Symposium Report: Working Group on Curriculum Design at the Secondary Level. In Online Proceedings of the *Computational Thinking in Mathematics Education Symposium*, Oshawa, Ontario (Canada) *October 2017. Lovric & Taylor*

REPORT OF WORKING GROUP ON CURRICULUM DESIGN AT THE SECONDARY LEVEL.

From notes taken by Miroslav Lovric, McMaster University and Peter Taylor, Queen's University.

October 2017

The discussion ranged over a large number of issues. Certainly, we all felt there was great value in having a significant number of activities in the curriculum that involved different kinds of computational thinking. We felt there might be two kinds of such activities:

1. those that were computational in nature and that perhaps even offered immediate opportunities for coding
2. those that might be enhanced or developed into a form that provided a computational opportunity.

The first category contained problems that might be infeasible without computational power.

It seemed to us that many of the activities found in a standard math classroom were of the second type and some systematic attention might be paid to several these. The question arises as to how we might construct a CT version of such activities. As an example to work on, we chose a modeling problem in biology. A bird has to allocate its time between bouts of foraging in a patch and traveling from a depleted patch to a new one. The payoff p from foraging in a patch for time t exhibits diminishing returns so that the bird must decide how long to remain in the patch before "moving on," and paying the cost of travel. There were many ideas for building a computational structure around this problem that ranged all the way from a python code, to a spreadsheet, to having kids engage in a series of activities and have to run around the school yard when they decided to move to a new one!

We moved onto a more general discussion. Time was a big factor. Teachers already often feel hard-pressed to "cover" the curriculum. Coding takes time, and indeed if we want the kids to code there is even instructional time (in coding) that needs to be factored in. Perhaps in the future that will be seamlessly built into the curriculum at different stages.

It was suggested that time spent learning to code contributed in an important way to mathematical development, for example it can build organized, algorithmic ways of thought. A great feature of coding is that the algorithm can be immediately "run" as a check on its correctness.

Related to this is the obvious "hands-on" aspect of CT. This elevates it in a significant way above the use of standard technology (applets/ready-made software) used to do calculations.