Exercise Set 3: Counting Loops

Topics:
Patterns: for loop: initialize, verify, loop body, update, post-mortem
Objects: type double and its operations
Java language: for loop syntax and semantics
built-in functions (math and others)
System: infinite loop - abort program
Exercises: print lists, star patterns, rolling ball

Overview:

These exercises let you practice simple counting loops. Remember to think about the problem first - specifying the task that has to be repeated, designing the counting method, figuring out what needs to be initialize, so the loop would start correctly, defining the update (counting) operation, and then designing the Loop Continuation Condition. To see the power of the counting loop we consider a number of different problems:

- We start with exercises where the loop counting variable only keeps track of the number of iterations (repetitions).
- We work on loops that use the counting variable for a simple task in the Loop Body.
- In some exercises the number of iterations is determined at run time.
- In some loops the body of the loop uses value that is related to the iteration number in non-trivial way. For example, we may want to plot the values of a function \( \sin(x) \) for the values of \( x \) in the interval (60, 270), in five degree intervals. Here, it is possible to set up the loop control variable to be the required value of \( x \).

They first key idea is to separate setup tasks from the repeating task. The second key step is to understand how is each iteration step related to the proposed counting method. It helps to make a table of the iterations, writing on each line the values that are iteration-dependent. It may help you in discovering a pattern, designing a counting method, or designing the formula for computing iteration-dependent values.

Exercise strategies for student

The first ten exercises use just straight counting from one to some limit. There are two harder exercises in this set (exercises 5 and 10). Here you need to compute the total of some values that were used in the loop body. For this type of problems you must make sure that the variable that will keep the cumulative result is declared and initialized before the loop starts. In exercise 8 a variable that keeps track of the current total price of the purchase needs to be initialized to zero before the loop starts. It is incremented by the price of each purchase inside the loop and the final value is printed after the loop exit.
Exercise Set 3: Counting Loops

The next set of exercises uses more complex counting patterns. For each of these problems you should write down a table of values needed in the consecutive iterations of the loop and look for the relationship to the basic counting by one (either starting at one or starting at zero). Try to see if one of the patterns in the table of counting patterns applies. Then select the appropriate loop control and loop body formula for the desired values. It is important to work out a few examples by hand to make sure your program will work correctly.

Exercise 11 shows three different methods that can be used when the task requires values that start at a value other than zero or one and increment by value other than one. Exercise 17 is another example of computing cumulative result.
Exercise Set 3: Counting Loops

Exercises:

Part 1: Solved problems

Simple counting:

1. Write a program that prints number n, n * n, and n * log(n) for the values of n from 1 to 10.

2. Write a program that will print the values of n and n! for the values of n from 1 to 8.

3. Write a program that will simulate ten rolls of a die. To do this use MathUtilities.randomInt function. Print the result of each roll.

4. Write the program that will simulate twenty tosses of a coin and print 0 for heads, 1 for tails.

5. Write a program that will count the number of tails in 100 tosses of a coin.

6. Print one column of a multiplication table. User types in the multiplier. Program prints the first ten multiples of the given multiplier.

7. Frame ten circles. Use random number generator to select the center (between 100 and 200). Use random number generator to select the radius (between 10 and 50).

8. Print the price of each purchase for five purchases. User types in number of items purchased and the price per item. Program prints the total for each set of items. For simplicity, all prices are in whole dollars.

9. Modify Exercise 8 so that the user first specifies how many sets of items will be purchased.

10. Modify Exercise 9 so that it also computes the total price for the whole purchase.

Complex counting patterns:

11. Paint ten circles in a row. All of them have the same size and touch each other. User selects the radius of the circles and the location of the first circle.
   (a) Count from 0 to nine, and compute the coordinates of the i-th circle relative to the first one.
   (b) Count by ones and compute the coordinates of the i-th circle relative to the previous one.
Exercise Set 3: Counting Loops

(c) Use the horizontal coordinate of the circle as the counting variable. Start with \( x_{\text{Center}} \) = coordinate of the first circle. Increment by 2 \( \times \) radius. Continue as long as the horizontal coordinate is less than 20 \( \times \) radius off from the first center.

12. Print the values of the function \( \sin(x) \).

(a) Ask the user to type in the low and high end of the interval (in whole degrees) and the increment. For example, you may want to plot the values of \( \sin(x) \) from 0\(^\circ\) to 360\(^\circ\) in 5\(^\circ\) intervals.

(b) Ask the user to type in the low and the high end of the interval (in whole degrees) and the number of values desired. Note, that if the interval is divided into \( n \) segments you will need to print \( n+1 \) values.

(c) Ask the user to type in the low end of the interval, the increment, and the number of segments.

13. Modify Exercise 11 so that the user selects the number of circles and the radius. The program determines how many will fit into the graphics window (size 300 by 300) and it will print the whole row in the middle of the window (vertical 150).

14. Write a program that will request one number between 1 and 9 from the user and will print the following pattern of stars (user input determines the number of rows and the number of stars in the first row).

```
*****
****
***
**
*
```

15. Write a function that will paint an arc (part of a circle). The arguments specify the center of the arc, the radius, the starting and ending angle. Use one degree angle increments. The coordinates of a point on a circle with the center \((x_{\text{center}}, y_{\text{center}})\), radius \(r\), and angle \(a\) are:

\[
x = x_{\text{center}} + r \times \text{MathUtilities.cosdeg}(a);
\]

\[
y = y_{\text{center}} + r \times \text{MathUtilities.sindeg}(a);
\]

16. Write a function that will fill a pyramid, given the center of its base and its width. The pyramid is a triangle with the base parallel to the ground with a 45\(^\circ\) angle rise.

17. Write a function that will compute the total savings accumulated after a given time. User types in the monthly contributions, the interest rate, and the number of months the contributions have been made. Assume that the interest is compounded and credited monthly.

18. Write a function that will paint a grid in the graphics window. User selects the number of squares into which the window should be divided.
Exercise Set 3: Counting Loops

19. Write a function that will paint ten circles of diameter 30 with the centers on the main diagonal of the graphics window.

20. Repeat exercise 19 using the other diagonal.