Warmup and Review Lab
Simplified Loan Accounts

I strongly recommend that you work in pairs on this lab. If you have not used Netbeans and JUnit before, please ask someone with experience in those to work with you. They are required to say "yes" if you ask them and they are not already working with someone. Make sure that you copy all the files to your partner's account at the end of the session. You will each have to turn in this lab when it is completed next week.

Start Netbeans and create a new Java project called **Loan**. Create a class called **LoanAccounting** that contains only a **main** that will instantiate the objects and run everything.

You are to write a program that manages various kinds of loans. Keep in mind that this is an *academic* exercise – it's not intended to be extremely realistic. The main goals of the assignment are to give you some experience with inheritance and polymorphism. Be sure to follow the specifications exactly.

There are three kinds of loans:

1. Home Equity Loans
2. Regular Mortgage Loans
3. Debt Consolidation Loans

The loans have slightly different terms, rates, and conditions as described below.

Three files are involved in this program:

1. input file ***loans*** containing information about existing loans
2. input file ***transactions*** of new transactions, and
3. output file ***results*** of updated account objects.

The program must read in the entire ***loans*** file, storing all the information in objects created at run time. Then the program must read ***transactions***, one record at a time, updating accounts or creating new ones. After all ***transactions*** have been read and accounts updated, the output ***results*** file is created and the program terminates. For this program, you may assume that the data files each contain no more than 100 records, and that all data in the input files are valid (you do not have to do any data validation in this assignment). You can find sample **loans** and **transactions** files at **`/home/jcaristi/158`** (these files do NOT cover all possible cases – they are just intended to verify for you the form of the data).

Regardless of the kind of loan, each uses **currentBalance, interest Rate,** and **userID** variables, and an **output** method**.** So the program must use a (parent) base class for a loan containing at least those variables and methods, and any others that are reasonably inheritable by the descendent classes. Assume that all loans involve monthly payments, and that the **interestRate** is already set to the monthly rate. Interest is always computed and added to the existing balance before any payments are made, but after any additional loan amounts are added to the current balance (if appropriate for the type of loan).

Home Equity Loans involve, in addition to **currentBalance, interestRate,** and **userID**, the equity value of the home and minimum monthly payment. Additional amounts can be borrowed, and any amount above the minimum can be paid in a month. Transaction records begin with a single character, 'E', followed by a space, followed by a 6-character ID, followed by a space, followed by a character that is either 'P' for a payment or 'C' for a charge, or 'N' for a new account. That is followed by a blank and then the amount of the payment or charge (as the case may be). New account records do not have a payment or charge, but have amounts following the 'N' (all separated by blanks) for the initial balance, interest rate, and equity value of the home. No negative numbers are used. The master ***loans*** file contains a line for each Home Equity Loan that begins with the single character 'E', followed by a space, followed by the 6-character ID, followed by a blank, followed by the current balance, followed by a blank, followed by the interest rate, followed by a blank, followed by the equity value of the home. The minimum monthly payment is computed as 1% of the current balance. If a payment amount made is below the minimum monthly payment, the transaction is handled normally, and a $50 fee is added to the current balance.

Mortgage Loans are similar to Home Equity Loans, except that additional amounts cannot be borrowed, and there is a fixed monthly payment. Transaction records begin with a single character, 'M', followed by a space, followed by the 6-character ID, followed by a space, followed by the single character 'P' for payment or 'N' for new account, followed by a blank. If the transaction is a payment, nothing else is on the line. If it's a new account, the blank is followed by the initial balance, interest rate, equity value of the home, and monthly payment (all separated by blanks). The master ***loans*** file record for a mortgage loan is just like that of a home equity loan, except that it begins with 'M' instead of 'E', and has the monthly payment following the equity value and a blank.

Debt Consolidation Loans are like Mortgage Loans, except that the interest is computed completely differently. When a new loan is made, it is for 60 months. The interest is computed by multiplying the interest rate by the borrowed amount and multiplying that by 60. That amount is then added on to the beginning balance. That amount then becomes the initial loan balance, and is divided by 60 to obtain the fixed monthly payment. For example, if $10,000 is borrowed, and the interest rate is 0.01 (1% per month), then 6,000 is added to 10,000 to obtain an initial loan balance of $16,000. The monthly payment is then $266.67 (always round in the bank's favor). Transaction records are just like those of mortgage loans, except that the first character is 'D' instead of 'M', and there is no equity value or monthly payment for new accounts. The master ***loans*** file record begins with 'D', followed by a blank, followed by the ID, a blank, and the following (separated by blanks): current balance, interest rate, and monthly payment.

Your program must at the very least have the base class for loans and three subclasses for the various kinds of loans, and it must use polymorphism as much as is reasonable.