

1. The tasks described in this worksheet are part of the formative assessment. They serve the purpose to prepare you for the examination. We will discuss the solutions during the next **interactive session** after they are handed out – while they fit to the lecture of the week they are handed out, they might be discussed in two weeks time due to the bi-weekly exercise schedule.
2. Make sure to plan your time for the whole sheet carefully. The complete exercise should represent approximately three hours of independent study. The time limit indicates how much time you should spend on each task, and not how much time you may actually need; it is important that you engage with the material and not that you complete all tasks perfectly. Feel free to collaborate and team up.
3. The exercises are designed to challenge you and train you further as guided self-study. The time limit might be too ambitious for you; you may team up with colleagues. It is not an issue as long as you manage to at least partially resolve each task within the time budget. If you (and your team) are struggling, reach out for help in Teams! You may also share your thoughts via the Studip Forum.
4. We recommend that you create a (private) Git repository (see <https://gitlab.gwdg.de>) where you store your findings and outcomes while processing the exercises. This portfolio of work can be useful in the future.

## Contents

<b>Task 1: Understanding the NetCDF Data Modell (90 min)</b>	<b>1</b>
<b>Task 2: Creating you Own Benchmark using the Scientific Method (150 min)</b>	<b>2</b>

## Task 1: Understanding the NetCDF Data Modell (90 min)

As part of this exercise, you will inspect NetCDF data and visualize it. Then, you will create a data model and a corresponding NetCDF file from it.

Follow these steps for the inspection:

- Download the NetCDF file:  
[https://www.unidata.ucar.edu/software/netcdf/examples/ECMWF\\_ERA-40\\_subset.nc](https://www.unidata.ucar.edu/software/netcdf/examples/ECMWF_ERA-40_subset.nc)
- Inspect the **header** first and then the data using `ncdump`
- Visualize the variables using `ncview`

Create a CDL data model for the following use case:

- `wikipediaWC` should be an array of word counts for 10 specific words
- `population` should be a 3D array of the number of people on the world surface over time
- `populationCity` should be a 2D array of the number of people living in cities over time

Create a corresponding CDL file with some sample data and autogenerate the NetCDF file.

---

## Portfolio (directory: 11/nc)

11/nc/sample.cdl    The CDL file for the header of our sample NetCDF file  
11/nc/model.cdl    The data model in CDL suitable for `ncgen`

## Hints

- You may need to install additional packages on your system to add `ncdump` and `ncview` such as `netcdf-bin` and `ncview`.
- Use `ncgen3` to create the model
- With `ncdump`, a corresponding ASCII file in CDL can be created from a NetCDF file
- With `ncgen3 <INPUT> -o <OUT>`, the NetCDF file can be created again
- You may find this tutorial for `ncview` useful  
[http://cirrus.ucsd.edu/~pierce/software/ncview/quick\\_intro.html](http://cirrus.ucsd.edu/~pierce/software/ncview/quick_intro.html)
- More example files can be found on  
<https://www.unidata.ucar.edu/software/netcdf/examples/files.html>
- An additional tutorial can be found here  
<https://youtu.be/L-BOT--meTw>

## Task 2: Creating you Own Benchmark using the Scientific Method (150 min)

In the lecture, you saw the importance of benchmarks to assess system performance. As part of this task, you'll create a simple benchmark (in the programming language of your choice) for measuring I/O performance and assess its outcome.

The recommendation for the approach is as follows:

1. First, think about the pattern you like to measure, e.g., sequential I/O of 100 KiB accesses.
2. Define the measurement protocol and metrics: How to time the operations, granularity of the measurement. How do you increase confidence that the returned value reflects the true value in the system?
3. Think about the expected performance on your system and document it.
4. Implement the benchmark.
5. Run the benchmark in such a way that it meets your experiment's measurement protocol.
6. Compare the results with your observations and document them briefly.
7. Conclude the experiment in your measurement protocol. Compare your steps and methodology with the scientific method.

## Portfolio (directory: 11/benchmark)

11/benchmark/protocol.txt    The experimental protocol and theoretic considerations  
11/benchmark/io-bench.[py|c]    The code for your benchmark