```
Chp 7: Looping

- This chapter has four parts:
  1. Basics: Looping? Why and How?
  2. The three constructs: while, for, do-while
  3. Break and continue statement
  4. Nested loops - for 2D looping

================================================

(1)
- Looping? Why and How?
  - Why?
    - Sometimes we need statements to be executed **repeatedly**
    - And we can dynamically control the repeated-ness during
      program runtime
    - The program can be more flexible and powerful
      e.g., multiplication table, average mark of students
  - How?
    - Always four elements:
      1. Initialize
         - initialize the loop control variable
      2. Test condition
         - evaluate it -> whether to continue the looping???
         - involve loop control variable)
      3. Loop body
         - statements to be executed in the loop
      4. Update
         - need to modify the value of the loop control variable
         - so that... test condition result could be changed
           (otherwise, it is a forever loop)
    - usually, we use i,j,k,counter as variable names for
      loop control variables (if purely for counting purpose)

- Two types of loops:
  - Counter-controlled loops
    - the number of loop-body repetitions is known before
      the loop starts execution
  - Sentinel-controlled loops
    - the number of loop-body repetitions is NOT known before
      the loop starts execution.
      - Note: Usually, a sentinel value (such as -1, different
        from the regular data) is used to determine whether to
        execute the loop body.


================================================

(2)
- The three constructs: while, for, do-while
  - they all have the four steps above
    (note: but some of them may do it **implicitly**)

  (A) while
      - Syntax:
           while ( Test )    // boolean expression
              Statement
      - Sentinel-controlled loops
      - Note:
        - We may setup some related variables before
          starting the loop body
        - and then modify them inside the loop body

  (B) for
      - Syntax:
           for ( Initialize ; Test ; Update )
              Statement;
      - Counter-controlled loops
```

```
      - Note:
        - Explicitly contains all four elements
        - The execution order!!!!
          Initialize -> Test -> Statement -> Update -> Test -> ...
        - for (;;)  /* an infinite loop */

  (C) do-while
      - Syntax:
           do
              Statement;
           while (Test);
      - Sentinel-controlled loops
      - Note:
        - Test is performed after executing the statement
        - loop body always executed at least once in the
          program logic

  - Which loop do we use?
    - if counter-controlled loops
         -> for loop
      else
      if loop body runs at least once
         -> do-while loop
      else
         -> while loop


==============================================

(3)
- break statement
  - can be used inside a switch OR loop
  - causes immediate termination of the **innermost enclosing**
    loop or the switch statement

- continue statement
  - can be used inside a loop
  - control immediately passed to
    IF while/do-while
       -> the test condition step in the nearest enclosing loop
    IF for loop
       -> the update step in the "for" loop
    - i.e., any other statements after continue (and before the
      test/update step) will be skipped

==============================================

(4)
- Nested loops
  - A loop may appear inside another loop
  - can nest as many levels of loops as the hardware allows
  - applications (2D looping)
    - Printing 2-D tables
    - Printing patterns (also 2-D)
```