Greeting:
As the school year wraps up, we want to send one more huge thank you for all that you do to support student learning.
We have been busy preparing to enroll some of you in Code VA’s K-6 coaches, and we welcome those of you who completed the academy last summer into our CS Integration microcredentials courses. Please be on the lookout for communications from TCEP@odu.edu about enrollment procedures, stipends, and other details.

We hope you enjoy a peaceful and fun summer, and perhaps see examples of computer science and computational thinking in the many computing devices that bring us together and help us accomplish our summer goals!
The ARCS team

Concept Corner
Computational Thinking is an interconnected set of skills and practices for solving complex problems, and a necessary core content area in computer science for the collective good to expand computer science in the Commonwealth. Computational thinking relates to making efficient use of the capabilities of our computing device to solve complex real-world problems. It is important to note that computational thinking is not just programming.

Programming is the practice of developing a set of instructions that a computer can understand and execute, as well as debugging, organizing, and applying that code to appropriate problem-solving contexts. The skills and practices required for computational thinking are broader, leveraging concepts and skills from computer science and applying them to other contexts. Examples of such skills include problem abstraction, proper tool selection, problem decomposition, pattern recognition, algorithmic thinking and precise communication of a “method” to solve a problem. All of us use “computational thinking” in solving our day-to-day problems to some extent already. Identifying and listing down all the steps of your favorite baking recipe is an example. Similarly, implicit computational thinking activities can also be seen and demonstrated in all academic areas. Instructional resources related to computational thinking can be found on the Virginia Department of Education’s GoOpenVA. Educators are encouraged to use these resources to better integrate computational thinking and computer science for the collective good to expand computer science in the Commonwealth.

Pedagogy Pointers
Free unplugged lesson: code.org has excellent free lesson plans, including one focused on computational thinking at an elementary level. This lesson is unplugged and does not require computing devices to be used by students. The lesson includes materials, a teacher guide, and videos. Can be integrated into lessons for many CS SOLs pertaining to Algorithms and Programming.

Cross-curricular computational thinking: code.org is a site jam-packed with cross-curricular ideas for integrating computational thinking into other areas of the curriculum and free lessons. All four core content areas – math, science, social studies, and language arts – are included with examples of lessons and activities. Some content may be too advanced for early elementary students. Can be integrated into lessons for many CS SOLs pertaining to Algorithms and Programming.

Computer Science in the Commonwealth
Computational Thinking is a building block of computer science. It is the thought process behind creativity and innovation in computer science. It provides guidance in problem-solving and the application of creativity. The four main pillars of computational thinking are abstraction, pattern recognition, decomposition, and algorithmic thinking. The utilization of computational thinking can be seen and demonstrated in all academic areas. Instructional resources related to computational thinking can be found on the Virginia Department of Education’s GoOpenVA. Educators are encouraged to use these resources to better integrate computational thinking and computer science for the collective good to expand computer science in the Commonwealth.

Engaging All Learners
This month, our newsletter is focused on computational thinking, a problem-solving process initiated through steps or stages that can be automated for completion by a computing device. Moreover, computational thinking activities can also be “unplugged,” making them appropriate for use in solving problems in areas like social studies and literacy in addition to math and science fields. Digital Promise is a non-profit agency whose mission is to promote learning for all students through innovation in education. Digital Promise is committed to expanding opportunities for underrepresented and underprivileged students to gain experiences in computing, and many of their resources are aligned with a diversity, equity and inclusion lens. A great example is their report titled “Computational Thinking for an Inclusive World: A Resource for Educators to Learn and Lead.” Digital Promise and CodeVA are valuable sources of information and activities designed to support student engagement in computational thinking.

United States Education Department Grant U411C190032. The contents of this newsletter were developed under a grant from the US Department of Education. However, these contents do not necessarily represent the policy of the US Department of Education, and you should not assume endorsement by the Federal Government.