Homework 1

Submission guidelines

The following homework submission guidelines apply to this and all future homework assignments in this course. Please follow these directions to prepare your submission:

- All code should be written and tested on the `cslab` machine. This guarantees that our Teaching Assistants will be able to compile and execute your programs in the same environment where you tested them.

- Create a separate directory for each homework assignment, and a separate subdirectory for each question in the assignment. For example, Question 2 of Homework 1 should be implemented in a directory named `~/hw1/q2`, where symbol `~` represents your home folder.

- Make sure that all of your code compiles correctly on `cslab` before submission. Any code that produces compilation errors is assumed to not have been tested, and cannot be considered as a valid submission. You should be able to successfully run command `javac *.java` on the directory where you implemented each question.

- Once you have verified the correctness of your program, remove all compiled `.class` files with command `rm *.class` (please be very careful when running this command, as a small typo could lead to a rather tragic and unrecoverable loss of information). Your submission should not include any compiled `.class` files, but only `.java` source files.

- Create a ZIP package for each question. A ZIP file should include the directory associated with a given question, plus all of its files. You can do so by first making the main homework directory your current working directory (e.g., with command `cd ~/hw1` for this homework), and then creating the ZIP file (e.g., with command `zip -r q1.zip q1` for Question 1).

- After you complete all assignments, import all ZIP files (`q1.zip`, `q2.zip`, …) into your local machine. If you are on Windows, use WinSCP for this purpose. If you are on MacOS or Linux, use command `scp` as described on the support material of Unit 1.

- Upload all ZIP files on Canvas as your homework submission. Once submitted, download them one by one and verify that their content includes all the material requested in the submission, and that no mistake was made while creating the ZIP files.

The recommendations bellow will minimize the chances of invalid or late homework submissions, so please follow them carefully:

- Verify that your submission is correct by downloading and checking the content of every file previously uploaded. Make sure that the submission format is exactly as requested in the homework manual. Only those answers submitted in the correct format will be graded.

- Start your submission long before the homework deadline in order to give yourself extra time in case of a submission problem. Remember that homework deadlines are strict, and late homework is not accepted under any circumstances.

- Please direct any homework-related questions to our course teaching assistants, whose contact information can be found on the Syllabus document. They will answer your questions as soon as possible, but they may need up to 24 hours to do so. Please do not expect rushed answers before homework deadlines. Plan accordingly.

Question 1 (5 pt.)

Write a Java program with the following two functions:

- A private, static function named `Factorial()`. This function should take an integer value `n` as an argument and calculate its factorial (`n`!), which should be provided as the return value of the function. The factorial of a number *n* is calculated as $n \times (n-1) \times (n-2) \times \ldots \times 3 \times 2 \times 1$.
- A public, static function `main()` containing the main program. This function should first ask the user to enter an integer value. The program should read this value from the keyboard, calculate its factorial by invoking function `Factorial()`, and print the result.

Wrap this code in a class named `Test` within a file named `Test.java`. Create a package named `q1.zip` containing directory `q1`, and submit it on Canvas.

Question 2 (5 pt.)

Write a Java project composed of two classes, each written in a file with the same name plus extension `.java`:

- Class `Person`, representing a person with the following properties:
  - A private string field called `name`.
  - A private integer field called `age`.
  - A constructor that takes the person's name and age as arguments, and assigns them to the corresponding private properties.
  - A public function called `Print()` with no arguments, which prints the person's name and age with the following format:

    ```
    Name: John Smith
    Age: 25
    ```

  - A public function called `IsUnderage()`, which takes no arguments and returns a Boolean value indicating whether the person's age is less than 18.
- Class `Test`, containing the main program in a public, static function called `main()`. The function should perform the following actions:
  - Ask the user to enter name, and read it from the keyboard.
  - Ask the user to enter an age, and read it from the keyboard.
  - Create an instance of `Person`, and initialize it with the given name and age.
  - Print the person by invoking function `Print()` on the new object.
  - Print whether the person is underage by invoking function `IsUnderage()`.

The following output is an example of the execution of the program:

```
Enter name: John Smith
Enter age: 19
```

```
Name: John Smith
Age: 19
Underage: false
```

Create a ZIP file named `q2.zip` containing directory `q2` and all `.java` files in it. Upload this file as part of your homework submission.