Updated: August 26, 2025

GRA4152 – Object Oriented Programming with Python Autum 2025

Course responsible: ***Rogelio A Mancisidor***

- Lecturer: Assistant Professor ***Rogelio A Mancisidor,*** e-mail: rogelio.a.mancisidor@bi.no

  Lectures: Tuesdays 10:00 - 11:45 (See Web for details etc.)

  Office hour: B3y-077 Tuesdays 14-15

- Compendium Book: Python for Everyone (PfE), Horstmann, Cay & Necaise, Rance. Wiley 3/e.

  Additional book: Pro Git (PG), Scott, Chacon & Ben Straub. Apress 2/e. https://git-scm.com/book/en/v2

- Quiz website: https://trysakai.longsight.com/

- Git online training (GT): link

- Course GitHub page: https://github.com/BI-DS/GRA-4152

- **Note:** It is highly recommended to work regularly on the practical exercises and quizzes as they are highly relevant for passing the final project.

You are advised to read the curriculum in advance of the lectures. There may be changes to the schedule.

Date	*Topic*	*Reading*	*Asynchronous*	*TA session*
26/08	***Background and Python recap*** *- About the course* *- Python recap*	- Slides - PfE: 2.1, 2.4, 3.1, 3.4, 3.7, 4.1, 4.6, 5.2.	- Sign up in Sakai - Revise https://github.com/BI-DS/EBA3400 - P2.3 - P2.10 - P3.12 - P4.17	
0/09	***Control versioning*** *- Python recap* *- Git and GitHub* *- Git commands*	- Slides	- git_exercises.pdf -CH 2 PG - GT: staging, investigating, and branching	
09/09	***Parsing arguments and Parallelising Python***	- Slides	- Go through exercise approximating pi. Both slides and pi_approx.py code (Lecture 9 in Github)	**TA session 11/Sep 14:00-15:45 C2-095** - git_exercises.pdf - GT: Basic branching, 3 way merge

	- Use of argparse for easy argument optimization *- Learn to use Numba to speed up Python code*			- Exercise approximating pi (If I didnt have time during class)
16/09	***Debugging techniques*** ***-*** *Exceptions* *- Assertions* *- Debugging tools: GNB online debugger*	- Slides	- Go through exercise List. Both slides and listo.py code (Lecture 10 in Github) - GitExercises.pdf	
23/09	***Big Data***	- Slides	- Argparse.pdf	
30/09	***Object and Classes*** *- UML diagrams* *- Class* *- counter.py* *- Constructors*	- Slides - PfE: 9.1, 9.2, 9.3, 9-4, 9.5, 9.6	- Worked Example 9.1 - P9.1 - P9.2 - P9.3	**TA session 02/Oct 14:00-15:45 B2-095** - P9.5 - P9.19 - P9.20
07/10	***Object and Classes*** *- Constructors* *- Methods* *- Decorators* *- cashregister.py*	- Slides - PfE: 9.5, 9.6	- Quizz 1 (Sakai) - P9.4 - P9.24 - P9.26	
1/10	***Inheritance*** *- Hierarchies* *- Superclass, Subclass* *- car.py*	- Slides - PfE: 10.1, 10.2, 10.3	- Exercises cap 9 - Worked Example 10.1 - P10.1 - P10.3	
21/10	***Inheritance*** ***-*** *car.py* *- Overriding methods* *- Polymorphism*	- Slides - PfE: 10.4, 10.5	- Quizz 2 (Sakai) - P10.2 - P10.5	
28/10	***Inheritance*** *- Polymorphism* *- accounts.py* *- employees.py*	- Slides - PfE: 10.4, 10.5,	- P10.6 - P10.8	**TA session 30/Oct 14:00-15:45 B2-095** - P10.4 - P10.7 - P10.9
06/11	***Inheritance*** *- Polymorphism* *- accounts.py* *- employees.py*	- Slides	- P10.21 - P10.23	
11211	***Recap lecture and Project:*** *Create superclass & subclass using*	- Project description and instructions	- Project	

	different OOP concepts			