|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LEARNING PROFILE FOR ReverseInputNumbers | | | | | |
| *Name* | *:* | *Tyler Lucas* | *Due Date* | *:* | *N/A* |
| *Student ID* | *:* | *3305203* | *Submission Date* | *:* | *N/A* |

# Problem Statement

Textbook example program.

# Description of the Code

Reads the numbers input by the user and then prints them in the reverse of the order in which they were entered. Assumes that an input value equal to zero marks the end of the data.

# Errors and Warnings

Version 1.0

No errors.

Version 1.1-1.2

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Errors / Warnings** | **Details** | **How I solved them** |
| 1 | \*\*\* Error in input: Integer value not found in input. | Was attempting to catch a standard Exception (NumberFormatException), but Eck’s TextIO didn’t throw it. | Stopped using TextIO. |
| 2 | Exception in thread "main" java.lang.IndexOutOfBoundsException: Index: 3, Size: 1  at java.util.ArrayList.rangeCheck(ArrayList.java:653)  at java.util.ArrayList.get(ArrayList.java:429)  at ReverseInputNumbers.main(ReverseInputNumbers.java:43) | Was attempting to use Java 8 List and Iterator features. Index went out of bounds. | Lots of research. |
| 3 | Incorrect behaviour. | See Discussion. | Changed regex expression. |

# Sample Input and Output

Version 1.0

Enter up to 100 positive integers; enter 0 to end.

? 3

? 7654

? 3

? 987654

? 93816

? 4

? 2

? 3

? 4

? 4

? 798765

? 0

Your numbers in reverse order are:

798765

4

4

3

2

4

93816

987654

3

7654

3

Version 1.2: incorrect behaviour

Enter a list of comma-separated integers.

> 1, 234, 5,

Your numbers in reverse order are:

5, 4, 3, 2, 1

Version 1.3

Example 1: end list with comma and [Enter]

Enter a list of comma-separated integers.

Press [Enter] twice to finish.

> 12354,56432,2345,2,5,7,-20,

Your numbers in reverse order are:

-20, 7, 5, 2, 2345, 56432, 12354

Example 2: end list with integer and 2x[Enter]

Enter a list of comma-separated integers.

Press [Enter] twice to finish.

> -213482,0,235123,3,3,3,3,1231

Your numbers in reverse order are:

1231, 3, 3, 3, 3, 235123, 0, -213482

# Discussion

Interesting.

Version 1.2 (version 1.1 was a bust) used several more advanced Java objects. Here is the commit snippet:

Fancy version. Uses List<Integer> = new ArrayList<>() instead of an array of int's. Uses Scanner instead of TextIO. Uses Collectione reverse() algorithm to reverse list. Uses Iterator<Integer> to output. I don't understand it 100%, but enough to get it right.

I didn’t understand the regular expression in the useDelimiter(Pattern) method (it was “(, \*)\*”), and am not experienced with regex in general, so I had to do some extra research to fix it. Wound up with a handy regex cheat sheet from AddedBytes.com. I couldn’t figure out a pattern to detect a single carriage return ([Enter] key or newline) after an integer, so I changed the instructions to end the input with two [Enter] keypresses[[1]](#footnote-1) and used the pattern ",|\\n" (match comma or newline character).

1. System.out.println( "Press [Enter] twice to finish." ); [↑](#footnote-ref-1)