CMPT 103 – Lab #12

General Information

Python version and IDE:	Python 3.3 / Wing IDE 101
Allocated lab time:	2 hrs and 50 min
Due date:	At the end of the lab period
Lab weight:	3%

Topics

- ✓ Creating a class (object-oriented programming)

Submission

- ✓ **All code file (.py) should be submitted electronically** to your Lab Blackboard site.
- ✓ A portion of the total marks (20%) will be allocated for the programming style. For example, functions should be small; avoid writing duplicate code; names should be meaningful and descriptive; naming conventions should be followed consistently; code should be formatted properly; and comments should be accurate and well written.
- ✓ Comments are **required** for:
  - EACH program indicating the student name and program name.
  - EACH function indicating the function purpose, syntax (example usage of the function), parameters, and return value
  - Any block of code for which the purpose may be unclear (Note: you should always try to write clean code that can be understood easily without comments).

Assignment

For this lab, please put all functions into a file called `Lab12your_initials.py` (e.g., `Lab12FL.py` where F and L are the first letter of your first name and last name). Please feel free to write helper methods if necessary.

Design and implement a class `ComboLock` that works like the combination lock in a gym locker, as shown below. The lock is constructed with a combination – three numbers between 0 and 59. The `reset` method resets the dial so that it points to 0. The `turn_left` and `turn_right` methods turn the dial by a given number of ticks to the left or right. The `isopen` method attempts to open the lock. The lock opens if the user first turned it right to the first number in the combination, then left to the second, and then right to the third.

Here is a list of methods that you need to implement in the `ComboLock` class:

- A constructor that takes three integers between 0 and 59.
- The `__repr__` method that returns a representation of a `ComboLock` object.
- The `reset` method that sets the dial to point to 0.
- The `turn_left` method that takes an integer representing the number of ticks and that turns the dial to the left by the given number of ticks.
- The `turn_right` method that takes an integer representing the number of ticks and that turns the dial to the right by the given number of ticks.
- The `isopen` method that returns `True` if the lock opens, or `False` otherwise.

`Lab12.py` contains a test function that you can use to check your code. Do not modify the test function. If your code passes all of the test cases, you should see a message 'passed all test cases…' printed out. If the test function raises any `Error`, it means that there is something wrong with your code.

Acknowledgement: the assignment specification is adapted from "Java for Everyone" by C. S. Horstmann.