

UČNI NAČRT PREDMETA / COURSE SYLLABUS						
<b>Predmet:</b>		Uvod v programiranje				
<b>Course title:</b>		Introduction to programming				
<b>Študijski program in stopnja</b> Study programme and level		<b>Študijska smer</b> Study field		<b>Letnik</b> Academic year	<b>Semester</b> Semester	
Univerzitetni študijski program Matematika		ni smeri		1	drugi	
First cycle academic study programme Mathematics		none		1	second	
<b>Vrsta predmeta / Course type</b>				obvezni		
<b>Univerzitetna koda predmeta / University course code:</b>				M0204		
<b>Predavanja</b> Lectures	<b>Seminar</b> Seminar	<b>Vaje</b> Tutorial	<b>Klinične vaje</b> work	<b>Druge oblike</b> študija	<b>Samost. delo</b> Individ. work	<b>ECTS</b>
30		45			105	6
<b>Nosilec predmeta / Lecturer:</b>		doc. Matija Pretnar, prof. Andrej Bauer, prof. Marko Petkovšek				
<b>Jeziki / Languages:</b>		<b>Predavanja / Lectures:</b> slovenski/Slovene				
		<b>Vaje / Tutorial:</b> slovenski/Slovene				
<b>Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:</b>				<b>Prerequisites:</b>		
Opravljen predmet Računalniški praktikum.				Completed course Computer laboratory.		
<b>Vsebina:</b>				<b>Content (Syllabus outline):</b>		
Zgradba računalnika in pojem algoritma. Osnovni koncepti v programiranju: spremenljivke, aritmetika, osnovni podatkovni tipi. Strukturirano programiranje: funkcije,				The structure of a computer and the concept of an algorithm. Basic programming concepts: variables, arithmetic, basic data types. Structured programming: functions, conditional		

<p>pogojni stavek, rekurzija, zanke.</p> <p>Podatkovni tipi: tabele, slovarji, objekti, datoteke. Organizacija izvorne kode. Orodja za razvoj programske opreme. Osnove računske zahtevnosti algoritmov.</p>	<p>statement, recursion, loops.</p> <p>Data types: arrays, dictionaries, objects, files. Managing source code. Software development tools. Basics of computational complexity.</p>
--	--

**Temeljni literatura in viri / Readings:**

<p>Priročniki in učbeniki za programske jezike, ki jih študenti spoznajo.</p> <p>Manuals and textbooks for programming languages that students learn.</p>
---

**Cilji in kompetence:**

<p>Študent spozna osnovne tehnike programiranja.</p>
--

**Objectives and competences:**

<p>Student learns the basic programming techniques.</p>
---

**Predvideni študijski rezultati:**

**Intended learning outcomes:**

Znanje in razumevanje: Znanje iz osnov programiranja.  
 Uporaba: Reševanje matematičnih in drugih problemov z računalnikom, predvsem kadar je treba za rešitev problema sestaviti preprost program.

Refleksija: Sposobnost programiranja omogoča uporabniku višji nivo nadzora nad računalnikom in mu omogoča reševanje problemov, ki jih z običajnimi aplikacijami ne more rešiti.

Prenosljive spretnosti – niso vezane le na en predmet: Programerske sposobnosti študent uporabi pri ostalih računalniških in numeričnih predmetih.

Knowledge and understanding: Knowledge of basic programming.  
 Application: Solving mathematical and other problems with a computer, in particular when a simple computer program is required for this task.

Reflection: The ability of programming enables a higher lever of control over the computer and enables the student to solve the problems that cannot be solved using the standard applications.

Transferable skills: The skill of programming is required in other computer and numerical courses.

**Metode poučevanja in učenja:**

Predavanja, vaje, domače naloge, konzultacije

**Learning and teaching methods:**

Lectures, exercises, homework, consultations

Delež (v %) /

**Načini ocenjevanja:**

domače naloge, kolokviji, projekti, pisni izpit, ustni izpit  
 ocene: 5 (negativno), 6-10 (pozitivno) (po Statutu UL)

Weight (in %)

**Assessment:**

homework, midterm exams, projects, written exam, oral exam  
 grading: 5 (fail), 6-10 (pass) (according to the Statute of UL)

100%

**Reference nosilca / Lecturer's references:**

Andrej Bauer:  
 – BAUER, Andrej. A relationship between equilogical spaces and Type Two Effectivity.

Mathematical logic quarterly, ISSN 0942-5616, 2002, vol. 48, suppl. 1, str. 1-15 [COBISS.SI-ID 12033369]

– AWODEY, Steve, BAUER, Andrej. Propositions as [Types]. Journal of logic and computation, ISSN 0955-792X, 2004, vol. 14, no. 4, str. 447-471 [COBISS.SI-ID 13374809]

– BAUER, Andrej, SIMPSON, Alex. Two constructive embedding-extension theorems with applications to continuity principles and to Banach-Mazur computability. Mathematical logic quarterly, ISSN 0942-5616, 2004, vol. 50, no. 4/5, str. 351-369 [COBISS.SI-ID 13378649]

Marko Petkovšek:

– PETKOVŠEK, Marko, WILF, Herbert S., ZEILBERGER, Doron. A=B. Wellesley (Massachusetts): A. K. Peters, cop. 1996. VII, 212 str. ISBN 1-56881-063-6 [COBISS.SI-ID 4085337]

– PETKOVŠEK, Marko. Counting Young tableaux when rows are cosets. Ars combinatoria, ISSN 0381-7032, 1994, let. 37, str. 87-95 [COBISS.SI-ID 8048473]

– PETKOVŠEK, Marko. Letter graphs and well-quasi-order by induced subgraphs. Discrete Mathematics, ISSN 0012-365X. [Print ed.], 2002, vol. 244, no. 1-3, str. 375-388 [COBISS.SI-ID 11414873]

Matija Pretnar:

– PLOTKIN, Gordon, PRETNAR, Matija. Handling algebraic effects. Logical methods in computer science, ISSN 1860-5974, 2013, vol. 9, iss. 4, paper 23 (str. 1-36) [COBISS.SI-ID 16816729]

– PRETNAR, Matija. Inferring algebraic effects. Logical methods in computer science, ISSN 1860-5974, 2014, vol. 10, iss. 3, paper 21 (str. 1-43) [COBISS.SI-ID 17190745]

– BAUER, Andrej, PRETNAR, Matija. An effect system for algebraic effects and handlers. Logical methods in computer science, ISSN 1860-5974, 2014, vol. 10, iss. 4, paper 9 (str. 1-29). <http://arxiv.org/pdf/1306.6316> [COBISS.SI-ID 17191001]