COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Scientific programming in Ada	

Lecturer(s)	Department where the course unit is delivered
Coordinator: dr. Saulius Gražulis **Other lecturers:** -	Institute of Computer Science Faculty of Mathematics and Informatics/Life Sciences Center, Vilnius University

Cycle	Type of the course unit
1st (BSc)	Elective

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	spring (2,4) semester	Lithuanian, English

Prerequisites
Prerequisites: Basics of programming; at least one compilable language (C, C++, Pascal or Java)

Number of credits allocated	Student's workload	Contact hours	Individual work
5	134	64	70

Purpose of the course unit: programme competences to be developed

Purpose of the course unit:
The purpose of the course unit is to provide students with insights into reliable program construction and programming best practices, using Ada/SPARK system as an example. Best practices of reliable program construction: strict typing, data hiding, stable interface construction and versioning, program testing and version control will be covered. Formal verification methods will be briefly mentioned. Emphasis will be placed on constructing reliable scientific software for bioinformatics, cheminformatics, computational material science and related disciplines.

Generic competences:

- Ability to analyse 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 *(GK3)*.

Specific competences:

- Algorithms and data structures *(SK5)*.
- Programming models and internet technology *(SK6)*.
- Software engineering (*SK8*).
- Formal software verification and mathematical logics *(SK8)*.

Learning outcomes of the course unit: students will be able to	Teaching and learning methods	Assessment methods
Learn to construct readable, reliable, usable, correct Ada programs for scientific computations. Lern how to formally prove correctness of programs; learn to prove correctness of Ada/SPARK programs Design reusable code (programs and libraries) for scientific com-	Teaching methods: • Lectures; • Laboratory works. Learning methods: • Actual knowledge gathering and accumulation; • Knowledge synthesis – generalization, abstraction and aggregation of actual knowledge; • Knowledge analysis – new knowledge matching with aggregated knowledge, their verification and correction is needed;	Examination. Laboratory works presentation. Criteria: • Use of exercises to assess understanding and ability to apply theoretical knowledge • Ability to develop, explain and modify the

putations.	• Application of aggregated and validated knowledge.	own Ada/SPARK program

Course content: breakdown of the topics	Contact hours							Individual work: time and assignments	
	Lectures	Tutorials	Seminars	Practice	Laboratory work	Practical training	Contact hours	Individual work	Assignments
1. Ada design principles – strict static typing, modularity, information hiding, run/compile time checks. Application areas of Ada. Compiling and running the first program	2					2	4	5	
2. Basic Ada types and control structures, Expressions	2					2	4	5	
3. Subroutines – procedures and functions, parameter passing	2					4	6	6	
4. Ada type compatibility and conversions. User-defined types	2					4	6	6	
5. String and array processing. Dynamic memory handling. Standard container libraries	2					4	6	6	
6. Exceptions. Pragma statements	2					4	6	6	
7. Compilation units, control structures, code reuse, separate compilation	2					4	6	6	
8. Ada generics	2					4	6	6	
9. Parallel programming in Ada – tasks, task types, controlled types, mutual exclusion, randevu	2					4	6	6	
10. Package interface design in Ada. Private types.	2					4	6	6	
11. Object-oriented programming in Ada, tagged types, dispatching	2					2	4	6	
12. Introduction to formal correctness proofs – Hoare logic, SPARK, pre- and post-conditions; invariants, loop variants	2					2	4	6	
	24					**40**	**64**	**70**	

Assessment strategy	Weight %	Deadline	Assessment criteria
Classwork assessment	10	Day of the lecture or practical	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 strategy	Weight %	Deadline	Assessment criteria
Exam. Evaluation of theoretical knowledge using open question exercises	15	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.	Exam session
Performance of laboratory works	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 subtractive 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) assignments 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 subtractive method, as for the assignments.

Assessment strategy	Weight %	Deadline	Assessment criteria
External students			Taking the exam as an external student is permitted by the decision of the lecturer coordinating the subject. As a rule, taking exam as an external student is permitted for very good students (with the academic average of at least 8) who are able to master the subject on their own and only need a knowledge assessment by a qualified VU representative. The requirements that apply to an external student are the same as those to a regular course attendee. A student applying for external student status may not have academic debts; only non-academic debts are permitted.

Author	Publ. year	Title	Number or volume	Publisher or URL
Required reading				
Chapin, P.	2015	Ada–a crash course		URL: https://www.inf.ed.ac.uk/teaching/courses/fv/spark/Ada-A_Crash_Course.pdf
Rogers, P.	2025	Ada in practice		Ada Core. URL: https://learn.adacore.com/pdf_books/courses/ada-in-practice.pdf
Amiard, R. & Hoffmann, G. A. (Editor: Kenner, Richard)	2022	Learning Ada		AdaCore. URL: https://learn.adacore.com/pdf_books/learning-ada.pdf
AdaCore	2025	SPARK Reference Manual		URL: https://docs.adacore.com/live/wave/spark2014/html/spark2014_rm/index.html
AdaCore & Capgemini Engineering	2025	SPARK user's guide		AdaCore. URL: https://docs.adacore.com/spark2014-docs/pdf/spark2014_ug.pdf
Recommended reading				
AdaCore & Thales	2020	Implementation guidance for the adoption of SPARK. Release 1.2		AdaCore and Thales. URL: https://www.adacore.com/uploads/books/pdf/Spark-Guidance-1.2-web.pdf