South Louisiana Community College
ASDV 1220, Programming Fundamentals
Lab 12

Work with same partner unless your instructor reassigns you. ALTERNATE the roles of Coder, Navigator in each problem.

Learning Objectives

After completion of this lab, you should be able to

1. Understand keywords continue, break
2. Understand nested for loops
2. Understand nested while loops

Create project Lab12

Problem 1

Create a class **ForNested1**, write the code as shown below that prints the indexes of the outer and inner loop. Consider the two nested loops a table generator where the outer loop generates a row number **i** starting at 0 and the inner loop generates the columns **j** of the row **i**, again starting at 0.

(i, j) = (0, 0) (0, 1) (0, 2) (0, 3)
(i, j) = (1, 0) (1, 1) (1, 2) (1, 3)
(i, j) = (2, 0) (2, 1) (2, 2) (2, 3)

```
1   package lab12;
2
3   public class ForNested1
4   {
5       public static void main(String[] args)
6       {
7           for ( int i = 0; i < 3; ++i)
8               {
9               System.out.print("(i, j) = " );
10              for ( int j = 0; j < 4; ++j)
11                  System.out.print( "(" + i + ", " + j + ") ");
12              System.out.println();
13              }
14      }
15  }
```

Problem 2

Create a class **WhileNested2**, write code as shown below. Again it prints the indexes of the outer and the inner loop. The only difference between this problem and Problem 1 is that in this problem the outer for-loop was replaced by a while-loop.

Line 6: Condition
Line 7: Initialization of condition
Line 15: Update of the condition

```
1   package lab12;
2
3   public class WhileNested1
4   {
5       public static void main(String[] args)
6       {
7       int i = 0;
8       while ( i < 3 )
9           {
10          System.out.print("(i, j) = " );
11          for ( int j = 0; j < 4; ++j)
12              System.out.print( "(" + i + ", " + j + ") ");
13          System.out.println();
14
15          ++i;
16          }
17      }
18  }
```

Problem 3

Create a class **WhileNested3**, write code to replace the for-loop of Problem 2 with a while-loop. In other words, generate the indexes i,j as in Problem1 and Problem 2 but this time use 2 nested while-loops.

Problem 4

Create a class **NestedForPatternA**. Use nested for-loops that display the following pattern:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

Problem 5

Create a class **NestedForPatternB**. Use nested for-loops that display the following pattern:

```
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Problem 6
Create a class **NestedWhilePatternC**. Use nested while-loops that display the following pattern:

```
      1
    2 1
  3 2 1
4 3 2 1
5 4 3 2 1
6 5 4 3 2 1
```

Problem 7
Create a class **NestedWhilePatternD**. Use nested while-loops that display the following pattern:

```
1 2 3 4 5 6
  1 2 3 4 5
    1 2 3 4
      1 2 3
        1 2
          1
```

Problem 8
Create a class **NestedForPyramid**. Write nested for-loops that print the following output:

```
                    1
                  1 2 1
                1 2 4 2 1
              1 2 4 8 4 2 1
            1 2 4 8 16 8 4 2 1
          1 2 4 8 16 32 16 8 4 2 1
        1 2 4 8 16 32 64 32 16 8 4 2 1
      1 2 4 8 16 32 64 128 64 32 16 8 4 2 1
```

Problem 9
Create a class **NestedWhilePyramid**. Write nested while-loops hat print the same output as problem 8.

Problem 10

Create a class **TestBreak1** *as shown below that uses a while loop to execute 20 times and* ***breaks*** *when the variable sum is greater or equal to 100.*

```
3    public class TestBreak1
4    {
5
6        public static void main(String[] args)
7        {
8            int sum = 0;
9            int number = 0;
10
11           while (number < 20)
12             {
13
14               if (sum >= 100)
15                 {
16                   break;
17                 }
18               number++;
19               sum += number;
20             }
21
22           System.out.println("The number is " + number);
23           System.out.println("The sum is " + sum);
24       }
25   }
```

Problem 11

Create a class **TestBreak2** *which replaces the while loop of problem 9 with a for loop.*

Problem 12

Create a class **TestContinue1** as shown below that uses a while loop to execute 20 times and *continues* when the variable number is 10 or 11. The program has an infinite loop. Fix it.

```
3    public class TestContinue1
4    {
5
6        public static void main(String[] args)
7        {
8            int sum = 0;
9            int number = 0;
10
11           while (number < 20)
12             {
13
14               if (number == 10 || number == 11)
15                 {
16                   continue;
17                 }
18               number++;
19               sum += number;
20             }
21
22           System.out.println("The sum is " + sum);
23       }
24   }
```

Problem 13

Create a class **TestContinue2** which replaces the while loop of problem 11 with a for loop.

Problem 14

Create a class **Palindrome1** as shown below. The class uses a while loop to determine whether a word (string) entered by the user is a palindrome. For example, "mom" , "dad" are palindromes. Set a breakpoint at line 24 and use the debugger to understand how the loop works for strings "mom" and "aba".

```
1   package lab9;
2   import java.util.Scanner;
3   public class Palindrome1
4   {
5
6       public static void main(String[] args)
7       {
8           // Create a Scanner
9           Scanner input = new Scanner(System.in);
10
11          // Prompt the user to enter a string
12          System.out.print("Enter a string: ");
13          String s = input.nextLine();
14
15          // The index of the first character in the string
16          int low = 0;
17
18          // The index of the last character in the string
19          int high = s.length() - 1;
20
21          boolean isPalindrome = true;
22          while (low < high)
23            {
24              if (s.charAt(low) != s.charAt(high))
25                {
26                  isPalindrome = false;
27                  break;
28                }
29
30              low++;
31              high--;
32            }
33
34          if (isPalindrome)
35            {
36              System.out.println(s + " is a palindrome");
37            }
38          else
39            {
40              System.out.println(s + " is not a palindrome");
41            }
42      }
43  }
```

Problem 15

Create a class **Palindrome2** which replaces the while loop of problem 13 with a for loop.