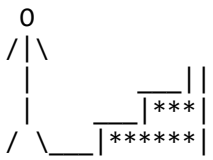
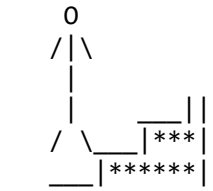
CmpE 150 Introduction to Computing, Fall 2019

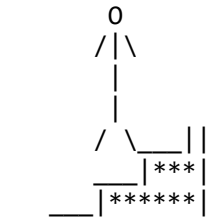
Assignment 1 – Due: 27/10/2019, 23.59

You will write a Java program to implement the ASCII stickman climbing the stairs below using two parameters **stickmanHeight** as the height of the stickman and **stairHeight** as the height of the stairs. One example run is given below, other example outputs are provided as .txt files. Please read the assignment description and check out the example outputs before beginning to write any code.

Example Run: **stickmanHeight** = 5, **stairHeight** = 2







Please make sure you follow these rules in your implementation:

1. Your program should have at least two static methods in addition to your main method. Try to write your program as modular as possible (without overusing methods).
2. You are not allowed to generate the entire picture or any single line using printing statements such as System.out.print or System.out.println, with the exceptions:

- The head of the stickman (“ O “),
- The arms and torso of the stickman (“/|\”), and
- The legs of the stickman (“/ \”)

For example, you cannot have a statement that says:

System.out.println(“___|***|”);

It was intended to be generated using for loops.

3. Assume that **stickmanHeight** will be greater than **stairHeight** +2. (In other words, **stairHeight** + 2 < **stickmanHeight**) Also, **stairHeight** will always be greater than or equal zero.
4. You are not allowed to use statements that we haven’t learned in class as of 15/10/19 (such as while loops, arrays, if statements and so on).
5. Try to minimize the number of for loops you use.
6. We have given you a part of the code for the main method in arguments.java file. This code runs your program with arguments given to **stickmanHeight** and **stairHeight** respectively. Copy this code to your own .java file. To test and run your program with arguments, click Run -> Run Configurations -> Arguments -> Program Arguments. Enter two integers here, and then click run at the bottom of the window. These two integers represent **stickmanHeight** and **stairHeight** respectively.

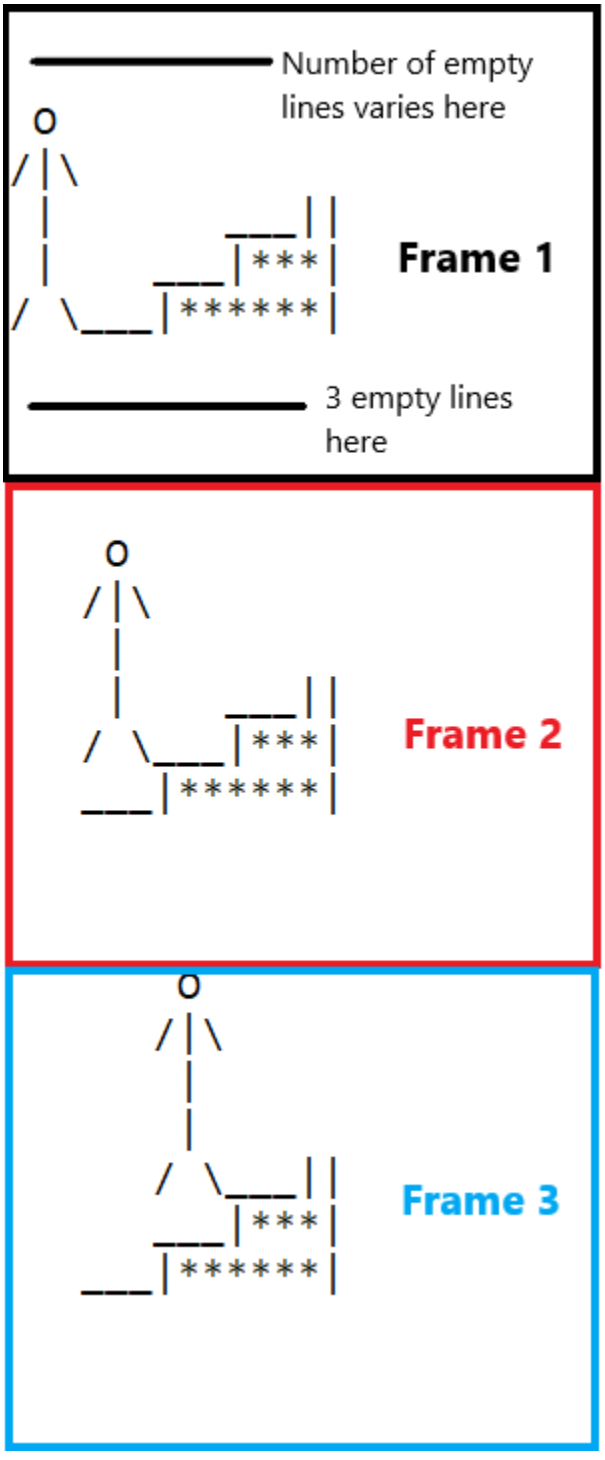
Implementation Details:

- Think of the stickman as a rectangle with dimensions 3 x **stickmanHeight**. The parts with the head or just the torso (“ | “) of the stickman should have one blank space around them.
- Stairs are formed of 3 parts:

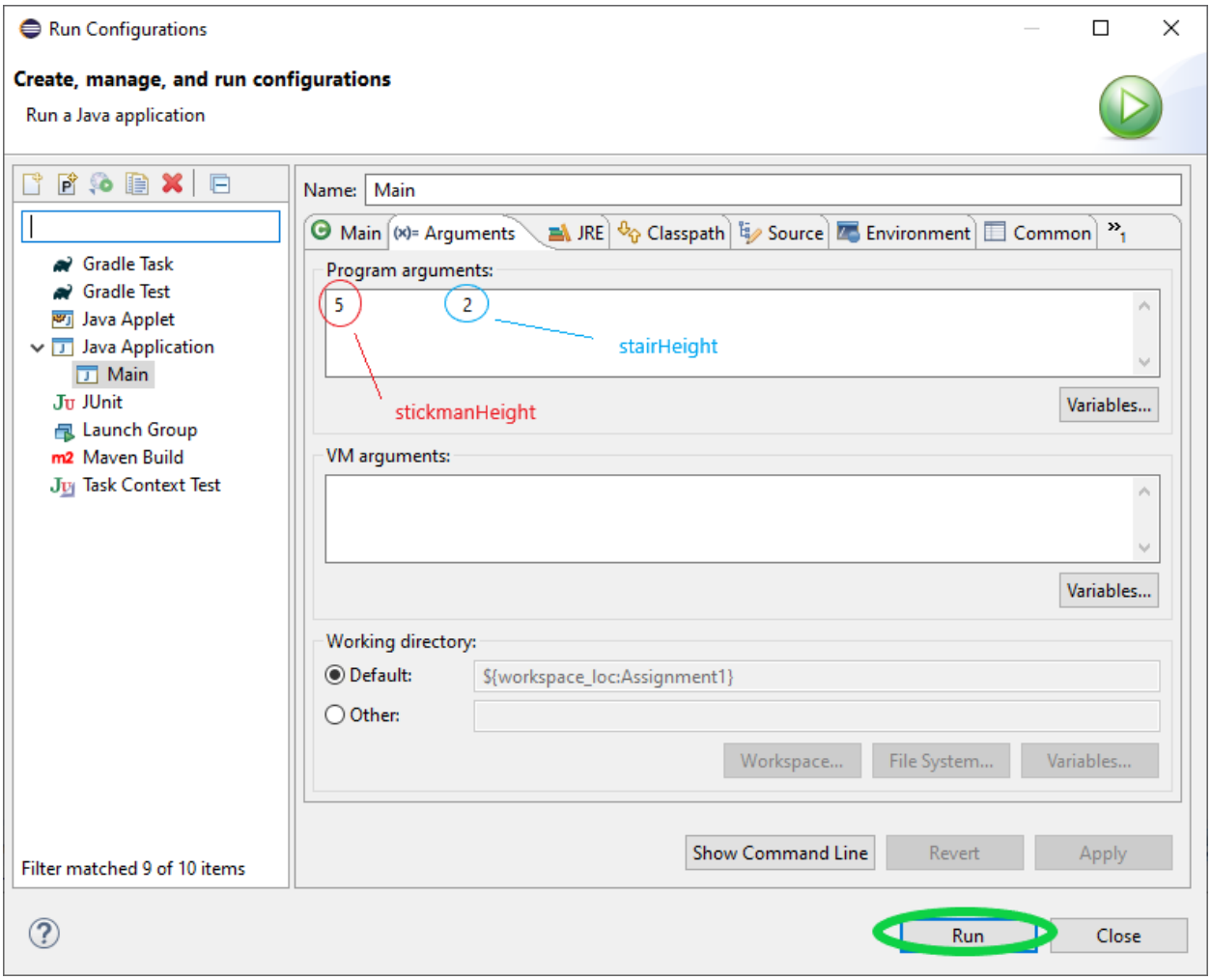
  1. The floor part which is fixed (“___”),
  2. The beginning and the end of the stair (“|……|”),
  3. The filling pattern between the beginning and the end of the stair (******).

- The amount of blank spaces between the stickman and the stairs, and the filling pattern inside the stairs (the number of stars) are both multiples of 3. The amount of blank lines after a single frame is outputted is exactly 3.
- The amount of blank lines at the beginning of each output frame is also important. Each frame should have the same total height, which has the relation **stickmanHeight** + **stairHeight**.

- A frame is each printed pattern, like every single step the stickman takes. See the image below.



- The appearance of the output and the values of parameters may be confusing, so interpret **stickmanHeight** and **stairHeight** as starting from the bottom of the figure and assume each line has height of 1 unit. For example, in the below example **stickmanHeight** = 5 and **stairHeight** is 2, not 3.



Submission: You will submit a project report and your code over Moodle.

Project report should consist of five sections. These are:

1. Problem Description: In this section, you should describe the problem in your own words.
2. Problem Solution: In this section, you should specify the concepts (methods, for loop, etc.) that you use in your program. Explain each one (i.e. why you need it, what you accomplish by using it, so on.). Report how many for loops you use.
3. Implementation: This section will include your whole code with comments. You need to pay attention to indentation in order to improve readability.
   - Do not forget to explain each variable that you use (i.e. int count = 0; // count is the number of items).
   - Before each method, specify what the method does (i.e. /* This method … */)
4. Output of the program: A screenshot of your program output should be put in this section. Two example runs are enough.
5. Conclusion: You should evaluate your work here. State whether you have solved the problem correctly. If not, state what is missing, what could have been improved, and so on.
   Your .java file should be named with your initials and your student number together (e.g., OS2013800027). If you have Turkish characters as your initials, please change them to non-Turkish. (Example: ÖS2013800027 should be OS2013800027) You will submit these over Moodle as a single zip file where the file name is your student number. Your zip file should consist of your .java file and your report in .doc or .pdf format. Do not use any Turkish characters in your code, class/variable names, or .java file names.

Partial Submission: If you cannot generate the picture above, you should still submit your code as well as your report. Try to generate most of the picture. In your report, explain which parts you can generate and which parts you cannot. Partial credit will be given if your submission is able to produce the correct output for **stickmanHeight** = 7, **stairHeight** = 3 case. If you can produce the correct output for multiple configurations but not all, you will still get partial credit, and if you only have minor errors like blank space count or empty line count is not exactly as specified, you will still get partial credit provided the

rest of the output is accurate.

Late Submission: Late submission is possible with penalty = -10*(daysLate)^2.

Evaluation Procedure: We will evaluate your code based on the program output for many test cases. We will consider the number and locations of blank spaces and empty lines as well as the other characters in your output. You can check whether your output is correct by copying the output from the provided example output files, and copying your output to an online text diff checker, like https://www.diffchecker.com for example.

More example runs:

Run #1: **stickmanHeight** = 3, **stairHeight** = 0

```
 O
/|\
/ \___||
```

Run #2: **stickmanHeight** = 4, **stairHeight** = 1

```
 O
/|\
 |    ___||
/ \___|***|
```

```
   O
  /|\
   |
  / \___||
  ___|***|
```