

Problem Description

According to the problem on Canvas, “Short circuit evaluation is when the language evaluates the first portion of a BOOLEAN expression and if, knowing the result of the value, then skips the evaluation of second expression. For example, A & B is false if A is false... no need to evaluate B.”

So, I needed to test out ADA, C-SHELL, PHP, and PERL to check if they have Short Circuit Evaluation. Below are the results.

Results

| Language | Has Short Circuit Evaluation |
|--------------|------------------------------|
| ADA | No |
| Bourne Shell | Yes |
| PHP | Yes |
| PERL | Yes |

Code for each program

| Language | Code | Output |
|----------|--|---|
| Ada | <pre>-- Name: marco salazar, Date: 9/8/2020, -- purpose: to check if Ada has short circuit evaluation with Ada.Text_IO; use Ada.Text_IO; procedure Hello is -- function that prints out a line when executed. function Short_Circuit(Var : Integer) return Boolean is begin</pre> | <p>If this line is followed by two lines then Ada does not have short circuit evaluation.</p> <p>Function call is executed.</p> <p>Function call is executed.</p> |

| | | |
|--------|--|--|
| | <pre> Put_Line("Function call is executed."); return True; end Short_Circuit; begin Put_Line("If this line is followed by two lines then Ada does not have short circuit evaluation."); -- if ada has short circuit evaluation after it evaluates 1=0 it should not call Short_Circuit() if 1=0 and Short_Circuit(1) then Put(""); end if; if Short_Circuit(1) and 1=0 then Put(""); end if; end Hello; </pre> | |
| Bourne | <pre> #!/bin/sh # Name: Marco Salazar, Date: 9/9/2020, Purpose: to test if Bourne shell has short circuit evaluation prints() { echo "Not " } if prints && [[0 = 1]]; then echo "" fi echo "If there is a not above then the prints function works. real test starts below" echo "" if [[0 = 1]] && prints; then echo "" fi echo "Short Circuit Evaluation" </pre> | <p>Not</p> <p>If there is a not above then the prints function works. real test starts below</p> <p>Short Circuit Evaluation</p> |
| PHP | <pre> <!-- Name: Marco Salazar, Date: 9/9/2020, Purpose: to test if php has short circuit evaluation --> <!DOCTYPE html> <html> <body> <?php function sayNot(){ echo "Not Short Circuited evaluation"; </pre> | <p>Not Short Circuited evaluation</p> <p>Ignore above line. sayNot() function works.</p> <p>Has Short Circuited evaluation</p> |

| | | |
|------|--|---|
| | <pre> return 1; } if(sayNot() == 1 && 1 == 0){ echo "\nif"; } else { echo "\nIgnore above line. sayNot() function works.\n\n"; } if(1 == 0 && sayNot() == 1){ echo "\nif"; } else { echo "Has Short Circuited evaluation"; } ?> </body> </html> </pre> | |
| PERL | <pre> #!/usr/bin/perl # Name: Marco Salazar, Date: 9/9/2020, Purpose: test to see if Perl has short circuit evaluation. sub sayNot { print "Not "; } if(sayNot() && 1 == 0) { print "\nignore\n"; } else { print "\nIgnore above line. sayNot() function works. test below\n\n"; } if(1 == 0 && sayNot()) { print "ignore\n"; } else { print "short circuit evaluation\n"; } </pre> | <p>Not Ignore above line. sayNot() function works. test below</p> <p>short circuit evaluation</p> |