Problem-Solving Techniques for Programmers (Detailed)

1. To begin, step away from the computer. Start with a pencil and paper.
2. Understand the problem:
   * Read the problem several times.
   * Express the problem in your own words. Write this down. Make sketches if applicable.
   * Clearly define the “requirements”. What determines that the problem is solved? Write this down in your own words.
3. Identify the given information:
   * Some problems are presented with given values. Does this apply to this problem?
   * If so, identify the given values and select variable names that will hold them in the program.
4. Identify user inputs needed:
   * Many (not all) problems require inputs from the user. Are user inputs needed?
   * If so, select and write down the prompt that the user will see for each input.
   * Select and write down the variable name that will hold each input.
5. Identify the processing steps and any calculations that are needed:

* Are arithmetic calculations or formulas needed?
  + If so, write down expressions for the formula(s) using the variable names chosen.
  + Solve the problem manually with some sample numbers. Don’t know a formula? Google it.
* Does the program need to make any decisions?
  + If so, write down the logic for each decision, using the variable names you have chosen.
* Does the program need to repeat the same processing steps multiple times?
  + If so, consider using a loop and write down the steps that must be inside the loop.
* Do the requirements specify that you use/create one or more functions (in addition to the main function)?
  + If so, determine precisely what task each function should do, or accomplish, and choose a function name that describes the task. Write down the function names you select.
  + Does a specific function need data to perform its task?
    - If so, select parameter names for the data and place them inside the parentheses of the function.
  + Write down the processing steps and/or output steps that the function must perform, using the parameter names chosen above.
  + Determine where in main each function should be called (executed).

1. Identify the required outputs:
   * Usually, this can be determined by re-checking the requirements.
   * Write down the outputs required. Include any formats required (such as currency or decimal places).
2. Sequence all of the above steps in the order that is required.
3. Now, turn to your computer and start the IDLE program, but keep the paper handy.
4. Write the pseudocode:

* Open a new window in IDLE script mode and enter the sequence from Step 7 as Python comments.
* Use plain English and the chosen variable names to describe the steps needed.
* Make sure all steps are sequenced in the proper order.

1. Create your program in Python syntax by referring to the pseudocode.

* Translate the pseudocode into Python statements.
* Edit the pseudocode as needed as you code the program.
* Add more comments as needed to clarify code.

1. Test and debug the program.

* Run the program several times with different inputs to be sure it works as required. Fix errors.
* Check the requirements again. Be sure that your program meets all requirements.
* Check your spelling. Spelling mistakes reflect poor attention to detail, and that will worry your customers.