COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Methods of computer program construction	5BIOMC

Lecturer(s)	Department where the course unit is delivered
Coordinator: Saulius Gražulis **Other lecturers:**	Department of Mathematical Computer Science Faculty of Mathematics and Informatics Vilnius University

Cycle	Type of the course unit
1st (BA)	Compulsory

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	5 semester	Lithuanian (English)

Prerequisites
Prerequisites: Perl programming language, introduction to informatics, data structures **Desirable knowledge:** linear algebra, operating systems

Number of credits allocated	Student‘s workload	Contact hours	Individual work
5	130	50	80

Purpose of the course unit: programme competences to be developed

Purpose of the course unit is to provide students with basic skills of efficient software construction for the purpose of bioinformatics data processing: students should learn how to use version control systems for program release cycle management, automated testing for improvement of software quality. Students will get acquainted with the benefits of consistent programming style, software verification, and will learn to work efficiently in Unix-like operating system environments (for instance, in a GNU/Linux system).

This course is intended as a prerequisite for the course of structural bioinformatics, where the acquired skills and knowledge will be used to write programs for bioinformatics and to process bioinformatics data.

Generic competences:

- Ability to search, analyse, represent and organise the information *(GK1)*.
- Ability to apply the knowledge in practice *(GK2)*.
- Ability to organise and plan the work, to work in a team as well as individually, ability to interact with the professionals of different areas. *(GK3)*.

Specific competences:

- Algorithms and data structures *(SK5)*.
- Programming models and internet technology *(SK6)*.
- Software engineering (*SK8*).
- Extracting, representation and analysis of bioinformatics data (*SK11*).

Learning outcomes of the course unit: students will be able to	Teaching and learning methods	Assessment methods
Use efficiently Unix-like computer environments for software production and data processing.	Lectures, seminars, problem-based learning, individual assignments, practical classes, self-study.	Midterm exams; final exam; topic-related practical assignment evaluation, practical work report.
Understand the basic concepts of version control; use the Subversion version control system efficiently for software development.		

Understand the necessity, benefits and limitations of software testing; use the automated testing environment based on GNU Make utility for their own programs.		
Perform the basic program verification steps.		
Write a readable, easy to maintain program code.		
To understand the basic concepts of the modern programming techniques such ax agile programming extreme programming (XP), test driven development ad to apply the basic elements of these techniques in practice.		

Course content: breakdown of the topics	**Contact hours**							**Individual work: time and assignments**
	Lectures	Tutorials	Seminars	Practice	Laboratory work	**Contact hours**	**Individual work**	**Assignments**
1. Basic principle of the Unix architecture, file system, commands	4			2		**6**	**7**	
2. Version control and Subversion	4			2		**6**	**14**	
3. The Unix programming environment; the GNU Linux systems and their capabilities	4			4		**8**	**14**	
4. Automated program building and testing using the GNU Make tool	8			4		**12**	**14**	
5. The use of Unix-like environments and of the GNU Make system for data processing	4			4		**8**	**14**	
6. Program verification and correctness proofs; their application for everyday programming	4					**4**	**7**	
7. The history and advanced features of Unix and Linux OSes	4					**4**	**7**	
8. Preparation for a exam, exam	2					**2**	**3**	
Total	**34**			**16**		**50**	**80**	

Assessment strategy	**Weight %**	**Deadline**	**Assessment criteria**
Classwork assessment	10	Beginning of each practical seminar	A quiz (virtual learning environment) of 4 questions from topics the topics covered in the previous lectures. The scores from all answers in all quizzes are summed up; maximal sum is 100 points.
Midterm exam	15	Middle of the course	Test (virtual learning environment) including questions from the topics learned so far; maximum score from this test is 150 points.

Assessment of individual assignments	50	After each assignment, according to the schedule provided in the Virtual Learning Environment	Students must upload their assignment to the Virtual Learning Environment. The evaluation criteria of each practical assignment will include: achievement of the goals set for the practical work, coding style and readability of the code, general knowledge on the subject. Evaluation will be conducted using subtrative method: an assignment that was carried out ideally will be worth 100% of the score; each deficiency will attract negative scores depending on its importance (the importance and the nature of the deficiency will be explained). Additional (bonus) assignements may be issued to help students to correct the previous deficiencies.
Presentation of the practical work results	10	Last week of the course	Students must upload a report (type-setted according to the presentation standards of the Vilnius University) to the Virtual Learning Environment and prepare a 5 – 10 min. talk on his/her work. Evaluation criteria will include: achievement of the goals set for the practical work, understanding of the topic (as judged from the answers to several topic related questions), written presentation of the work, oral presentation. The evaluation will be carried out either using the Moodle Rubric method or the subtractve method, as for the assignments.
Exam	15	Exam session	Approx. 30-question quiz covering several recent lectures (Bloom's 1 to 9 level questions) using an electronic teaching environment (Moodle, Open edX or similar). To be eligible for the exam, students must fulfil all following criteria: 1. carry out at least one practical work and get a positive grade for the practicals; 2. have enough accumulated points to be able to pass the exam in principle if they score maximum points at the exam quiz; Participation in the final exam quiz is obligatory to pass the course, regardless of the accumulated points. Students who do not show up in the final exam will be indicated as such in the exam grading report. To pass the exam, on must score at least 50% of possible points.
Total	100		The final mark is obtained by summing up points earned in all quizzes and tests (summing up to 1000 points), dividing by 100 and rounding to the next largest integer (thus a sum, for instance, of 901 point would give the final mark 10).

Author	**Publishing year**	**Title**	**Number or volume**	**Publisher or URL**
Required reading				
Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato	2011	Version control with Subversion		O'Reilly Media, Inc., http://shop.oreilly.com/product/9780596004484.do, ISBN 978-0596510336, http://svnbook.red-bean.com/.
Richard M. Stallman, Roland McGrath, Paul D. Smith	2010	GNU Make		Free Software Foundation, http://www.gnu.org/software/make/manual/
Kernighan, Brian W.	1984	The UNIX programming environment		Prentice-Hall, Inc.; ISBN 0-13-937681-X
Бриан Керниган	1992	UNIX-универсальная среда программирования		Москва: Финансы и статистика
Recommended reading				
Wikipedia	2013	Test-driven development		http://en.wikipedia.org/wiki/Test-driven_development
Kent Beck	2003	Test-Driven Development		Addison-Wesley, Boston, ISBN-

		By Example		13: 978-0321146533
Kent Beck, Erich Gamma	2005	Extreme Programming Explained: Embrace Change, 2nd Edition (The XP Series)		Addison-Wesley, Boston, ISBN-13: 978-0321278654
Bourne, S. R.	1983	The UNIX system		Addison-Wesley, Boston, ISBN 0-201-13791-7
С. Баурн	1986	Операционная система UNIX		Москва: Мир