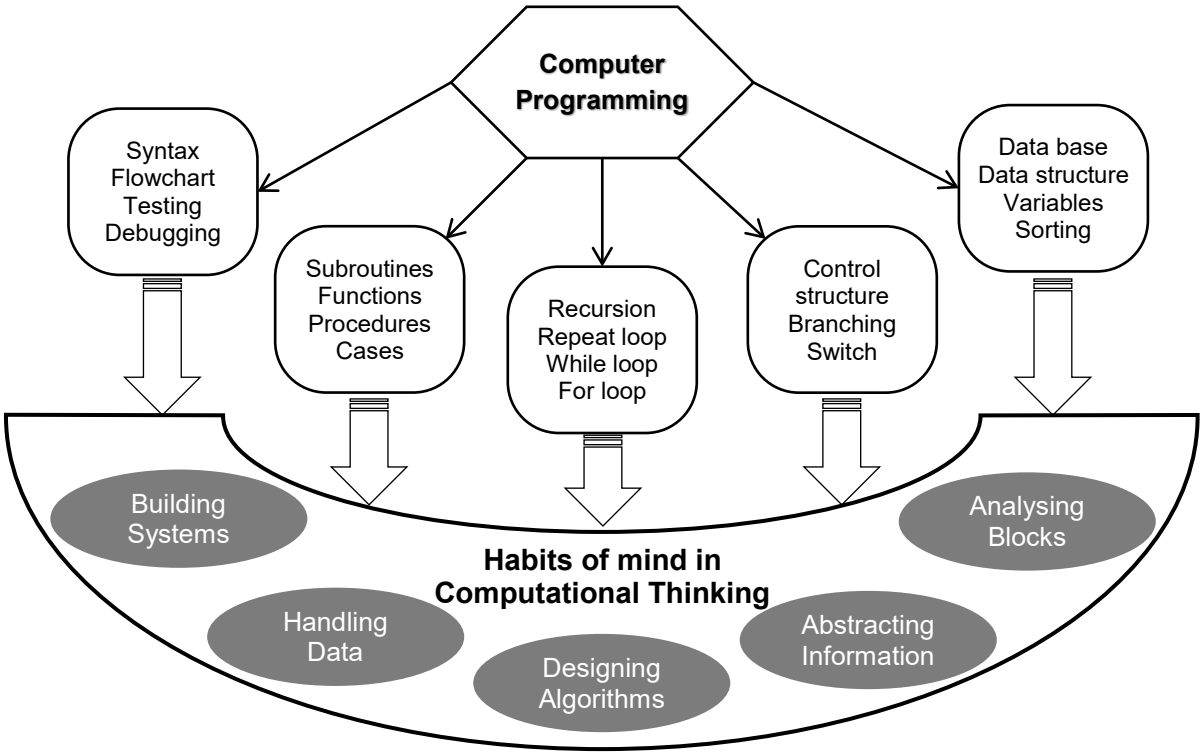

Figure 6. Concepts and skills in computer programming leading to the development of habits in computational thinking

It is also typical of such courses to require students to complete programming exercises to reinforce these concepts, achieve a deeper understanding, and be skillful in using them. Over time, as the student does more of these exercises and uses these concepts more frequently, it is not unreasonable to expect these skills to become internalized as habits that the student is likely to rely on in problem situation.

While it is difficult to say which skills or concepts develop what aspect of computational thinking, it is not hard to see that these can lead to developing some of the habits in Figure 6. For instance, constantly working with flowcharts, subroutines, functions and procedures in writing programs and solving problems serves to train one's mind to think systematically, visualize a big picture, while taking care to analyze smaller units and blocks in the solution process. At the same time, working with data and defining or creating variables and data structures are part and parcel of computer programming.

These skills eventually build one's familiarity with methods of looking at data and finding ways to use them effectively in problem solving, as is also demonstrated in Example 1 in the preceding section. Simulation models can be effectively designed and implemented if one has the knowledge and habit of thinking algorithmically, as illustrated in Example 2. Finally, abstracting information and describing them as variables in a computer program helps develop one's mind to think of factors in the real world as mathematical variables, as shown in Example 3. It is evident that these are the type of thinking and habits of mind that have eventually led to successful designs of solutions, implementable on a computing machine in the examples discussed.

Concluding Remarks

In this paper, we examine and explicate aspects of habits of mind related to computational thinking, so that we may see their relevance in mathematical modelling. Using three examples, these habits or dispositions towards problem solving are teased out and linked to certain parts or phases in the solution process of the mathematical modelling tasks.

It is *not* the intent of this paper to advocate or even suggest that computational thinking is important for mathematics, nor is it its purpose to discuss related issues in mathematics education. It is acknowledged that in this respect, opinions do differ, and the diverse views on computational thinking and its relevance to mathematics education, coupled with the lack of research in this area, have meant that a common understanding does not seem to exist at the moment. Nonetheless, what this paper attempts to do is to illustrate and elucidate the link between some habits developed consciously through purposeful computer programming exercises and certain skill sets useful to mathematical modelling.

These mental habits of computer programmers or scientists, however, are not something that can be acquired through a few hours of coding lessons or even one course on a programming language. Rather, these are developed through many hours of computer programming exercises or projects, and frequent use of relevant programming skills and techniques in problem solving. In terms of mathematical modelling, these habits will further strengthen one's ability and competence in handling and solving modelling tasks.

References

[1] Ang, K.C. (2018). *Mathematical Modelling for Teachers: Resources, Pedagogy and Practice*. London: Routledge.

[2] Ang, K.C. (2019). Exploring the nexus between computational thinking and mathematical modelling, paper presented at *The 19th International Conference on the Teaching of Mathematical Modelling and Applications*, Hong Kong.

[3] Costa, A. L. and Kallick, B. (2008). Learning and leading with habits of mind: 16 essential characteristics for success. Alexandria: Association for Supervision and Curriculum Development.

[4] Goldenberg, E. P. (1996). 'Habits of Mind' as an Organizer for the Curriculum. *Journal of Education*, 178 (1), 13–34.

[5] Hu, C. (2011). Computational thinking – what it might mean and what we might do about it, *Proceedings of the 16th annual joint conference on innovation and technology in computer science education*, 223–227. Darmstadt, Germany.

[6] Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L. and Settle, A. (2014). Computational thinking in K-9 Education, *Proceedings of the 19th annual joint conference on innovation and technology in computer science education*, 1–28. Uppsala, Sweden.

[7] Mazumdar, J., Ang, K. C. and Soh, L. L. (1991). A mathematical study of non-Newtonian blood flow through elastic arteries, *Australasian Physical & Engineering Sciences in Medicine*, 14(2), 65–73.

[8] Mohaghegh, M. and McCauley, M. (2016). Computational thinking: the skill set of the 21st century, *International Journal of Computer Science and Information Technologies*, 7(3), 1524–1530.

[9] Nardelli, E. (2019). Do we really need computational thinking? *Communications of the ACM,* 62(2), 32–35.

[10] Roach, M. R. and Burton, A. C. (1957). The reason for the shape of the distensibility curves of arteries, *Canadian Journal of Biochemistry and Physiology*, 35, 681–690.

[11] Sanford, J. (2013). Core concepts of computational thinking, *International Journal of Teaching and Case Studies*, 4(1), 1–12.

[12] Sanford, J. F. and Naidu, J. T. (2016). Computational thinking concepts for grade school, *Contemporary Issues in Educational Research*, 9(1), 23–31.

[13] Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. and Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms, *Journal of Science Education and Technology*, 25, 127–147.

[14] Wing, J. (2006). Computational Thinking, *Communications of the ACM,* 19(3), 33–35.

[15] Yasar, O. (2018). A new perspective on computational thinking, *Communications of the ACM,* 61(7), 33–39.