

# Faculty of Biosciences, Fisheries and Economics

|  |  |
| --- | --- |
| **Home exam in:** | BIO-3027/8027 |
| **Exam starts:** |  |
| **Exam ends:** |  |
| **Course coordinator** | Ines Heiland |
| **Support:** | On the submission deadline day, call 776 20 880 for support. If you experience waiting time while calling the support hotline, email your answer file to[**eksamen@hjelp.uit.no**](mailto:%20eksamen@hjelp.uit.no)while you are waiting. **Do not leave the queue.** |
| **Number of pages (including this front page):** | 2 |
| **Weighting of questions or other information:** |  |
| **Important information about citation and plagiarism:** | This is an individual examination and must be completed without cooperation with other students.  You are allowed to use any kind of sources (lecture notes, literature, internet-sources, AI-tools or other sources) in answering the topics of the exam.  Be aware that reproducing sources, output from AI-tools, web pages, other students or literature without reference is not allowed, and will be considered as cheating. Use of AI-tools must be described in a dedicated paragraph in your answer paper. Reusing your own previously submitted examination work without adequate source referencing is also regarded as cheating. All exam answers delivered in Wiseflow are automatically checked for plagiarism. |
| **Important information regarding submission:** | **NB!** Remember to make sure you submitted the correct document.  You need to submit at least one document as pdf. If you document and describe your pipeline in Python Notebook. You still need to submit a word or pdf document stating that the pipeline together with the required documentation is provided as Python Notebook.  Call the support line if you encounter any problems when submitting the exam. Do not leave the queue even if you experience long waiting. |

**Bio-3027 / Bio-8027 Assignment Guidelines**

***What's this about?***

Learning programming is best done by practicing on real-world examples relevant to you. The goal of this assignment is therefore to get you to combine and demonstrate what you have learned so far about Python and scientific programming overall.

Objective: You shall program a short pipeline to perform an analysis relevant to your project, whether it be a MSc. or PhD thesis, or another scientific project. It has to be different from the pipelines and scripts we used during classes! This assignment serves as your exam for this course, and the grade will be Pass/Fail.

You have a lot of freedom to design and construct this pipeline - with respect to the code format (e.g. script or notebook, or combination of these) or the packages you use (e.g. pandas, numpy, new packages not covered in the course that you found, etc.), and of course freedom to choose what your pipeline does. So, you'll have to do quite some thinking before you even start coding.

There are a few requirements for this assignment:

**Hard requirements:**

Before you submit your assignment, make sure:

1. Your code runs! It should run from start to finish smoothly, with no errors that cause early exits. If there are non-standard modules required to run the module you need to specify this in your submission.

2. Your code should be **well documented**. Make sure you have a header for your script, or headings for sections of your notebook. Give your functions meaningful docstrings, and use comments to clarify your code. Alternative provide a usage and installation guideline in a separate word or pdf-file!

3. The pipeline should be described. Describe the objective and design of your pipeline in the script/notebook directly or as a separate document (needs to be PDF if so)! Describe and discuss the results!

**Guidelines:**

Your pipeline should include the following:

- Data handling: file reading, file writing, data parsing, working with data structures

- Variety of data structures: show that you have mastery of data types and structures (when to use them, how to use them)

- Modular design: break up your code into logical chunks, use functions to perform actions such that you don't have to repeat yourself, minimize hard coding

- Use of data analysis packages: show you know how to import, read up on, and use packages

- Summary of results: either through visual plots or summarizing charts, depending on the goal of the analysis. Results shall be explained and described!