

Exercise – 4

Published on: 15.05.2022

Deadline: 22.05.2022 – 1:59pm

Submission Instructions:

- Download the handout zip archive here:
<https://sync.academiccloud.de/index.php/s/kKj7nvnwb1W1Pkd>
- Programming language: Python 3.10
- You can use this Virtual Machine for a pre-installed environment: [Link](#)
(Password: 5cnN59dzVEm5atc)
- Please watch the short tutorial video summarizing how to do the python programming exercises.
- General Instructions
 - Unzip the handout zip archive
 - The handout contains a Pipfile. You can install the dependencies for the exercises by running ``pipenv install``. (You might have to install [pipenv](#) and [pyenv](#) first)
 - Activate the python virtual environment using ``pipenv shell``.
 - In the first sub-directory, you will see two files:
 1. solution.py
 2. driver.py
 - **You only need to modify the „solution.py“ file.** More detailed instructions on where you need to insert your code can be found in this file. The automated grading mechanism can grade your solution only if you follow the structure provided in the „solution.py“ file.
 - Process all available data samples and build up a data structure according to the template given in the file. UUIDs should be unique for each entry. The timestamp field should be filled with an integer corresponding to a UNIX timestamp (UTC) converted from "yyyy-mm-dd hh:mm:ss.ms CEST" format.
 - For each timestamp, if data from multiple sources is available for the same field in the data structure, please compute the average.
 - You can use „driver.py“ to verify whether your program would pass the grading: ``python3 driver.py``.
 - This file will give you feedback on your solution.
- Create a **zip** file of your submission:
`zip -r lab03-<Your StudIP Username>.zip lab03 Makefile`

- To make it easier, you can just run ``make zip`` in the top-level handout folder to automatically create a zip archive with the correct directory structure.
- You can make a copy of the produced zip file named "lab03-submission.zip" and run "make" to check if your file can pass the automatic grading.
- To submit your solution, upload it with a filename corresponding to the format "lab03-<Your StudIP Username>.zip" in the timed submission folder in StudIP.

Task Description – IoT Sensing – Energy sellers

- Imagine a smart electric vehicle (EV) charging station that is connected to a solar panel. The goal of the smart charging station is to serve as many EVs with as much solar energy as possible. To do so, the smart charging station is interested in weather information (e.g., temperature, precipitation, cloudiness) from different sources. Bad (rainy, cloudy) indicates a lower output from the solar panel and thus less energy to charge EVs. However, lots of sun and a blue sky increase the solar panel output which allows the smart charging station to sell more energy.
- Your task is to create a program that queries (senses) the different data sources for weather information.
- The different sources provide different formats of weather data. Therefore, you need to transform it into a single data format for later processing.
- The expected output of your program is a combined set of weather data (that adheres to the defined format and structure) that contains all of the relevant weather data information from all available sensors.
- Voltage readings are from a TMP36 sensor and needs to be converted to temperature in °C before averaging.

Questions? : etce-etce@tu-clausthal.de