COP 3363

SPRING 2020

RECITATION 4

FOR LOOPS

▸ for(<initial cond>; <halt cond>; <increment cond>){<statements>}

▸ Repetition structure suited for a set number of iterations

 ▸ FOR loops aren't *required* to have a set number of iterations, but other loop structures are easier to implement in those scenarios

▸ The initial condition must meet (eval to TRUE | nonzero) the halt condition in order for the loop to begin execution

▸ The loop exits when the halt condition evaluates to FALSE | 0

WHILE LOOPS

▸ while(<halt condition>){<statements>}

▸ Repetition structure best suited for loops which do not use a set number of iterations

  ▸ Similarly, WHILE loops can be implemented to run for a set number of iterations, but FOR loops are easier to implement here

▸ The halt condition must evaluate to TRUE/nonzero in order for the loop to begin execution

▸ The loop exits when the halt condition evaluates to FALSE | 0

DO WHILE LOOPS

▸ do{<statements>}while(<halt cond>);

▸ Similar uses as a while loop

▸ Always runs at least once

▸ The loop exits when the halt condition evaluates to FALSE | 0

BREAK & CONTINUE

▸ BREAK will cause the loop to stop execution and jump to the first line of code below the loop block

▸ CONTINUE will cause the program to cease executing the statements in the current iteration of a loop, and jump to the halt condition