Lab Exercise 10

Software Design **Allen B. Downey**
Fall 2004

Due: never!!!

10.1 Threaders

1. Download and read

   ```
   wget http://wb/sd/code/Threader.py
   ```

   Now run it. When you press the Make Threader button, it creates a Threader (which is a kind of Turtle). The Threader executes `snowflake`, which runs for a while, drawing a Koch snowflake.

2. While the Threader is making the rounds, try pressing Run this code. Is the GUI responsive while `snowflake` is running? What does this tell you about the way the GUI is implemented in Tkinter?

3. While the Threader is hard at work, press the Quit button. Does the program quit right away? Why not? Try hitting Control-C while it is running.

4. While one Threader is hard at work, press the Make Threader button and see what happens. Can you explain what you see?

5. Change `make_threader` so that when it invokes `snowflake`, it runs in a new Thread. Now run more than one Threader at a time. See the difference? Do all the Threaders run at the same speed? Why or why not?

6. Unfortunately, you still can't quit while a Threader is running. To fix that:

   (a) Add an attribute named `running` to Threader, and initialize it to 1.

   (b) Change `koch` so that it checks `running` and quits if it is 0.

   (c) Create a new kind of TurtleWorld called ThreaderWorld. Override `quit` so that when the user presses quit, it loops through all the Threaders and sets their `running` attribute to 0.

   It ain't pretty, but unfortunately you can't really kill another thread; you have to ask it to kill itself.

7. Create a *lot* of Threaders. What happens? Why?