Table 2.2 ***How To Think Like A Programmer (HTTLAP)***

1. Understanding the problem	
You have to understand the problem. Do not go to the next stage until you have done this.	What are you being asked to do? What is required? Try restating the problem? Can the problem be better expressed by drawing a diagram or a picture? By using mathematical notation? By building a model out of wood, paper, or card? What is the *unknown*? Is finding the unknown part of the problem statement? Write down what you **do know** about the problem. Have you made any assumptions? If so, what can/should you do about it?
Sleep on the problem and come back to it fresh.	What are the principal parts of the problem? Are there several parts to the problem?
2. Devising a plan to solve the problem	
Start thinking about the information you have and what the solution is required to do.	Have you solved this problem before (perhaps with different values/quantities)? Is this problem *similar* to one you have met before? If so, can you use any knowledge from that problem here? Does the solution to that problem apply to this one in any way? Are some parts of the problem more easily solved than others? If the problem is too hard, can you solve a simpler version of it, or a related problem? Does restating the problem (perhaps telling it in your own words to someone else) help you to get a grip on it? Try describing the problem in a different *language* (e.g., diagrammatically, pictorially, using mathematics, building a physical model or representation).
Stop bothering your brain and sleep on the problem.	Did you make use of all the information in the problem statement? Can you satisfy all the conditions of the problem? Have you left anything out?
3. Carrying out the plan	
Write down your solution. Pay attention to things done in order (sequence), things done conditionally (selection) and things done repeatedly (iteration).	Write down the basic sequence of actions necessary to solve the general problem. This may involve hiding some of the detail in order to get the overall sequence correct. Is your ordering of actions correct? Does the order rely on certain things to be true? If so, do you know these things from the problem statement? Or have you made any assumptions? If you have made assumptions then how will you verify that they are correct? If you cannot see a way to solving the whole problem can you see any parts of the problem that you can solve? If the problem is too complicated, try removing some of the conditions/constraints and seeing if that gives you a way in. Go back to your sequence of actions. Should all actions be carried out in *every* circumstance? Should some actions (or groups/blocks of actions) only be carried out when certain conditions are met? Go back to your sequence of actions. Is carrying out each action once only sufficient to give the desired outcome? If not, do you have actions (or groups/blocks of actions) that must therefore be *repeated*?
Sleep on it again.	Go back once more. Do any of your actions/blocks of actions belong *inside* others? For example, do you have a block of actions that must be repeated, but only when some condition is met?
4. Assessing the result	
Examine the results obtained when using your solution.	Use your solution to meet the requirements of the problem. Did you get the right answer or the correct outcome? If not, why not? Where did your solution go wrong? If you think you did get the right answer, how can you be sure? Did any parts of the solution seem cumbersome or not very sensible? Can you make those parts simpler, quicker, or clearer?
Get someone else to use or follow your solution.	Give your solution to someone else and ask them to use it to complete the task. Were your instructions clear enough for them? Did they have to ask for help or clarification? Did they misinterpret your instructions? If so, why? Did they get the same answer as when you did it? If not, whose answer was correct?

5. Describing what you have learned

Make a record of your achievements and the difficulties you encountered.

What did you learn from this exercise? What do you know now that you did not know before you started?

What particular difficulties did you encounter? Were there any aspects of the problem that caused you particular difficulty? If so, do you think you would know how to tackle them if you met something similar in the future?

Compare your finished solution with your first attempt. What do the differences teach you?

6. Documenting the solution

Explain your solution. Make sure that it can be understood.

Are there any aspects of your solution that are hard to understand? Is this because they are badly written, or simply because the solution is just complicated? If you were to pick up your solution in five years time do you think any bits would be hard to understand?