Course unit (module) title	Code
Programming for Biological Data Analysis	

Lecturer(s)	Department(s) where the course unit (module) is delivered
Coordinator: dr. Tomas Plankis **Other(s):**	Faculty of Mathematics and Informatics Naugarduko str. 24, LT-03225 Vilnius

Study cycle	Type of the course unit (module)
Second cycle	Elective

Mode of delivery	Period when the course unit (module) is delivered	Language(s) of instruction
Face-to-face, self-study Lectures, seminars and practice	First semester	English

Requirements for students	
Prerequisites: Basics of computer literacy	**Additional requirements (if any):** GNU/Linux type operating systems

Course (module) volume in credits	Total student's workload	Contact hours	Self-study hours
5	132	64	68

Purpose of the course unit (module): programme competences to be developed		
The course aims to develop programming skills which are necessary for solving problems in systems biology. This course is based on Python programming language and specialized Python tools for data analysis and visualization. After completing the course, students should be able to (1) apply the skills they have learned to tackle problems in their own research and (2) continue programming learning in a self-directed way.		
Learning outcomes of the course unit (module)	**Teaching and learning methods**	**Assessment methods**
1.2 Be able to analyse, manage and model data from the specialized libraries in the field of system biology	Lectures, practical assignments	Practical assignments; Final exam
3.3 Be able to apply advanced data processing and programming techniques	Lectures, practical assignments	Practical assignments; Final exam
3.4 Be able perform practical calculations using modern high-performance open computing platforms	Lectures, practical assignments	Practical assignments; Final exam
4.1 Perform assignments within the deadlines and goals of a project	Lectures, practical assignments	Practical assignments; Final exam
5.1 Be able to work autonomously and as a part of a multidisciplinary team; act honestly and according to ethical obligations	Lectures, practical assignments	Practical assignments; Final exam
5.2 Be able to critically analyse their own professional practices with a view to improving them	Lectures, practical assignments	Practical assignments; Final exam

Content: breakdown of the topics	Contact hours							Self-study work: time and assignments	
	Lectures	Tutorials	Seminars	Exercises	Laboratory work	Internship/work placement	**Contact hours**	**Self-study hours**	**Assignments**
1. Introduction: basic concepts and programming principles	4			4			**8**	**8**	Practical assignment #1
2. Data: files, input/output, basic operations with various data types	4			4			**8**	**8**	Practical assignment #2
3. Data flow control: loops and conditionals, exception handling	4			4			**8**	**8**	Practical assignment #3
4. Organizing and structuring code: functions and procedures, libraries, version control	4			4			**8**	**8**	Practical assignment #4
5. Interaction with the file system. Data visualization (matplotlib)	4			4			**8**	**8**	Practical assignment #5
6. Data structures: collections and sets, vocabularies, lists, hierarchical data structures and recursion	4			4			**8**	**8**	Practical assignment #6
7. Object-oriented programming	4			4			**8**	**8**	Practical assignment #7
8. Specialized libraries for biological data analysis (BioPython, pandas etc.)	4			4			**8**	**8**	Practical assignment #8
Final exam (preparation)								**2**	
Total	**32**			**32**			**64**	**68**	

<table>
<tr><th>Assessment strategy</th><th>Weight,%</th><th>Deadline</th><th>Assessment criteria</th></tr>
<tr><td>Practical assignments</td><td>40</td><td>Till appointed deadlines</td><td>There will we eight practical assignments (one for each topic of the course). Each assignment consists of several exercises which must be solved and uploaded to Virtual learning environment before appointed deadlines. Practical assignments are defended during practical sessions immediately after submission deadlines. Participation in these practical sessions is obligatory. Each practical assignment is graded in a 10 point scale.</td></tr>
<tr><td>Final exam</td><td>60</td><td>During the official scheduled time</td><td>The final exam consists of practical questions and problems, which cover all topics of the course. The final exam is graded in a 10 point scale.
The final grade is based upon the final score. The final score is a weighted average of the practical assignments grades and the final exam grade, with the following weights:
Practical assignments – 5 % each
Final exam – 60 %
The final score is converted to the final grade according to the following table:
<table>
<tr><td>Final score</td><td>Final grade</td><td>ECTS equivalent</td></tr>
<tr><td>9.5–10.0 (95–100%)</td><td>10 (Excellent)</td><td>A</td></tr>
<tr><td>8.5–9.4 (85–94%)</td><td>9 (Very goodl)</td><td>A</td></tr>
<tr><td>7.5–8.4 (75–84%)</td><td>8 (Good)</td><td>B</td></tr>
<tr><td>6.5–7.4 (65–74%)</td><td>7 (Highly</td><td>C</td></tr>
</table>
</td></tr>
</table>

	satisfactory)	
5.5–6.4 (55–64%)	6 (Satisfactory)	D
5.0–5.4 (50–54%)	5 (Sufficient)	E
4.0–4.9 (40–49%)	4 (Insufficient)	F
3.0–3.9 (30–39%)	3 (Insufficient)	F
2.0–2.9 (20–29%)	2 (Insufficient)	F
0–1.9 (0–19%)	1 (Insufficient)	F

Author	**Year of publication**	**Title**	**Issue of a periodical or volume of a publication**	**Publishing place and house or web link**
Compulsary reading				
M. Jones	2013	Python for Biologists: A complete programming course for beginners		http://pythonforbiologists.com/
M. Jones	2014	Advanced Python for Biologists		http://pythonforbiologists.com/
T. J. Stevens, W. Boucher	2014	Python Programming for Biology: Bioinformatics and Beyond		Cambridge University Press
Optional reading				
R. Wunschiers	2013	Computational Biology: A Practical Introduction to BioData Processing and Analysis with Linux, MySQL, and R		Springer-Verlag Berlin Heidelberg, DOI: 10.1007/978-3-642-34749-8
M. Allerhand	2011	A Tiny Handbook of R		Springer Berlin Heidelberg, DOI: 10.1007/978-3-642-17980

****