

## Final Project Grading Schema

- Project registration until 11.06., hand in until 09.07., grading until 31.08.
- If you are not satisfied with the grade, you can always choose to take a “pass” instead

Projects should be handed in as GitHub repositories with README.md files and Python code in .py files. It's also possible to use Jupyter Notebooks, but only to demonstrate e.g. usage or interpretation of the outputs, not for the main body of the code. You can assume that the user has a local system with working Python 3.7, Miniconda, and Git installations and basic knowledge of how to use them. The following grading schema is going to be the guideline for the grading of the final project. The finer distinctions within design, implementation, and documentation may not be applicable to every project. If that's the case, we will communicate this when the project is registered.

<b>Design</b>	Suitable inputs, outputs, methods were chosen. <ul style="list-style-type: none"> <li>• Outputs relevant for project goal</li> <li>• Inputs relevant for project goal</li> <li>• Methods adequate for project goal</li> </ul>	6 points
	Project is - to some degree - customizable. <ul style="list-style-type: none"> <li>• User can select different settings or inputs</li> </ul>	4 points
<b>Implementation</b>	The project is structured sensibly overall into folders and files. <ul style="list-style-type: none"> <li>• Separate folders for inputs and outputs</li> <li>• README.md at top level</li> <li>• Separate file for requirements and settings where appropriate</li> </ul>	3 points
	The code is structured sensibly. <ul style="list-style-type: none"> <li>• Split up different functionality into different files and functions</li> <li>• As little code redundancy as possible</li> <li>• Bonus points for elegant solutions</li> </ul>	5 points
	The code is executable and yields the expected outputs.	6 points

	<p>The code is the most efficient solution available.</p> <ul style="list-style-type: none"> <li>• Known library functions for computationally intensive operations are used</li> <li>• There are no obvious inefficiencies (endless loops, functionality duplication)</li> </ul>	6 points
<b>Documentation</b>	<p>There is a comprehensive overview of the project's goal, motivation and structure</p>	5 points
	<p>There are instructions for setting up the project on a local machine.</p> <ul style="list-style-type: none"> <li>• Requirements are clearly described</li> <li>• Download links for external resources</li> </ul>	3 points
	<p>There are instructions for the intended usage of the project.</p> <ul style="list-style-type: none"> <li>• Commands to run the project</li> <li>• Information on selecting inputs</li> <li>• Information on selecting settings</li> <li>• Examples for interpreting possible outputs</li> </ul>	6 points
	<p>The code is well-documented.</p> <ul style="list-style-type: none"> <li>• Docstrings for all functions</li> <li>• Comments for complex pieces of code</li> <li>• Sensible variable names</li> </ul>	6 points

There are 50 points in total: 10 for design, 20 for implementation, and 20 for documentation. The total sum of points will be used to determine the numerical grade that each person in the group will receive for the whole course. Of course, you can also do a final project on your own without a group.

$\geq 47.5$	1.0	$\geq 35.0$	2.7
$\geq 45.0$	1.3	$\geq 32.5$	3.0
$\geq 42.5$	1.7	$\geq 30.0$	3.3
$\geq 40.0$	2.0	$\geq 27.5$	3.7
$\geq 37.5$	2.3	$\geq 25.0$	4.0