Lab 4: Control Structures

…Previously

Recall that in the previous lab you got introduced to Python and used various data types, operators, and variables to write simple Python codes. Moreover, the session required you to convert some of the algorithms/flowcharts we discussed a while back into Python codes.

Today…

Control structures in Python

- Conditionals/branching/decisions (**if…else**, **if…elif…else**)
- Repetition/ Loops/iteration (**for** and **while** loops, **break** and **continue** statements)

Activity 1: Refresher activity

1. Take two numbers from the user and display the results of various operators (arithmetic, relational, logical).
2. Accept three sides of a triangle from the user and compute its area using Heron's formula.

Activity 2: Self-reading

- Take a moment to recap our discussion on Control Structures in Python.
- As you recap, you may use your Python/iPython terminal or Spyder IDE to execute your codes.

Activity 3: Control Structures

Write a Python program to:

1. Find the minimum of two integer numbers accepted from the user. Do the same for three numbers. How about for four?

2. Determine if a given integer number is even or odd.

3. Get mark/score of a student from the user and display his/her performance according to the following criteria:
   Excellent = [100, 90] Satisfactory = (70, 60]
   Very good = (90, 80] Pass = (60, 50]
   Good = (80, 70] Fail = (50, 0]

4. Find the roots of a quadratic equation based on the following cases:
   Case 1: $b^2 - 4ac > 0$
   Case 2: $b^2 - 4ac = 0$
   Case 3: $b^2 - 4ac < 0$

5. Display the message "Kuzu zangpo" 25 times.

6. Display integer numbers from 1 up to N; where N is a user-supplied number (using both **for** and **while** loops). How about from N down to 1 (i.e. the reverse order)?

7. Redo the above question but this time introduce **break/continue** statement somewhere in your loop. Notice the difference between **break** and **continue**.

8. Display even numbers from 1 up to N (where N is a user-supplied number).

9. Find the product of numbers from 1 to 100 (i.e. 1x2x3…x100).

10. Compute the factorial of a user-supplied number N. Recall that factorial of (denoted by N!) is 1x2x3x…xN