TN: Coding and Calculating for models used for USACE planning community

Todd & Kyle

Principles of coding

1. A working model/code is always preferable to a broken one
   1. This concept is important for all coders, modelers, and/or programmers, but is especially pertinent for those new to coding. Reaching the solution through clunky code is a better approach than never solving the problem because the coder is attempting to solve the problem in a unique or overly complicated way.

Example: On the first test of the semester, two students in an intro to coding class are asked to write a block of code that calculates the number of leap years that have occurred in their lifetime.

Coder A solves the problem with a single line of code

* Roundup((Current year – first leap year after birth) / 4)

Coder B solves the problem with a while loop, and a counter.

* Let n = first leap year after birth
* let y = current year
* let counter = 1
* while n ≠ y [
  + n = n + 4
  + counter = counter + 1
  + next while
  + ]

The while loop iterates through the leap years, beginning with the first leap year after birth, and increasing until the current year is reached. A counter is increased by 1 during each iteration. Once the current year has been reached, the loop stops and counter is the number of leap years Coder B has experienced.

It doesn't matter which block of code is used because both will generate the same answer. Each coder developed a block of code that answered the question, and the students can move on to the next question. In more practical situations, coders should get their code working with an approach they are comfortable and proficient with first, then spend time optimizing and improving the code.

1. Avoid premature optimization
2. Don't repeat blocks of code:
3. Simplicity is the key. Overly complex code leads to errors
4. Make code easy to read
5. Test each function as it's written