

Bank

Description

Build an object-oriented program to process a list of bank transactions. Transactions include creating accounts, making deposits and withdrawals, adding accumulated interest, and deducting accumulated fees.

The following rules apply:

- A bounced check costs \$25 and is deducted immediately from the checking account balance.
- There is a fee for each check (checking account withdrawal). Each checking account may use a different fee. The fees are charged as part of an update (see below).
- Savings accounts accrue interest. Each savings account may have a different interest rate. Interest is credited as part of an update (see below).
- Interest is charged for loans. Each loan may have a different interest rate. Interest is charged as part of an update (see below).
- An attempt is made to withdraw more money than is available in a savings or checking account shall be rejected and a message indicating such shall be output.
- There is no maximum limit on the amount loaned.

The transaction file uses the following symbols: s, c, l, t, and u.

- **s** - Create a new savings account. An account number, a name, an initial balance, and the interest rate are supplied.
- **c** - Create a new checking account. An account number, a name, an initial balance, and the fee for each processed check are supplied.
- **l** - Create a new loan account. An account number, a name, an initial balance, and the interest rate are supplied.
- **t** - Perform a transaction. An account number and the amount of the transaction are supplied.
 - For savings accounts, a positive value is a deposit and a negative value is a withdrawal.
 - For checking accounts, a positive value is a deposit and a negative value represents a check.
 - For loan accounts, a positive value represents a payment on the loan and a negative value represents the borrowing of additional money.
- **u** - perform an update. An account number is supplied.
 - For savings accounts, calculate and add interest earned.
 - For checking accounts, calculate and deduct the fee for each check.
 - For loan accounts, calculate and add the interest charged to the outstanding loan balance.

Use the list of transactions contained in bank.txt (posted with this write up) as input; there is no user input. **All transactions shall be processed in a single loop.** Assume that you wrote the program that generated the transaction file. Consequently, you know that all data are present and correct and that the formatting is consistent. You do not need to check input for errors. You do not need to parse each line of input into its constituent parts.

Create a base class named Account. Created derived classes named Savings, Checking, and Loan.

Overload the stream insertion operator (<<) and use it for output. Do not output from any class. The output from your program shall exactly match the output contained in bankOutput.txt (posted with this write up).

Output

See bankOutput.txt (posted with this write up).