Chapter 4: Students should be able to:

- ☐ Import necessary classes from the PGL library (`GWindow, GRect`, etc.)
- ☐ Call and utilize common methods defined for a particular PGL object type (`move`, `get_x`, `get_width`, etc.).
- ☐ Create a `GWindow` object with the desired dimensions.
- ☐ Create `GRect`, `GOval` objects with the desired dimensions and placed at the desired location on the window.
- ☐ Control the color and fill of any `GFillableObject`.
- ☐ Create `GLabel` objects with a desired font and placed in a desired location on the screen.
- ☐ Decompose larger problems into smaller, simpler problems which can be tackled one at a time.

Chapter 5: Students should be able to:

- ☐ Define a syntactically correct simple function.
- ☐ Understand and describe the difference between a function definition and a function call.
- ☐ Understand and describe the difference between parameters and function arguments.
- ☐ Utilize `return` statements in correct places in their code to return the desired value(s) at the desired time (and not earlier!).
- ☐ Call a function utilizing keyword arguments.
- ☐ Define a function utilizing default values for a formal parameter.
- ☐ Identify what variables are defined within a particular scope and what values they possess.
- ☐ Write an appropriate doc-string for a function, including a description of the function, what inputs are required and their types, and what, if anything, the function returns.
- ☐ Import and use functions from other libraries, in particular Python's built-in `math` or `random` library.
- ☐ Write functions in a separate file and `import` them into a desired program.

Chapter 6: Students should be able to:

- ☐ Use functions as first class objects, assigning them to variable to be later used or returned by another function.
- ☐ Add event listeners to listen for mouse events within a PGL graphics window.
- ☐ Define appropriate callback functions to be called upon receiving an event.
- ☐ Use the `GWindow` object to share information between callback functions when necessary by adding values as attributes to the `GWindow` object.
- ☐ Retrieve what graphical objects (if any) are at a particular location on the graphics window.
- ☐ Create either interval or one-time timers which call a callback function with some specific timing.
- ☐ Create a `GArc` object with desired dimensions and starting and stopping points at the desired location on the graphics window.
- ☐ Create `GPolygon` objects, with properly placed vertices, at desired locations in the graphics window.
- ☐ Create a `GCompound` object and add other graphical elements to that object before placing at a desired location.

Chapter 8: Students should be able to:

- ☐ Create a Python list (array) with proper, valid elements inside.
- ☐ Add entries, remove entries, index, slice, and loop over lists.
- ☐ Create lists compactly using the list comprehension syntax.
- ☐ Open a text file to be read and looped over, performing desired operations.
- ☐ Open a text file to be written and add desired content.
- ☐ Use a list for tabulation, incrementing an index when some desired event occurs.
- ☐ Create multi-dimensional arrays and access specific elements within a multi-dimensional array.
- ☐ Use PGL's `GImage` class and associated methods to convert images to multi-dimensional arrays of pixel values.
- ☐ Manipulate the colors of pixels of a `GImage`.