Exercise 1: Simple Banking System (No OOP)

Create a Java program that simulates a simplified banking system for a single user. It should provide the following functionality:

1. **Initial Balance:** The program should start by prompting the user to enter their initial account balance. Store this balance in a variable.
2. **Transaction Menu:** Display a menu with the following options:
    - **Deposit:** Allow the user to enter a deposit amount, add it to the balance, and print the updated balance.
    - **Withdraw:** Allow the user to enter a withdrawal amount. If the withdrawal amount is less than or equal to the current balance, subtract it from the balance and print the updated balance. Otherwise, print an "Insufficient funds" message.
    - **Transaction History:** Store the last 5 transactions (deposit or withdrawal amounts) in an array. This option should print the last 5 transactions. If fewer than 5 transactions have occurred, print only the available transactions. Use a separate variable to keep track of the current number of transactions recorded. You might want to use another variable to act like a circular index so that the oldest transaction is overwritten when the array is full.
    - **Exit:** Terminate the program.
3. **Input Validation:** Ensure that the user enters valid input (e.g., positive numbers for deposit and withdrawal amounts). If the input is invalid, print an error message and prompt the user to enter the input again.
4. **Loop:** The program should continue to display the menu and process user input until the user chooses the "Exit" option.

Example Interaction:

```
Enter initial balance: 1000

Banking Menu:
1. Deposit
2. Withdraw
3. Transaction History
4. Exit

Enter your choice: 1
Enter deposit amount: 500
Updated balance: 1500

Banking Menu:
1. Deposit
2. Withdraw
3. Transaction History
4. Exit

Enter your choice: 2
Enter withdrawal amount: 200
Updated balance: 1300

Banking Menu:
1. Deposit
2. Withdraw
3. Transaction History
4. Exit

Enter your choice: 3
Transaction History:
+500
-200
```

Tips and Hints:

- Use a Scanner to get user input.
- Use do-while or while loops to repeat the menu.
- Use if-else statements to handle different menu options and input validation.
- Use an array and a counter variable to store and manage the transaction history. You might also want to use another variable to keep track of the index at which the next transaction should be recorded in the transaction history array.