Basic Banking System :

Create simple banking features like deposit, withdrawal, and balance check.

A simple banking simulation where users can deposit and withdraw money, check their balance, and view a history of their transactions. It can manage basic financial operations for a single user account.

Input: User commands to deposit, withdraw, or check balance.
Output: Updated balance after transactions.

Example:

- Input: Deposit 500
- Output: "Balance: 500"
- Input: Withdraw 200
- Output: "Balance: 300"

Here are two different solutions for the "Basic Banking System" project in Java:

Solution 1: Simple Banking System using basic methods

Code:

```java
// Solution 1: Simple Banking System using basic methods
import java.util.Scanner;

public class BasicBankingSystem {
    private double balance;  // To store the balance of the user

    // Constructor to initialize balance
    public BasicBankingSystem() {
        balance = 0;
    }

    // Method to deposit money
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

    // Method to withdraw money
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: " + amount);
        } else {
            System.out.println("Invalid or insufficient funds.");
        }
    }

    // Method to check balance
    public void checkBalance() {
        System.out.println("Current Balance: " + balance);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        BasicBankingSystem bank = new BasicBankingSystem();
        boolean exit = false;

        // Main loop for user input
        while (!exit) {
            System.out.println("Enter command (1: Deposit, 2: Withdraw, 3:
            int command = scanner.nextInt();

            switch (command) {
                case 1:
                    System.out.println("Enter deposit amount:");
                    double depositAmount = scanner.nextDouble();
                    bank.deposit(depositAmount);
                    break;
                case 2:
                    System.out.println("Enter withdraw amount:");
                    double withdrawAmount = scanner.nextDouble();
                    bank.withdraw(withdrawAmount);
                    break;
                case 3:
                    bank.checkBalance();
                    break;
                case 4:
                    exit = true;
                    System.out.println("Exiting the banking system.");
                    break;
                default:
                    System.out.println("Invalid command.");
                    break;
            }
        }
        scanner.close();
    }
}
```

Output:

```
Enter command (1: Deposit, 2: Withdraw, 3: Check Balance, 4: Exit):
 1
Enter deposit amount:
 2000
Deposited: 2000.0
Enter command (1: Deposit, 2: Withdraw, 3: Check Balance, 4: Exit):
 2
Enter withdraw amount:
 545
Withdrew: 545.0
Enter command (1: Deposit, 2: Withdraw, 3: Check Balance, 4: Exit):
 3
Current Balance: 1455.0
Enter command (1: Deposit, 2: Withdraw, 3: Check Balance, 4: Exit):
 4
Exiting the banking system.
```

Explanation :

- **Initialization:** The system starts with a balance initialized to 0.
- **Deposit method:** Allows the user to deposit an amount into their account if it's positive.
- **Withdraw method:** Allows the user to withdraw money, provided they have sufficient funds.
- Check **Balance:** Users can check their current balance at any time.
- **Main Loop:** Continually accepts user input to deposit, withdraw, or check balance, until the user exits.

Solution 2: Banking System with Transaction History

Code:

```
// Solution 2: Banking System with transaction history using ArrayList

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

public class BankingSystemWithHistory {
    private double balance;  // To store the balance of the user
    private ArrayList<String> transactions;  // To store transaction histor

    // Constructor to initialize balance and transaction history
    public BankingSystemWithHistory() {
        balance = 0;
        transactions = new ArrayList<>();
    }

    // Method to deposit money
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            transactions.add("Deposited: " + amount);
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

    // Method to withdraw money
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            transactions.add("Withdrew: " + amount);
            System.out.println("Withdrew: " + amount);
        } else {
            System.out.println("Invalid or insufficient funds.");
        }
    }
    // Method to check balance
    public void checkBalance() {
        System.out.println("Current Balance: " + balance);
    }

    // Method to print transaction history
    public void printTransactionHistory() {
        System.out.println("Transaction History:");
        for (String transaction : transactions) {
            System.out.println(transaction);
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        BankingSystemWithHistory bank = new BankingSystemWithHistory();
        boolean exit = false;

        // Main loop for user input
        while (!exit) {
            System.out.println("Enter command (1: Deposit, 2: Withdraw, 3:
            int command = scanner.nextInt();

            switch (command) {
                case 1:
                    System.out.println("Enter deposit amount:");
                    double depositAmount = scanner.nextDouble();
                    bank.deposit(depositAmount);
                    break;
                case 2:
                    System.out.println("Enter withdraw amount:");
                    double withdrawAmount = scanner.nextDouble();
                    bank.withdraw(withdrawAmount);
                    break;
                case 3:
                    bank.checkBalance();
                    break;
                case 4:
                    bank.printTransactionHistory();
                    break;
                case 5:
                    exit = true;
                    System.out.println("Exiting the banking system.");
                    break;
                default:
                    System.out.println("Invalid command.");
                    break;
            }
        }
        scanner.close();
    }
}
```

Output:

```
 Enter command (1: Deposit, 2: Withdraw, 3: Check Balance, 4: Transaction Histor
 1
Enter deposit amount:
 5000
Deposited: 5000.0
Enter command (1: Deposit, 2: Withdraw, 3: Check Balance, 4: Transaction History
 2
Enter withdraw amount:
 450.55
Withdrew: 450.55
Enter command (1: Deposit, 2: Withdraw, 3: Check Balance, 4: Transaction History
 3
Current Balance: 4549.45
Enter command (1: Deposit, 2: Withdraw, 3: Check Balance, 4: Transaction History
 5
Exiting the banking system.
```

Explanation: