balance).
- `displayCardInfo()`: Displays the cardholder's details, including name, bank, account number, balance, and credit limit.

File 2: CreditCardTest.java

```
J CreditCard.java 1  J CreditCardTest.java 1 X
C:\Users\CCV YOUTH\OneDrive\Desktop\CreditCard> J CreditCardTest.java & CreditCardTest > main(String[])

3  public class CreditCardTest {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6
7          System.out.print(s:"Enter the number of credit cards: ");
8          int numCards = scanner.nextInt();
9          scanner.nextLine(); // Consume the newline character
10
11         CreditCard[] cards = new CreditCard[numCards];
12
13         // Input credit card details
14         for (int i = 0; i < numCards; i++) {
15             System.out.println("\nCard " + (i + 1) + ":");
16             System.out.print(s:"Customer name: ");
17             String customerName = scanner.nextLine();
18             System.out.print(s:"Bank name: ");
19             String bankName = scanner.nextLine();
20             System.out.print(s:"Account number: ");
21             String accountNumber = scanner.nextLine();
22             System.out.print(s:"Credit limit: ");
23             double creditLimit = scanner.nextDouble();
24             System.out.print(s:"Initial balance: ");
25             double balance = scanner.nextDouble();
26             scanner.nextLine(); // Consume the newline character
27
28             cards[i] = new CreditCard(customerName, bankName, accountNumber, creditLimit, balance);
29         }
30         // Display and perform operations on the credit cards
31         for (CreditCard card : cards) {
32             card.displayCardInfo();
33             card.charge(amount:100); // Simulate a charge of 100
34             System.out.println("New balance = " + card.getBalance());
35             card.payment(amount:200); // Simulate a payment of 200
36             System.out.println("New balance = " + card.getBalance());
37             card.payment(amount:100); // Another payment of 100
38             System.out.println("New balance = " + card.getBalance());
39         }
40         scanner.close();
41     }
42 }
```

This is the main file that runs the program. It interacts with the user to get information about

multiple credit cards, performs operations on those cards (charging and making payments), and displays the updated information.

Key components of `CreditCardTest.java`:

1. Input from the user:

- The program asks the user to enter the number of credit cards they want to simulate.
- For each card, the user is prompted to enter the customer's name, bank name, account number, credit limit, and initial balance.

2. Card Operations:

- For each card, the program performs a series of actions:
 - It displays the initial information (balance, credit limit, etc.).
 - It "charges" \$100 to the card (simulating a purchase).
 - It then makes a payment of \$200 to the card.
 - Finally, it makes another payment of \$100 to the card.

3. Displaying Results:

- After each action (charge or payment), the updated balance of the card is shown.

The output

```
Enter the number of credit cards:
```

```
Card 1:
```

```
Customer name: MARK
```

```
Bank name: PDO
```

```
Account number: 2413 3142 3143 1413
```

```
Credit limit: 5000
```

```
Initial balance: 2000
```

```
Card 2:
```

```
Customer name: JESUS CHRIST
```

```
Bank name: HEAVEN
```

```
Account number: 1234 1234 1234 1234
```

```
Credit limit: 9999
```

```
Initial balance: 8888
```

```
Card 3:
```

```
Customer name: mama
```

```
Bank name: NBI
```

```
Account number: 1531
```

```
Credit limit: 3000
```

```
Initial balance: 533
```

```
Customer = MARK
```

```
Bank = PDO
```

```
Account = 2413 3142 3143 1413
```

```
Balance = 2000.0
```

```
Limit = 5000.0
```

```
New balance = 2100.0
```

```
New balance = 1900.0
```

```
New balance = 1800.0
```

```
Customer = JESUS CHRIST
```

```
Bank = HEAVEN
```

```
Account = 1234 1234 1234 1234
```

```
Balance = 8888.0
```

```
Limit = 9999.0
```

```
New balance = 8988.0
```

```
New balance = 8788.0
```

```
New balance = 8688.0
```

```
New balance = 8988.0
New balance = 8788.0
New balance = 8688.0
Customer = mama
Bank = NBI
Account = 1531
Account = 1531
Balance = 533.0
Limit = 3000.0
New balance = 633.0
New balance = 433.0
New balance = 333.0
PS C:\Users\CCV YOUTH>
```

How It Works Together:

1. The `CreditCardTest.java` file creates multiple `CreditCard` objects, each representing a credit card with specific details (such as the cardholder's name, bank, etc.).
2. For each card, the program performs charges and payments, updating the balance.
3. After each operation (charge or payment), it shows the new balance of the card.

Summary in Simple Terms:

1. **Card 1 (MARK):**
 - Starts with a balance of \$2000.
 - A charge of \$100 is added, making the balance \$2100.
 - A \$200 payment reduces it to \$1900.
 - Another \$100 payment reduces it to \$1800.
2. **Card 2 (JESUS CHRIST):**
 - Starts with a balance of \$8888.
 - A charge of \$100 is added, making the balance \$8988.
 - A \$200 payment reduces it to \$8788.
 - Another \$100 payment reduces it to \$8688.
3. **Card 3 (mama):**
 - Starts with a balance of \$533.
 - A charge of \$100 is added, making the balance \$633.

- A \$200 payment reduces it to \$433.
- Another \$100 payment reduces it to \$333.

In each case, the program simulates adding charges and making payments, showing how the balance changes after each transaction.