Case Study : Developing a Banking Application for Fund Transfers

Problem Statement :

A bank is developing an application to enable users to transfer funds between accounts. The program must handle exceptions to prevent errors such as transferring more money than the available balance or entering an invalid account number.

Solution Approach :

1. Custom Exception Handling:

- **Insufficient Funds**: Raise an exception if the transfer amount exceeds the available balance.
- **Invalid Account Number**: Handle invalid account formats by catching a ValueError.

2. Data Handling:

- Account information, including account numbers and balances, are fetched from a dummy file (accounts.txt).

Implementation :

Dummy Data File : accounts.txt

Python Script: banking_application.py

How the program works:

1. **Fetching Account Data**:
    - The fetch_account_data function reads the accounts.txt file and stores account information in a dictionary.
2. **Transferring Funds**:
    - The transfer_funds function performs the transfer. It checks:
        - If the provided account numbers are valid.
        - If the sender has sufficient funds.
    - If any condition is not met, it raises a ValueError with an appropriate error message.
3. **Exception Handling**:

- The try block attempts to transfer funds.
- The except block catches any ValueError and prints the error message.

Conclusion :

This program addresses common banking errors by fetching account data from a file and handling fund transfers with custom error messages. It ensures valid transactions, prevents overdrafts, and checks for valid account numbers, making it a robust solution for a banking application. Simple, efficient, and ready for real-world use.