

Repetition Control Statements

<i>Repetition Control Statements: Sample Program List</i>	
basic_for_loop.py	<i>This program illustrates basic for-each loops.</i>
repeat_until_valid_age.py	<i>This program illustrates a commonly used pattern looping until the user enters a valid input value. The while statement is used for this program.</i>
compute_sum_of_N_positive_integers.py	<i>This program shows two variations of computing the sum of N positive integers entered by the user. The first variation stops and aborts the computation if a negative number is entered. The second variation continues the computation even when negative numbers are entered. Negative numbers are simply ignored by the program. The while statement is used in both variations.</i>
imprecise_loop_example_1.py	<i>The purpose of this sample program is to provide a concrete example of approximation errors and the danger of infinite loop due to approximation errors. The lesson to learn here is to avoid using float variables in the repetition termination test.</i>
imprecise_loop_example_2.py	<i>This is another example of approximation errors to reiterate the danger of using float variables in the repetition termination test.</i>
play_arithmetic_games_Step1.py (version 1) play_arithmetic_games_Step2.py (version 2) play_arithmetic_games.py (final version)	<i>This shows a sequence of incrementally developed program for playing arithmetic games. For each game, a user is asked for N, the number of questions to answer. Each question is in the form <num1> <op> <num2> where the <op> is either addition (+) or subtraction (-). The numbers ranges from 0 to 99, inclusive.</i> <i>The sample program illustrates the use of several repetition statements of very common pattern. In addition, an effective use of selection statements and functions are shown in the program.</i>
draw_cross_using_for.py	<i>This program illustrates the creation of formatted strings using the for loop. Instead of using a fixed format string such as "%3d", the program creates the format string dynamically that changes in each iteration of the loop. The program outputs asterisks (*) is patterns.</i>