Lesson 3.15 — Fencepost & Sentinel Loops

Overview

Objectives — ***Students will be able to...***

- **Describe** when to use fencepost and sentinel loops.
- **Use** proper syntax to construct these control structures.

Assessments — ***Students will...***

- **Teach** a mini-lesson explaining the relationship between parameters and values stored in memory

Homework — ***Students will...***

- **Read** BJP 5.3
- **Complete** exercises #6 & 8
- **Summarize** all of your daily notes if not already done

Materials & Prep

- **Projector and computer**
- **Whiteboard and markers**
- **Group copies** of WS 3.15
- **3 or more classroom copies** of the textbook

Pacing Guide

Section	Total Time
Bell-work and attendance	5min
Mini-lesson planning & prep	15min
Student presentations & practice	30min

Procedure

Today 3 student teams will teach a lesson on fencepost algorithms, sentinel loops, or sentinel loops with if statements. Your hook will be to turn the class over to students as soon as they enter. Student groups are expected to generate sample questions; let students know that you will collect those questions to use on quizzes or as bellwork.

Bell-work and Attendance [5 minutes]

On the board, on the projector, or in a group handout, let students know that they need to prepare a 5 minute lesson and a 2- 5 minute class activity to teach their topic. Use the grading rubric as outlined here:

3 pts.	2 pts.	1 pts.	0 pts.
Presentation includes definitions and an example with proper syntax.	Presentation includes definitions or an example with proper syntax.	Presentation includes definitions or an example with proper syntax with one mistake.	Presentation includes definitions or an example with proper syntax with many mistakes.
Presentation includes a non-example as helpful contrast.	Presentation includes a non-example that is marginally helpful.	Presentation includes a non-example that does not add to comprehension.	Presentation includes a non-example that adds confusion, or presentation does not include a non-example.

3 pts.	2 pts.	1 pts.	0 pts.
Presentation includes a helpful tip that is clearly explained and concisely stated.	Presentation includes a helpful tip that is clearly explained or concisely stated.	Presentation includes a helpful tip that is not clearly explained and may include a small error.	Presentation does not include a helpful tip or hint.

Mini-Lesson Planning & Prep [15 minutes]

1. Assign each group a subsection of section 5.2 "Fencepost Algorithms," and make sure that you circle that assignment on each groups' copy of WWS 3.15. Student groups should take 15 minutes to review their section, re-read the example on the pages following the example, then figure out how they want to explain the algorithm to the class.
2. On the board or overhead, give students a few things they should consider in planning their mini lesson:
   a. Who is going to speak when?
   b. How are you going to illustrate the flow of control?
   c. What do you need to have up on the board to illustrate your mini-lesson, and who is in charge of writing it out?
   d. Where and how will you feature the output produced by your code segment?
   e. What is your mini-activity going to look like? (You might want to assign 1–2 people to work on this section while the rest of the group works on the lesson.)
3. Have student groups sequentially teach through fencepost algorithms, sentinel loops, and fenceposts with `if` statements.
4. Encourage students to add these strategies to their Tricky Code Cheat Sheet.

Student Presentations & Practice [30 minutes]

Accommodation and Differentiation

If your class learns better through tactile or visio-spatial learning, you can change this assignment to a make-a-poster lesson, having students work in pairs or triplets to create an informative poster on one of the topics. If you're fortunate enough to have a theatrical or musical class, invite them to create a song, poem, or narrated dance/play that teaches their topic. For poems or songs, encourage students to write them out or record them so you can display them around your room.

Common Student Questions

Since your student-instructors won't be able to answer in-depth questions on their topic, you should be ready to assist during the Q & A section of their lesson. Some student questions that have popped up in the past, with their answers, are listed below:

1. The sentinel loop example in the book says that the sentinel value will be read and added to the sum unless we do a fencepost algorithm or if statement. Why is that? Doesn't the test evaluate to false and terminate the loop?

   Write an example on the board in pseudocode or actual code, and trace the flow of control with your marker. In this case, the pseudocode in the book is a bit misleading because it looks like the test will terminate in the header. The prompt for the sentinel is already in the loop body, so the sentinel will be evaluated before loop termination.

   If you think your class will be comfortable, instruct your teaching group to do this illustration, and just help along.

2. What do you do if you don't know the first value to put in your fencepost algorithm? What if you're getting all of your data from user input?

3. When do we know to reverse the order of loop construction? The example from the book has us switching around a lot of stuff for the sentinel loop with if statements.

   Put an example up on the board (or have the student instructors do so), and trace the flow of control before and after rearranging the loop body.

Video

- BJP 5-3, *Sentinel Loops* http://media.pearsoncmg.com/aw/aw_reges_bjp_2/videoPlayer.php?id=c5-3
- CSE 142, *Fencepost* (0:28-11:20) https://www.youtube.com/watch?v=hpDQ9tdrj1Y&start=28
- CSE 142, *Sentinel Loops* (15:56) https://www.youtube.com/watch?v=hpDQ9tdrj1Y&start=681

Forum discussion

Lesson 3.15 Fencepost & Sentinel Loops (TEALS Discourse account required)