Preparing young minds for computational thinking

Conferencista: Srini Devadas

Famous computer scientists have invented new programming paradigms, languages, software engineering practices and algorithms that have resulted in today's Information Age. Many of these talented inventors spent their formative years learning how to program computers that were available during their youth. How can we prepare and nurture young minds to follow in their footsteps?

Computational thinking involves modeling problems in the real world and expressing them in a way that computers can execute their solutions. It has been touted as a prerequisite to being a successful computer scientist or programmer, or using computer technology in a facile manner. We believe that computational thinking is a higher level skill as compared to the more rudimentary skills of reading, comprehension and analysis and has to be treated that way.

We argue that learning and sharpening analytical skills such as performing logical analysis and arguments is a prerequisite to computational thinking. Further, it is important to learn precision in analysis and implication at an early stage. One way of learning analytic precision is to programmatically write arguments or solutions to problems so they can be checked by a computer, which can then provide diagnostic feedback that is crucial to learning. We briefly demonstrate preliminary lessons we have developed that embody the teaching philosophy described above; these lessons involve solving recreational puzzles and writing commands that represent their solutions.