

<b>Subject code</b>	<b>Credits</b>
INF2027	4

**Title**

PROGRAMAVIMO TECHNOLOGIJOS

**Title in English**

PROGRAMMING TECHNOLOGIES

**Subject goal and annotation**

The course aims to introduce students to functional and object-orientated programming, using Python programming language. During the course programming terminology is introduced; importance of programming is discussed; programming principles, syntax, rules and software design stages are explained; the documentation requirements are presented. Students learn how to manipulate different datasets (such as sequences, lists, and dictionaries), conditional sentences, cycles, functions, objects and their methods. Acquired knowledge can be applied in practice when analyzing the tasks, writhing down corresponding algorithms in Python.

**Prerequisites**

Rudimentary computer and information technology skills.

**Relationship between the learning outcomes of the Programme and learning outcomes of the subject**

<b>Learning outcomes of the Programme</b>	<b>Learning outcomes of the subject</b>	<b>Criteria for measuring the achievement of learning outcomes</b>
3. Knowledge of basic and advanced computer science and its application. 7. Formalization and specification of real-world problems, and ability to describe them at an abstract level 10. Analysis, design and development of advanced Internet systems. 12. Analysis, design and development of diverse software systems. 13. Ability to analyse the newest trends in Internet and multimedia systems (and general computer science and digital arts) and apply them in development of novel systems.	To name the advantages of Python against other programming languages	Name advantages of Python against other programming languages
	To write software documentations	Write software documentations for Python programs
	To use digital types in mathematical operations of self-composed algorithms	Develop algorithms using digital types and manipulate it in mathematical operations
	To use String, Tuple, Set, List, Dictionary types in the operations of self-composed algorithms. To choose which type fits solving problem the best	Develop algorithms using String, Tuple, Set, List, Dictionary types and manipulate them in different operations
	To recognize when conditional sentence is needed for the solving task and use it in self-composed algorithms	Recognize the necessity (and type) of conditional sentence for the solving task and use it in the developed algorithm
	To recognize the most appropriate type of loop for the solving task and use it in self-composed algorithms	Recognize the necessity (and type) of loop for the solving task and use it in the developed algorithm
	To split algorithm code into modules, functions, to pass and return values through the parameters. To recognize the necessity to use functions and modules for the solving task	Recognize the necessity and use modules and functions for the solving task
	To read/write/append text files with the help of Python programs	Develop algorithms that use different manipulation (reading/writing/appending) with files
	To name differences between function and object programming. To write simple algorithms using classes and methods.	Develop simple algorithms using object-oriented programming

	To write simple algorithms using basic elements of graphical user interface (buttons, input fields, etc.).	Develop simple algorithms using basic elements of graphical user interface
--	--	--

### Subject content

	Lecture topics and contents	Hours
1.	Introduction: advantages programming with python	1
2.	Requirements for software documentations	2
3.	Data types: digital types. Mathematical operations	6
4.	Data types: string type	6
5.	Data types: tuples and sets	6
6.	Data types: lists and dictionaries	6
7.	Conditional sentences	6
8.	Loops	6
9.	Functions and modules	6
10.	Files	5
11.	Basics of classes, methods and inheritance	5
12.	Basics of graphical user interface	5
	<b>Total</b>	<b>60</b>

### Practical work contents

Two practical works. Programs in python have to be presented and documentations have to be prepared.  
 1. Task manipulating different data types.  
 2. Task using different advantages of function programming paradigm (conditional sentences, loops, functions and modules, files, etc.).

### Evaluation of study results

Final written exam (50%), mid-term written exam (17%), and assessments of laboratory (practical) work (33%).

### Distribution of subject study hours

Lectures + laboratory work (practicum)	60
Homework	25
Individual studies (preparation for the mid-term and final exams)	19
<b>Total</b>	<b>104</b>

### Recommended literature

No	Year of publication	Authors and title	Publisher	Number of copies available		
				<i>in the Library of VMU</i>	<i>in specialized publication collections at VMU</i>	<i>in other libraries</i>
<b>Main reading</b>						
1.	2011	The Python Tutorial. Release 3.2	On the internet, <a href="http://docs.python.org/py3k/tutorial/index.html">http://docs.python.org/py3k/tutorial/index.html</a>			
2.	2010	Non-Programmer's Tutorial for Python 3	On the internet, WikiBooks			
3.	2010	Python Programming	On the internet, WikiBooks			
<b>Additional reading</b>						
1.	2010	Chris Meyers. Python for Fun	On the internet, OpenBook project			

### Subject prepared and coordinated by

Dr. Jurgita Kapo i t -Dzikien