

Course Manual

PRA2031: Python Programming Language

Academic Year: 2025-2026
Course Period: 4

Table of Contents

Introduction	3
General Introduction	3
Course team	3
Course Description	4
Intended learning outcomes	4
Course Design	5
Structure	5
Schedule	5
Resources	5
Remote access	5
Additional resources	5
Inclusive classroom	6
Course Assessment	7
Attendance requirements	7
Assessment	7
Grading	8
Weighting	8
Resit	8
Exam inspection	8
Fraud and Plagiarism	9

Introduction

General Introduction

Python is one of the most popular programming languages in the world. It is a high-level programming language, which was designed to emphasize code readability. Its application ranges from web applications over scientific computing to artificial intelligence projects.

Course team

Course coordinators



Dr. Ruth Großholz
Faculty of Science and Engineering
Maastricht Centre of Systems Biology and Bioinformatics (MaCSBio)
Brightlands Future Farming Institute (BFFI)
ruth.grosseholz@maastrichtuniversity.nl

Tutors

Course Description

Python is one of the most popular programming languages in the world. It is a high-level programming language, which was designed to emphasize code readability. Its application ranges from web applications over scientific computing to artificial intelligence projects.

In this course you will expand your knowledge and understanding of python programming by working in groups on an example project from one of the scientific fields (biology, physics and chemistry). Together, we will learn the basics of object-oriented programming, take a closer look at methods within classes and code debugging methods. You will learn to identify and use relevant python packages for your project and to document your code so that is understandable to other people. Furthermore, this course will introduce git and GitHub repositories for version control, collaborative programming work, and code sharing.

Intended learning outcomes

At the end of this course, we expect you to be able to:

- Learn to use the principles object-oriented programming and classes in python
- Explore using python to program, simulate and visualize a problem
- Learn basic, essential techniques to debug your code, using virtual environments and package management
- Using GitHub repositories for version control, collaborative programming and code sharing

Course Design

Structure

Each session will consist of a short lecture to introduce the week's topic, followed by short programming tasks to practice the introduced concepts. The rest of the session will be spent on the group project. Students should bring their own laptop.

These sessions will take place between 9:00 and 17:00, with a lunch break from 12:00 to 13:00. In week 5, you will have an exam covering the topics of week 1 – 4.

Schedule

Week 1	Interactive session	Python recap, GitHub and introduction to OOP	
	Group project	Working on the group project	
Week 2	Interactive session	Git Collaboration and Core OOP Concepts	
	Group project	Working on the group project	
Week 3	Interactive session	Inheritance and Polymorphism	
	Group project	Working on the group project	
Week 4	Interactive session	Special Methods and Operator Overloading	
	Group project	Working on the group project	
	Assessment	Submit your GitHub repository for peer review	Formative assessment
Week 5	Assessment	Midterm exam	Topics of weeks 1-4
	Group project	Finalising the group project	
	Assessment	Submit review another group's GitHub repository	Formative assessment
Week 6	Assessment	Presenting the group project	
	Assessment	Deadline for the group project	Submit GitHub repository, and peer review

Resources

No specific textbook is required. However, relevant literature will be provided on Canvas for each topic.

Remote access

Maastricht University students have off-campus (this means outside Maastricht) access to protected services and resources, like databases, e-journals and e-books of the UM Library. On Brightlands locations, you can use the public "Brightlands-Guest" Wi-Fi to get internet access in case "Eduroam" is unavailable. Different solutions are provided for remote access, such as the Lean Library Browser Extension and Student Desktop Anywhere (SDA). All information is provided on the website: <https://umlib.nl/remote-access>.

Additional resources

Other resources important to your education, such as the academic calendar, course descriptions, and the education and examination regulations (EER), can be found on the [student intranet](#).

Inclusive classroom

Course coordinators uphold the Maastricht University's commitment to ensuring equality and valuing diversity. UM recognizes that our individual differences can deepen our understanding of one another and the world around us, rather than divide us. In this course, people of all ethnicities, genders and gender identities, religions, ages, sexual orientations, disabilities, socioeconomic backgrounds, regions, and nationalities are strongly encouraged to share their rich array of perspectives and experiences. If you feel your differences may in some way isolate you from UM, please speak with the course coordinator early in the period about your concerns and what we can do together to help you become an active and engaged member of our class and community. For some of you, the following resources may be helpful:

[General information on diversity and inclusivity at UM](#)

[Disability support](#)

[Community support](#)

[Psychological support](#)

[Confidential counsellor](#)

Using GenAI

You may use generative AI tools (such as ChatGPT, Copilot, or similar systems) for this course and its assessments, provided that:

- You use them responsibly and transparently, in accordance with academic integrity principles.
- You disclose any use of AI in your submission, indicating which tools you used and how they supported your work.
- The final product must still reflect your own understanding and achievement.

Remember: AI tools can assist your learning but do not replace your own critical thinking or accountability for submitted work. It is your responsibility to understand the code that you work and that you submit for the group project.

Course Assessment

Attendance requirements

For skills trainings, you must be present, 100% of all sessions. If you have a valid reason (illness etc.), it is possible to compensate 1 session with an additional assignment. The absence with reason has to be communicated as soon as possible, but always before the missed session has ended. The coordinator will decide on an individual basis whether the reason is valid.

Late policy:

The practical course takes place between 9.00 to 17.00. If you arrive late, you will be marked as late in canvas (attendance of 80%). A passing grade for each day (55%) for attendance is required.

Assessment

Formative assessment

In this course, you will have two formative assessments, which will give you feedback on your study progress.

- You will have weekly quizzes which will help you prepare for the midterm exam and give you the chance to check your understanding of the covered concepts. You need to complete 4 of the 5 quizzes to pass the course. The quizzes will have a similar format to the questions that will be asked in the midterm exam.
- You will receive peer feedback on your GitHub repository in week 5. For this, you will need to submit a link to your repository in week 4. *Failure to provide a link to your repository or to provide constructive feedback on the GitHub repository of another group will result in a 1point reduction from your grade for the group assignment.*

Summative assessment

Your grade in this course consists of the following graded components:

- Midterm exam (50%) in week 5, which covers the topics of weeks 1-4.
- Individual tutor assessment (10%),
- Peer review of your contribution to the group project (10%),
- Group grade on the project (30%) (including GitHub repository, code quality, code documentation, and presentation). The grade of the presentation will include an individual component for the presentation style and quality of answers to questions.

Alignment with intended learning outcomes

The table below illustrates the relative weight of the ILOs in the summative assessments.

Intended learning outcomes	Assessment 1	Assessment 2	Assessment 3	Assessment 4
Learn to use the principles of OOP	80 %	25 %	-	20 %
Explore python to program, simulate and visualize a problem	-	25 %	50 %	50 %
Learn debugging, virtual environments and package management	-	25 %	30 %	10 %
Use GitHub repositories for version control, collaborative programming and code sharing	20 %	25 %	20 %	20 %
	100 %	100 %	100 %	100 %

Grading

To pass the course, you need a weighted grade of 5.5 or higher. Whether individual assessments need a passing grade as well, you can find in the table below.

Graded course component	Weight	Pass requirement	If failed
Attendance	–	See above	Redo module
Weekly quizzes	–	Complete 4 out of 5	Redo module
Summative assessment 1	50 %	-	-
Summative assessment 2	10 %	-	-
Summative assessment 3	10 %	-	-
Summative assessment 4	30 %	-	-
Your final grade will be calculated as: (assessment 1 * 0.5) + (assessment 2 * 0.1) + (assessment 3 * 0.1) + (assessment 4 * 0.3)			

Weighting

This composition allows you to be evaluated at different time points and for different aspects. The formative assessment points help you to establish your baseline for this course, and to monitor your progress. These are not counted towards the final grade. The two summative assessments are a way to test if you have obtained the necessary basis towards the final exam. In the final summative exam, you can test but also apply your basic knowledge to analyse a biological system. The weighting reflects partially the importance of the topics and partially the time spent on them.

Resit

In case you fail the course, or a required course component, please check the table above what type of assignment may be offered, subject to conditions. A resit will be offered if the student made an honest attempt at passing the course. Relevant information will be communicated through the [Student Portal](#) in due time.

Exam inspection

Publication of grade

The examiner decides on the grade for a written or other examination and provides the Office of Student Affairs with the written or electronic assessment, within a maximum of 15 working days after the date of the examination. The Office of Student Affairs (OSA) processes the grade, after which it will be published on your [Student Portal](#).

Inspection

Within 10 working days after the publication of the final grade:

- The course coordinator communicates details regarding the inspection of assessments.
- You may request inspection of your assessed work.
- An (individual) appointment is made for inspection.

In case of written exams, note that you are not allowed to take these with you or share the exam with other students or third parties.

Appeal

For information regarding the appeal procedure, please consult the programme's Education and Examination Regulations (EER).

Fraud and Plagiarism

In order to protect the reputation of the degree that you will receive, instances of fraud such as cheating or plagiarism are treated extremely seriously. Fraud is understood as a student's act or failure to act that makes it (partially or fully) impossible to accurately assess his/her knowledge, insight and skills. Plagiarism is understood as the presentation of one's own or other people's ideas or words without adequate reference to the source. This includes plagiarizing from published or unpublished work, such as utilizing written pieces from past students or other students in your group. If it is published, the work should be properly cited and paraphrased to avoid issues. Work (partially) generated by Artificial Intelligence also makes it (partially or fully) impossible to accurately assess your knowledge, and is therefore also considered fraud, unless the assignment explicitly asks for the use of AI and is cited as such.

Any assignment is an individual piece of work, which means that plagiarism is strictly forbidden. Equally, the use of mobile phones, communication devices or any other information carrier (whether the phone or other device is turned on or off, used or not used, etc. is irrelevant) during a sat examination is also forbidden. All irregularities are sent to the Board of Examiners. If the Board of Examiners concludes that anything has occurred in an examination that makes it partially or fully impossible to correctly assess his/her knowledge, insight and skills, they may impose a sanction in accordance with UM's policy on fraud, including plagiarism.