Due: Tuesday, October 23, 2012 at 10:50 AM

Learning objectives:
- to use while loops
- to use functions to modularize code and reuse code

Description:
Implement a modular solution for the dice game, **Pig**, with the user playing against the computer. Call the file `pig.py`. On each turn the current player repeatedly rolls a pair of dice and accumulates the total points of each roll. The object is to be the first to reach 100 points. If the player rolls a 1, all points for that turn are lost and the turn passes to the other player. However, if the player rolls two 1's ("snake eyes"), all points accumulated thus far in the game are lost, and the turn passes to the other player.

The human player may "give up" the dice voluntarily after any roll. So the player must decide whether to roll again (be a pig) and risk losing points, or give up the dice and risk having the other player win. The computer player gives up the dice after accumulating 20 or more points in a given turn, or rolling a 1.

The Pig games is described in Lewis and Loftus, *Java Software Solutions*, 6th edition, Addison-Wesley, 2009, p. 291.

Specifications:
The functions needed are:
`roll():` Simulates a roll of a single die, using `randint()` to generate an `int` between 1 and 6, inclusive.
`playerPlays(totalPlayerPoints)` → `totalPlayerPoints:` This function accepts the `totalPlayerPoints` before the turn begins. It "rolls" two dice, outputs their values, and depending on the roll either accumulates points for the turn or ends the turn. It displays the total points thus far for the turn after the roll of the dice. The function continues until a condition for stopping is met (see Description above). It returns the `totalPlayerPoints` after the turn is over.
`computerPlays(totalComputerPoints)` → `totalComputerPoints:` This function is similar to `playerPlays()` except it deals with the computer's total points and the condition for stopping is different (see Description above).
`playPig():` Controls one game. As long as neither player has scored 100 or more points it calls, in this order `playerPlays()` and `computerPlays()`.
`main():` Starts and ends the game, calls `playPig()`, and after the first game asks whether the user wants to play again.

Submission: Upload the `pig.py` file to your class account.
Policies: The policies given on Oaks are in effect for this and all assignments.