Java Assignment

Java Assignment 3: Loop Practice

Ver 3.2   Last Updated: 4/28/2021 8:52 AM

Let’s get some practice creating programs that repeat commands inside of a loop in order to accomplish a particular task. You may want to create these inside a main method of a class, one right after the next. For example,

```
public class LoopPractice {

  public static void main(String[] args) {

    //Loop 1
    System.out.println(“------------------Loop 1---------------------“
    int i = 1;
    while(  ) {

    }
    //Loop 2
    System.out.println(“------------------Loop 2---------------------“
    i = 1;
  …
  }//end main
} //end class
```

1. Make the following numbers print to the screen
   1
   4
   7
   10
   13
   16

2. Make the following numbers print to the screen
   20
   15
   10
   5

3. Print out 500 asterisks with no spaces or newlines – think System.out.print (make sure to use a blank System.out.println() when this is done so your next loop will start on a new line)

4. Ask the user to enter in a number. (You will need to create an instance of the Scanner class to read the System.in stream(keyboard)) Print out that many *'s on the screen without spaces or newlines. You may use your Scanner object in numbers 5, 6 and 9 also!!
   Example,
   Please enter a number
   7
   ****
   HINT: This is a lot like the last loop only the loop invariant is now a variable

5. Print all the odd numbers from 1 to a user specifiable upper limit (inclusive)
   Example,
   Please enter a number
   9
   1
   3
   5
   7
   9
   HINT: The number 9 should be included!!!

6. It's about now that most students are getting agitated with running through each of the loops in order to test the later loops. Make a menu that the user can choose from and allow them to select which loop practice option to run. The following loop will run forever until the user enters an exit option
   int choice;

```
while(true) {
  System.out.println("Please enter in a choice, 0 to quit");
  choice = input.nextInt();  //assumes a Scanner object has been created
  if (choice == 0) {
    break;  //exits the loop
  }
}//end loop
```

7. Write a program that reads in numbers from the keyboard and then when the user enters a -1, gives the average of the numbers. Allow for floating point values. Round off the answer to the nearest hundredth. See cookbook for rounding. Below is shown an example:

    ```
    Input a number, -1 to find average: 20
    Sum so far = 20, #'s so far = 1
    Input a number, -1 to find average: 25
    Sum so far = 45, #'s so far = 2
    Input a number, -1 to find average: 4
    Sum so far = 49, #'s so far = 3
    Input a number, -1 to find average: -1

    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
    Your average is 16.33 for the 3 numbers entered
    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
    ```

8. Write a method called dispAuthor(). You may define this above or below the main but NOT inside the main. The method should display your name, period you have java and the date and title of this project inside some sort of symbols surrounding the text. Call your method from the beginning of your main. Remember, to call a static method that has 0 inputs and 0 outputs, use the following approach;
    dispAuthor();
    EXAMPLE:

    ```
    ***************************
    *  Loop Practice          *
    *  4/20/2020              *
    *  by mr Hanley           *
    *  Java Period 4          *
    ***************************
    ```

9. Write a method called playSound() that plays a random sound from among 5 sounds that you have loaded up in the beginning of the program (see the cookbook for loading up sounds, use .wav, .midi or .au formats) Also, add a menu option to play a random sound.
    Call your method after you call dispAuthor from the main. You may define this above or below the other 2 methods but NOT inside. Remember, to call a non-static method that has 0 inputs and 0 outputs, use the following approach;

playSound();
NOTE: You MUST successfully load the 5 audio clips from disk into RAM before they can be played. To do this, we must declare 5 global AudioClips above the main but inside the class. See the cookbook for how to load sounds into your java program.

10. Probability:
    Rolling 6 sided dice can teach us about probability.
    Allow the user to choose a number of rolls for 2 six sided dice. Add the dice together and keep track of how many of each of the sums are rolled. Print out a frequency distribution table and then a frequency histogram.
    Example,
    How many pair to roll->**1000**
    Rolling…

```
Roll            Frequency
------          --------
2               23
3               52
4               84
5               94
6               156
7               167
8               150
9               117
10              78
11              49
12              30


            @@@ Frequency Histogram @@@
2      |=======================
3      |====================================================
4      |====================================================================================
5      |==============================================================================================
6
       |==========================================================================================================================
==================================
7
       |==========================================================================================================================
=============================================
8
       |==========================================================================================================================
============================
9      |=====================================================================================================================
10     |==============================================================================
11     |=================================================
12     |==============================
```

NOTE: During the this project, you are going to define 3 different methods;
the main method: Will load up 5 sound files from disk into RAM, call the other two methods in

the beginning and then run through several example loops.

dispAuthor: this method will display some information about who wrote the program, what the title of the program is and what period and date this program was generated.

playSound: this method will choose from among 5 loaded audio clips and play one of them randomly when the program starts up. For this, there will need to be 5 global variables to hold the audio clips.

Bonus # 1: Pyramid

Using a combination of loops, ask the user what size pyramid they want to display and then print a scalable pyramid.
Example,
What size pyramid? **4**

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

What size pyramid? **5**

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

What size pyramid? **2**

```
     *
    ***
```

Bonus # 2: Multiplication Table

Using nested loops, print out a multiplication table for the numbers 1-?
To help out, try printing out the first two lines of the heading as follows;

```
System.out.println(“X\t|\t1\t2\t3\t4\t5\t6\t7\t8\t9”);//etc
System.out.println(“-------------------------------------------------”);
```

Then start your loops

```
int r=?. c=?
while(    )
```

MUST ASK THE USER WHAT THE HIGHEST NUMBER FOR THE TABLE HEADINGS IS From 2->20

```
X	|	1	2	3	4	5	6	7	8	9
--------------------------------------------------------------------------------
1	|	1	2	3	4	5	6	7	8	9
2	|	2	4	6	8	10	12	14	16	18
3	|	3	6	9	12	15	18	21	24	27
4	|	4	8	12	16	20	24	28	32	36
```

```
5    |    5    10    15    20    25    30    35    40    45
6    |    6    12    18    24    30    36    42    48    54
7    |    7    14    21    28    35    42    49    56    63
8    |    8    16    24    32    40    48    56    64    72
9    |    9    18    27    36    45    54    63    72    81
```

Project Name	Assign3_LoopPractice
App Name	LoopPractice

Rubric	
Loops 1-5	**75**
Menu for Part 6	**15**
Averager Loop	**30**
dispAuthor (part 8)	**15**
playSound (part 9)	**15**
Probability (part 10)	**40**
Comments	**10**
TOTAL	**200**

Rubric for Bonus	
Scalable Pyramid	**15**
Multiplication Table with nested loops	**8**