**WS2 - Flow Control workshop**

The aim of this workshop is to practice writing conditions & loops. We will do so by implementing methods from Math library, printing various geometric shapes that have repetitive patterns, focusing on both “while” and “for” loops.

**Important notes:**

1. The relevant knowledge of the students is up to the end of week 2 (Lecture 2-2, recitation 2 - If, else, for, switch while and do-while loops, concat).
2. This is the first "serious" programming workshop - try to simplify the terms and not to use / mention functions or any other term that has never been discussed in class before. And it is divided into 4 main parts
   1. Conditions
   2. Loops
   3. Nested Loops
   4. Hard questions

The first 2 consists of 4 parts (each):

1. Intro + discussion
2. 1 example together
3. Few exercises to do alone
4. Cover solution
5. The latter 2 should be led by you but allow them to share inputs.
6. It is recommended to start the workshop with a reminder of the relevant notes.   
   consider s as a String of 0 or more characters:
   * 1. System.out.print(s); - Prints s at the cursor position , and moves the cursor just after s.
     2. System.out.println(); - Moves the cursor to the beginning on the next line
     3. System.out.println(s); - Prints s at the cursor position,

and moves the cursor to the beginning on the next line.

1. During the whole workshop, it’s recommended to draw up solutions together with the students before writing ‘real code’, and encourage them to do so when writing code at home.
2. Git repo for skeletons will be provided: <https://github.com/cs-intro-reichman/WS2-2024-Skeleton> , this is a good way to practice git more.
3. The solutions repo will be provided but will be published on Friday. And they are also in the folder.
4. Encourage the students to check their solutions by creating tracking tables - that could help them understand better what happens in every single step of the execution.
5. This workshop takes a long time and most likely you won’t be able to cover all the examples. It is important to highly recommend the students to continue the rest at home.
6. **Conditions Introduction (10 minutes)**

Start the workshop with the following example:

int count = 50;

if (count < 100) {

System.out.println(“less than 100”);

}

Points for discussion at this point:

* What does conditions do? And this one in particular
* What happens if the condition is changed to “count <= 100”?

(discuss the Difference between ‘<’ and ‘<=’ in conditions)

* Discuss else, else if, trenary operator (boolean ? trueOpt : falseOpt)
* Discuss switch case
* Discuss the different between the following codes

| if (count < 80) {  System.out.println(“less than 80”);  }  if (count < 100) {  System.out.println(“less than 100”);  } | if (count < 80) {  System.out.println(“less than 80”);  } else if (count < 100) {  System.out.println(“less than 100”);  } |
| --- | --- |

**2. Solve together (5 mins):**

* A condition which prints the absolute value of a number given from the user without using Math.abs(int a)

**3. Practice (10 min):**

Ask students to write the following conditions:

* A condition which checks if a number given from the user is odd or even and then print it to the user.
* A condition which prints the if a number given from the user is 2 digits, without using any String operations
  + /\*\* note: args[0].length == 2 is not what we want, we aim for -> 10 <= input && input <= 99 \*\*/
* A condition which checks if the first char of the first argument is digit.
  + We aim for “0123456789”.indexOf(args[0]) != -1), in week 3 they will see the more simple one with ASCII.
* A switch case condition which receives the name of the language, and you will return a greeting for that language (Must include at least 5, will be provided, English -> Hello, Hebrew -> Shalom, french -> Bonjour, Italian-> Caio, german -> Guten Tag), add default to be English.

**4. Cover Solutions (7 minutes)**

* Solve each exercise.

**5. Loop intro Introduction (10 minutes)**

Start the workshop with the following example:

int count = 0;

while (count < 10) {

count = count + 1;

System.out.println(count);

}

Points for discussion at this point:

* What does loops do?
* What does this loop do?
* What happens if the condition is changed to “count <= 10”?

(discuss the Difference between ‘<’ and ‘<=’ in loops)

* Discuss **break; continue;**
* What happens if we switch between the 2 lines in the body of the loop?
* **Discuss**

int count = 50;

while (count < 10) {

count = count + 1;

System.out.println(count);

}

**6. Solve together (5 minute):**

* A loop which checks if a string is valid number, **assume that the number is non negative**. And string.length() > 0;

**7. Practice (10 min):**

Ask students to write the following loops:

* A loop which prints all numbers between 10 and 1, in **descending** order.
* A loop which prints all **even** numbers between 1 and 20.
* A loop which prints the following sequence: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 (first 10 powers of 2).
* Expand the solution you did together, loop which checks if a string is valid number. **(negative are allowed)**

**8.Cover Solutions (8 minutes)**

* Solve each exercise.

**9. Geometric shapes -> Nested Loops (10 minutes):**

Introduce the following loop, which prints a line of asterisks.

int count = 0;

while (count < 10) {

count = count + 1;

System.out.print(“\*”);

}

Ask students to write a program which takes a String ‘s’ and an int ‘n’ (as command line arguments) and prints ‘s’ consecutively ‘n’ times, in the same line.

Here is an example with a string ‘Happy’ and n = 5, the output will be:

HappyHappyHappyHappyHappy

Points for discussion at this point:

* Difference between ‘print’’ and ‘println’.
* Use of loops to concatenate strings.
* Alternative use for this solution - ‘for’ loop.

**Left-justified triangle:** We will now draw a left-justified triangle. We will need a program which takes an int ‘n’ and prints a triangle of the following form with ‘n’ lines.

Here is an example with n=3:

\*

\*\*

\*\*\*

**Rectangle**:Discuss how to write code that draws a rectangle. The code takes 2 ints ‘n’,’m’ which are the height, width of the rectangle.

Here is an example with n=2, m=6:

\*\*\*\*\*\*

\*\*\*\*\*\*

Points for discussion regarding the ‘for’ solution:

* What would have happened if we chose the same iteration index for both loops?

Discuss the results of creating variables with the same name in a common scope.

* Creating a first sketch of the solution - and deciding which loop would be the best solution for the case (while/for and why).

**10. VERY HARD QUESTIONS (IF YOU HAVE TIME -> 15 MIN)**

Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

| Symbol | Value |
| --- | --- |
| I | 1 |
| V | 5 |
| X | 10 |
| L | 50 |
| C | 100 |
| D | 500 |
| M | 1000 |

For example, 2 is written as II in Roman numeral, just two ones added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

* I can be placed before V (5) and X (10) to make 4 and 9.
* X can be placed before L (50) and C (100) to make 40 and 90.
* C can be placed before D (500) and M (1000) to make 400 and 900.

1. Given a roman numeral as string, convert it to an integer.
   * Assume that the string is UPPER CASE
   * Assume that the string is a valid roman number
   * Assume that the value of the roman number is in the range of 1-3999

<https://leetcode.com/problems/roman-to-integer/description/> -> **Question level: Easy**

1. Given an integer, convert it to roman numeral.
   * Assume that the number is in the range of 1-3999

<https://leetcode.com/problems/integer-to-roman/description/> -> **Question level: Medium**