

# Appendix to the assignment.

Version 1.0

## Objective of the document.

This document aims to provide some guidance on two aspects of the assignment:

- How to upload the assignment.
- How to comment the code.

## How to upload the assignment.

The assignment should be contained in a single document.

One can decide to put everything in the script file (including comments and description on how to run the script), or in a docx/pdf document.

An explanation on how to upload the submission is available at the link below:

[Turniting assignment upload](#)

If you have any issue with the video, please send an email.

## How to comment the code

The code should be commented in a way that makes it clear that the code is fully understood by the student (rather than explaining the code to the instructor). It should describe:

- “what” the purpose of the line of code is,
- “what” do parameters mean, if any, and if not already shown.
- “why” one is taking a specific approach.

An example of code comment is described below:

```
@echo off
rem I disabled the echo, so that commands are not echoed back in the console.

rem General comment on the algo-function
rem this script allows to count up from 10 down to 0. To do so we use a loop.
rem at every run of the loop we decrement a variable and print the value of the counter
rem on the screen. When the counter reaches 0 the loop is not executed and the loop exits

rem setting the variable cnt to 10, I am using the parameter /a as I am going to deal with
rem numerical expressions (I will decrease cnt). My strategy is to decrement cnt and print
rem its value at every loop iteration.
set /a cnt=10
```

rem As I need to perform 10 loops I now set a label. The code will jump to this label until 0 is reached creating therefore a loop.

:loop

rem This is the exit condition for the loop, I am in the loop until cnt becomes lower than 0

if %cnt% geq 0 (

rem this is the block code in the loop (in bash a block of code is contained in parentheses).

rem at every iteration I perform the actions below.

rem print on the screen the value of the variable cnt at this iteration.

echo countdown %cnt%

rem decrement cnt preparing it for the next loop (again using option /a for set but no need

rem to write that again, as I already demonstrated I understand that).

rem set /a cnt-=1

rem we use goto to jump to the label :loop that was defined earlier, i.e. the start of the loop.

goto :loop

)

rem I am outside the if block, the goto did not execute, so cnt is now lower than 0.

rem I print on the screen countdown over.

echo countdown over

rem I wait for the user to press a key to terminate the script. This is a good idea because if one launches the script clicking on it, it would close after finishing not giving time to the user to see the result on the screen

pause