Demo 1 / 15.9

Please return your answers by using the web application NettiDemoWWW
https://www.mit.jyu.fi/demowww/ohj1/ on Monday by 12.00 o'clock at the latest. Bear in mind that this is a strict deadline for returning your answers! You can return your answers separately or all of them at one time and modify them until given deadline.

Note: links are in finnish at the moment!

Tasks

1.   Your task is to make a directory structure mentioned below on your computer (or at the computer lab on drive U) using command line. Directories are used for organizing your answers and are handy during demonstrations. Hints from the Wiki:https://trac.cc.jyu.fi/projects/ohj1/wiki/komentorivi. The indentions below show the structure:

```
courses
  ohj1
    demos
      demo1
      demo2
```

Give a try to the following command:

```
dir \courses /s    ( on Linux ls -r DIRECTORY )
```

In the case of a successful task one should see following lines:

```
Directory of U:\courses
Directory of U:\courses\ohj1
Directory of U:\courses\ohj1\demos
Directory of U:\courses\ohj1\demos\demo1
Directory of U:\courses\ohj1\demos\demo2
```

For returning the answer for this task one has to make a file teht1.txt as follows:

```
cd \courses\ohj1\demos\demo1
dir \courses /s >task1.txt
```

and have a look what was the result:

```
type task1.txt
```

The instructor will check during the computer lab tutorial that the directory structure is made the right way and the tutorial is the best place to ask if there are problems while making directories or text files. The idea of directories, files and how they work is very important considering this course!

2..   **How to use text editor:** In this task you have to use timer to measure how long does it take to write a text file. The file should be written using guidelines mentioned below. Cheating is not allowed! While this task is made tongue-in-cheek it is still serious competition.

In the beginning you should read guidelines carefully, memorize them and start with a blank text file. Only one text editor and your own hands are allowed! (one example of a real text editor is ConTEXT. Word or another "Office" type of a program is not considered as a text editor) All the features like copy, paste, find, replace and such available in the text editor are free to use. There is one rule more: the mouse is banned.

Start your timer as soon as you type in your first character and stop the timer when the last character is typed. The content of the text file should be:

```
1) At first you should type in exactly one hundred lines a text
   "I am practicing for making a text file!"

2) Exactly one line with 80 times a minus sign '-' and nothing else

3) Then forty lines with one number in each. The number should start
   from one (01) and increase by one every time.

   Here is an example:

      01
      02
      03
      ...    and in the end:
      39
      40

4) And the last line has to have exactly 80 times a minus sign '-'.
```

Stop the timer and save the file as task2.txt. Now, make a new line in the same text file and in this new line you type in the time it took to complete the task. The format of the time is mm:ss where mm stands for minutes and ss for seconds. Please write a brief description after the time as well. In this description one should tell how the task was done. This file acts as an answer for this task. Remember to save and give a proper name. The result will give you a picture how efficient you are as a writer in comparison to your friends in the course. Repeating this task in the end of this course will help you to notice improvement.

My time was 01:45 so that making this task doesn't take that much time after all. I made a mistake while typing 18 two times and forgot the number 19 completely so I had to correct my error. The guidelines are strict: the timer stops when the task is completed successfully without misspelling.

3   **Getting used to tools**: Please make sure that you have all the tools needed for completing the course Programming 1 and you are able to use them as well:

```
A sophisticated text editor
Java 6 SDK ( or newer; has to be SDK )
Command line
Internet connection
SSH telnet client ( e.g. SSH secure Shell, PuTTY or equivalent )
```

These tools are ready to use in the computer labs at agora. The programs are available in the Internet and can be downloaded freely but personal guidance is unfortunately not possible. One good way to learn is to ask friends or to find the solution by self. The course's Wiki might be handy as well as the mailing list. Do not hesitate to ask!

If one works at home it is advisable to install SDK because it will be needed anyway. In the beginning the use of SSH telnet client is fine if there has been problems while installing tools needed in this course. One way to edit text files directly on the server is to use a user-friendly but yet very restricted text editor called nano.

Try this in practice:

```
Save the example program Hello.java from the Internet to your
demodirectory. Compile and run the program with command line.

Hint:
javac Hello.java
java Hello
```

The answer for this task is the file Hello.java with your name in the beginning of the file. If you work as a group the name of every participant is needed.

4. **Formats of information storage:** Oh dear. Juha broke his USB memory stick. The stick contains only small shards of data and no information of their order. It had a lot of MP3 files and other stuff... Juha knows that one vital file was a text file written in capital letters. The bit strings below look promising but what do they mean:

```
01000111  01010010  01000001  01000100
01010101  01001110  00100000  01001010
01001111  01001000  01000100  01000001
01001110  01010100  01001001
```

Let's make it easier; the following byte-long strings are given as base ten (decimal) numbers and, to give an example, one of the numbers has been converted into a charater as per the table below:

```
71  82  65  68  85  78  ... fill in the rest ...
        =A
```

Decode the rest of the characters. What is the text saved at that point in the USB stick? By how many millimeters has Juhas finger missed one of the keys? He has the same kind of keyboard as you have in front of you. Return a file labeled task4.txt that begins with your name and that also has the decoded message and the amount of millimeters of the mispress.

Juha knows that the format he used uses the following codes (pieces of ASCII code) to represent the characters:

```
Decimal        Binary          Character
------------------------------------------
  32           00100000        [space]
  33           00100001           !
  46           00101110           .
  65           01000001           A
```

```
66          01000010        B
67          01000011        C
68          01000100        D
69          01000101        E
70          01000110        F
71          01000111        G
72          01001000        H
73          01001001        I
74          01001010        J
75          01001011        K
76          01001100        L
77          01001101        M
78          01001110        N
79          01001111        O
80          01010000        P
81          01010001        Q
82          01010010        R
83          01010011        S
84          01010100        T
85          01010101        U
86          01010110        V
87          01010111        W
88          01011000        X
89          01011001        Y
90          01011010        Z
```

5. **Console-based Java programs**: First off, copy the file `Hello.java` from the third task into a file labeled `Me.java`:

    ```
    copy Hello.java Me.java   (Linux: cp Hello.java Me.java)
    ```

    Open `Me.java` into an editor (for example: `ct Me.java`) and change the name of the class `Hello` into `Me`. Then modify the program so it prints your name, the manufacturer of your cell phone and the name of your operator, all on separate lines. Compile and run the program. Return `Me.java`

6. **Graphic Java programs:** This task must be performed on an actual workstation and cannot be done via SSH. First, copy the library `Graphics.jar` as per instructions given in the Wiki: https://trac.cc.jyu.fi/projects/ohj1/wiki/graphics. Create and run a program like the example `SimpleGraphics` that draws a line and a circle. Modify the program so that it uses lines to draw a square around the circle. Save as Square.java and return the file.

B1. Download and install Alice as per the instructions in the Wiki:https://trac.cc.jyu.fi/projects/ohj1/wiki/Alice. Go through the first tutorial (ice skating).

GURU-tasks

G1-2 Write a Java program that prints all the integers <= 1000 that have a square root that is also an integer. If absolutely necessary, "old" students get to do this in C or C++. TDD: Before coding write down numbers and their square roots by hand.

Getting points from doing tasks

The maximum of tasks is 6/demo (regarding percentages). If there are more tasks, the rest are either bonus or GURU tasks. These kinds of tasks are for putting some points into the reserve. The GURU tasks are also designed so people who already have some grasp of the matter could have challenges. Nevertheless the absolute maximum of points for each demo is eight (8). So even if you mark all nine as done this time around, you will only get points for eight (8).

Using Old or Someone Elses Answers

There might still be answers from previous years and if you get caught using those or your friends' answers the penalties will be comparable to those for cheating in exams! The most important thing on this course is to learn and not fishing for points or copying. You can get the copies ready-printed at the demos.

Working as a group is allowed but remember to include the names of everyone in the group – for example in the comments of the program.

Computer lab tutorials

Check the time and place of your computer lab tutorial in Korppi.

You can find the tasks in the Wiki: https://trac.cc.jyu.fi/projects/ohj1/wiki/paate

WWW pages

Remember the course has a WWW page:

    http://www.mit.jyu.fi/vesal/kurssit/ohjelmointi1/2008/