SWE 619 Assignment 1
Fall 2019

Goals:

- Getting started on Piazza.
- Basic familiarity with Java; Introduction to Contract Model.

There are two parts to this assignment.

1. Post a brief intro about yourself on the course Piazza page. For *any* credit, the posting must:
   - be a follow-up to my introduction. In other words, all intros need to be in the same thread.
   - Include a photo appropriate in size, content, and orientation.
2. Consider a method that calculates the number of months needed to pay off a loan of a given size at a fixed *annual* interest rate and a fixed *monthly* payment. For instance, a $100,000 loan at an 8% annual rate would take 166 months to discharge at a monthly payment of $1,000, and 141 months to discharge at a montly payment of $1,100. (In both of these cases, the final payment is smaller than the others; I rounded 165.34 up to 166 and 140.20 up to 141.) Continuing the example, the loan would never be paid off at a monthly payment of $100, since the principal would grow rather than shrink.
   Define a Java class called `Loan`. In that class, write a method that satisfies the following specification:

   ```
   public static int months (int principal, double rate, int payment)

   Requires: principal, rate, and payment all positive
             and payment is sufficiently large to drive the principal to zero.
   Effects:  return the number of months required to pay off the principal
   ```

   Note that the precondition is quite strong, which makes implementing the method easy. We will discuss specific methods of addressing this precondition in coming weeks. You should use double precision arithmetic internally, but the final result is an integer, not a floating point value. The key step in your calculation is to change the principal on each iteration with the following formula (which amounts to monthly compounding):

   ```
   newPrincipal = oldPrincipal * (1 + monthlyInterestRate) - payment;
   ```

   The variable names here are explanatory, not required. You may want to use different variables, which is fine.

   To make sure you understand the point about preconditions, your code is required to be minimal. Specifically, if it possible to delete parts of your implementation and still have it satisfy the requirements, you'll earn less than full credit.

Grading Criteria:

- Adherence to instructions. Do what I ask for above, not something else.

- Minimal implementation.
- Syntax: Java compiles and runs.