**Pseudocode**

1. Create a folder as a package to store the python files.
   1. Create the following class files and store them in a folder: area, distance, mass, volume, temperature.
   2. Create the file main, which will be used for the main menu.
2. Create another folder as package to store the test files.
3. In each class file, do the following:
   1. In the constructor, name the attribute appropriately. For example, naming the attribute temp for the Temperature class.
   2. Using the property decorator, make getters and setters to return the value and set the value.
   3. Create two functions, each should convert the opposite units to the other. For example: Celsius to Fahrenheit and Fahrenheit to Celsius for the Temperature class.
      1. Create a variable called “result” and make it equal to the appropriate calculation needed to make the conversion.
   4. In the *‘if \_\_name\_\_ == “\_\_main\_\_’* construct, showcase an example of both conversions taking place.
4. For each class file, we must now make tests for each. Using Pytest, create tests for each conversion function and store them in the tests folder. Each of them should already be named appropriately. For example, test\_area to test the Area class. Now do the following for each testing file:
   1. Import the required class from the folder where the python files are being stored.
   2. In each test function, do the following:
      1. Create a variable that will store the object for the class. For example, in the Temperature class we could write: *‘number = Temperature(65)’*
      2. Create a variable to store the result, in the result we will access the conversion function from the specific class and use it on our variable named above, in this case “number”.
      3. Check to make sure that the result is equal to the math calculation that is carried out in the specific conversion function, using the variable we have named, “number”.
5. Lastly, we can work on the main menu.
   1. Import all the class files and their class.
   2. Create the main menu function which will display the main menu.
   3. Create the conversion screen, it should display 3 options, two of which are the conversions, and the third is the option to return to the main menu.
      1. The user should be given a choice as to which option they would like to select. This will be done in a while loop.
         1. If they choose a conversion option, the appropriate nested function will be called.
         2. If they choose the return to main menu option, the while loop will close out and the main menu function will be called.
         3. If the user input results in an error, it will run the conversion screen again.
   4. For each conversion option, such as Temperature, create a function.
      1. Within this function there should be two nested functions, with each converting the opposite units to the other.
      2. Each should have a variable to store whether the user wants to convert again, this should be set to True to begin with.
      3. Create a while loop for this variable, inside do the following:
         1. It should ask the user to enter the specific units of measurement.
         2. Then it uses this to create an object.
         3. The conversion should then take place using the conversion function from the class file in a variable called “result”.
         4. It will then print the conversion.
         5. The option to convert again will be given.
            1. If True, it will run again.
            2. If False, it will load back up the conversion screen.
   5. Create a function called “main”, in the function do the following:
      1. Call the main menu function to display the main menu.
      2. Create a user input to allow the user to choose which option they would like.
      3. In a while loop, use if statements to call on the appropriate function of the conversion they are looking for. For example, if they choose “2” it will call on the temperature function.
      4. If they choose “6”, which is the quit option. Since the while loop is now no longer true, it should print “Goodbye!” and then use the quit function to exit the program.
6. Make optimizations to the code and add suitable comments.