



## MODULIO APRAŠAS

Modulio pavadinimas	Kodas
Programavimas Python kalba	

Dėstytojas (-ai)	Padalinys (-iai)
<b>Koordinuojantis:</b> dr. Tomas Plankis	Programų sistemų katedra Matematikos ir informatikos fakultetas Vilniaus universitetas
<b>Kitas (-i):</b>	

Studijų pakopa	Dalyko (modulio) tipas
pirmoji	pasirenkamasis

Igyvendinimo forma	Vykdyto laikotarpis	Vykdyto kalba (-os)
auditorinė	6 semestras (pavasario)	lietuvių

Reikalavimai studijuojančiajam
<b>Išankstiniai reikalavimai:</b> Procedūrinis programavimas, objektinis programavimas

Modulio apimtis kreditais	Visas studento darbo krūvis	Kontaktinio darbo valandos	Savarankiško darbo valandos
5	130	66	64

Modulio tikslas: studijų programos ugdomos kompetencijos									
<p>Modulio tikslas – supažindinti su programavimo Python kalba galimybėmis ir suteikti praktinių įgūdžių.</p> <p><b>Bendrosios kompetencijos:</b></p> <ul style="list-style-type: none"> <li>• Nuolatinis mokymasis (BK2). <ul style="list-style-type: none"> <li>○ Gebės atlikti literatūros paiešką ir analizę, naudoti duomenų bazes ir kitus informacijos šaltinius (BK2.2).</li> <li>○ Gebės savarankiškai įsisavinti naujas žinias, metodus ir įrankius bei taikyti juos praktikoje (BK2.3).</li> </ul> </li> </ul> <p><b>Dalykinės kompetencijos:</b></p> <ul style="list-style-type: none"> <li>• Konceptualių pagrindų žinios ir gebėjimai (DK4). <ul style="list-style-type: none"> <li>○ Supras pagrindines programų sistemų inžinerijos koncepcijas bei sąvokas, įskaitant kelias priešakines sritis, suvoks galimas taikymo sritis ir žinos disciplinos aprėptį (DK4.1).</li> <li>○ Gebės taikyti matematikos pagrindų, mokslo, inžinerijos, kompiuterių mokslo teorines žinias ir algoritminius principus programų sistemų kūrimo (DK4.2).</li> </ul> </li> <li>• Technologinės, metodinės žinios ir gebėjimai, profesinis kompetentingumas (DK6). <ul style="list-style-type: none"> <li>○ Gebės derinti teoriją ir praktiką programų sistemų taikymo įvairiose srityse uždavinių sprendimui, įvertinant technologinį, ekonominį, socialinį ir teisinį kontekstą (DK6.1).</li> </ul> </li> </ul>									
<table border="1"> <thead> <tr> <th>Modulio studijų siekiniai</th> <th>Studijų metodai</th> <th>Vertinimo metodai</th> </tr> </thead> <tbody> <tr> <td>Žinos Python programavimo kalbos sintaksę ir veikimo principus.</td> <td rowspan="4">Demonstravimas, diskusija, probleminis dėstymas, savarankiškas darbas</td> <td rowspan="4">Pratybų užduotys, egzaminas</td> </tr> <tr> <td>Gebės suprasti Python kalba parašytą programą ir ją pakeisti bei vykdyti.</td> </tr> <tr> <td>Gebės rašyti testais grįstas programas.</td> </tr> <tr> <td>Gebės rašyti taikomajai sričiai skirtas programas.</td> </tr> </tbody> </table>	Modulio studijų siekiniai	Studijų metodai	Vertinimo metodai	Žinos Python programavimo kalbos sintaksę ir veikimo principus.	Demonstravimas, diskusija, probleminis dėstymas, savarankiškas darbas	Pratybų užduotys, egzaminas	Gebės suprasti Python kalba parašytą programą ir ją pakeisti bei vykdyti.	Gebės rašyti testais grįstas programas.	Gebės rašyti taikomajai sričiai skirtas programas.
Modulio studijų siekiniai	Studijų metodai	Vertinimo metodai							
Žinos Python programavimo kalbos sintaksę ir veikimo principus.	Demonstravimas, diskusija, probleminis dėstymas, savarankiškas darbas	Pratybų užduotys, egzaminas							
Gebės suprasti Python kalba parašytą programą ir ją pakeisti bei vykdyti.									
Gebės rašyti testais grįstas programas.									
Gebės rašyti taikomajai sričiai skirtas programas.									

Mokės analizuoti pateikto internetinio karkaso struktūrą ir veikimo principus bei atlikti reikalingus kodo pakeitimus.		
Mokės dirbti su duomenų bazėmis.		

Temos	Kontaktinio darbo valandos							Savarankiškų studijų laikas ir užduotys	
	Paskaitos	Konsultacijos	Seminarai	Pratybos	Laboratoriniai darbai	Konsultavimas pratvbiu metu	Visas kontaktinis darbas	Savarankiškas darbas	Užduotys
1. Python interpretatorius, programavimo kalbos sintaksė (procedūrinis programavimas, objektinis programavimas, grafinė aplinka) ir stilius	2				2		4	2	Užduotys: 1) Išplėtimas ir įterpimas 2) Mokslinės bibliotekos 3) Duomenų bazės 4) Testais grįstas programavimas 5) Karkasai
2. Išplėtimas ir įterpimas (extending and embedding)	8				8		16	8	
3. Mokslinės bibliotekos (NumPy, Matplotlib, Mpi4Py ir kt.)	2				2	6	4	4	
4. Darbas su duomenų bazėmis	2				2		4	4	
5. Testais grįstas programavimas	4				4		8	8	
6. Internetinių svetainių karkaso veikimo principai ir pritaikymas	14				14		28	18	
7. Pasiruošimas egzaminui								20	
8. Egzamino laikymas							2		
<b>Iš viso</b>	<b>32</b>				<b>32</b>	<b>6</b>	<b>66</b>	<b>64</b>	

Vertinimo strategija	Svoris proc.	Atsiskaitymo laikas	Vertinimo kriterijai
Pratybų užduotys	50	Semestro metu pagal grafiką	Penkios užduotys. Kiekviena vertinama 10 balų skalėje. Vertinimo kriterijai: korektiškas veikimas (70%), nurodytų sąlygų išpildymas (20%), programos paaiškinimas (10%). Pratybų įvertinimas apskaičiuojamas pagal formulę: $PI = (3*U1+1*U2+1*U3+2*U4+5*U5)/12$
Egzaminas	50	Sesijos metu	Atviro ir/arba uždaro tipo klausimai. Maksimalus įvertinimas 10 balų.

Reikalavimai dalyko vertinimui eksterno būdu	
Įvertinimas galimas eksterno būdu:	Netaikomas

Autorius	Leidimo metai	Pavadinimas	Periodinio leidinio Nr. ar leidinio tomas	Leidimo vieta ir leidykla ar internetinė nuoroda
<b>Privalomoji literatūra</b>				
Allen B. Downey	2013	Think Python		<a href="http://www.greenteapress.com/thinkpython">http://www.greenteapress.com/thinkpython</a> , O'Reilly Media
Harry Percival	2013	Test-Driven Development with Python		<a href="http://chimera.labs.oreilly.com/books/1234000000754/index.html">http://chimera.labs.oreilly.com/books/1234000000754/index.html</a> , O'Reilly Media

Guido van Rossum	2009	Extending and Embedding Python (Release 2.6.3)		<a href="https://infohost.nmt.edu/tcc/help/lang/python/2_6_3/extending.pdf">https://infohost.nmt.edu/tcc/help/lang/python/2_6_3/extending.pdf</a>
<b>Papildoma literatūra</b>				
Python Software Foundation	2013	The Python Standard Library		<a href="https://docs.python.org/3/library">https://docs.python.org/3/library</a>
Python Software Foundation	2012	The Python Tutorial		<a href="https://docs.python.org/3/tutorial">https://docs.python.org/3/tutorial</a>



## COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Programming in Python	

Lecturer(s)	Department where the course unit is delivered
<b>Coordinator:</b> Tomas Plankis	Department of Computer Science II Faculty of Mathematics and Informatics Vilnius University
<b>Other lecturers:</b>	

Cycle	Type of the course unit
First	Optional

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	6th semester	Lithuanian

Prerequisites
Knowledge of procedural and objec-oriented programming

Number of ECTS credits allocated	Student's workload	Contact hours	Individual work
5	130	66	64

Purpose of the course unit: programme competences to be developed		
Purpose – introduction to Python and practical skills.		
<b>Generic competences:</b>		
<ul style="list-style-type: none"> <li>• Life-long learning (BK2).               <ol style="list-style-type: none"> <li>1. Will be able to perform literature search and analysis, and to use databases and other sources of information (BK2.2).</li> <li>2. Will be able independently to acquire new knowledge, methods and tools and apply them in practice (BK2.3).</li> </ol> </li> </ul>		
<b>Subject-specific competences:</b>		
<ul style="list-style-type: none"> <li>• Knowledge and skills of underlying conceptual basis (DK4).               <ol style="list-style-type: none"> <li>1. Will be able to understand the key aspects and concepts of software engineering, including some forefront areas, and have insight into potential application areas and know the scope of the discipline (DK4.1).</li> <li>2. Will be able to apply the theoretical knowledge and algorithmic principles of the basics of mathematics, science and engineering, computer science in the development of software systems (DK4.2).</li> </ol> </li> <li>• Technological and methodological knowledge and skills, professional competence (DK6)               <ol style="list-style-type: none"> <li>1. Will be able to combine the theory and practice of the application of program systems in various fields for problem solving, assessing the technological, economic, social and legal context (DK6.1).</li> </ol> </li> </ul>		
Learning outcomes of the course unit	Teaching and learning methods	Assessment methods
To know Python syntax and work principles	Slides, discussion, problem-based learning, self study	Assignments, exam
To understand Python code and to be able to change and execute it		
To write programs in test-driven manner		

To write programs in the application area									
To analyze given internet application framework and it's work principles and to do necessary code changes									
To work with databases									
Course content: breakdown of the topics	Contact hours						Individual work: time and assignments		
	Lectures	Tutorials	Seminars	Practice	Laboratory work	Consultation during practice	Contact hours	Individual work	Assignments
1. Introduction into Python (interpretator, syntax, style)	2			2		6	4	2	1) Extending and embedding
2. Extending and embedding	8			8			16	8	
3. Scientific modules (NumPy, Matplotlib, Mpi4Py, etc.)	2			2			4	4	2) Scientific modules
4. Databases	2			2			4	4	
5. Test-driven development	4			4			8	8	4) Test-driven development
6. Internet application frameworks	14			14			28	18	
7. Study for exam								20	5) Frameworks
8. Exam						2			
<b>Total</b>	<b>32</b>			<b>32</b>		<b>6</b>	<b>66</b>	<b>64</b>	

Assessment strategy	Weight %	Deadline	Assessment criteria
Assignments	50	Through semester by deadline	Five assignments, 10 points each. Criteria: correctness (70%), fulfillment of conditions (20%), explanation of code (10%). Total value calculated by formula: $PI = (3*A1+1*A2+1*A3+2*A4+5*A5)/12$
Exam	50	Session	Open and/or closed questions. Max. grade - 10 points.

Author	Publishing year	Title	Number or volume	Publisher or URL
<b>Required reading</b>				
Allen B. Downey	2013	Think Python		<a href="http://www.greenteapress.com/thinkpython">http://www.greenteapress.com/thinkpython</a> , O'Reilly Media
Harry Percival	2013	Test-Driven Development with Python		<a href="http://chimera.labs.oreilly.com/books/1234000000754/index.html">http://chimera.labs.oreilly.com/books/1234000000754/index.html</a> , O'Reilly Media
Guido van Rossum	2009	Extending and Embedding Python (Release 2.6.3)		<a href="https://infohost.nmt.edu/tcc/help/lang/python/2_6_3/extending.pdf">https://infohost.nmt.edu/tcc/help/lang/python/2_6_3/extending.pdf</a>
<b>Recommended reading</b>				
Python Software Foundation	2013	The Python Standard Library		<a href="https://docs.python.org/3/library">https://docs.python.org/3/library</a>
Python Software Foundation	2012	The Python Tutorial		<a href="https://docs.python.org/3/tutorial">https://docs.python.org/3/tutorial</a>