

UČNI NAČRT PREDMETA / COURSE SYLLABUS						
<b>Predmet:</b>		Programiranje 1				
<b>Course title:</b>		Programming 1				
<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		2	prvi	
First cycle academic study programme Mathematics		none		2	first	
<b>Vrsta predmeta / Course type</b>				obvezni		
<b>Univerzitetna koda predmeta / University course code:</b>				M0212		
<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		30			90	5
<b>Nosilec predmeta / Lecturer:</b>		doc. Matija Pretnar, prof. Andrej Bauer				
<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 Uvod v programiranje.				Completed course Introduction to programming.		
<b>Vsebina:</b>				<b>Content (Syllabus outline):</b>		
Uporaba kompleksnih podatkovnih tipov in implementacija podatkovnih struktur: seznam, sklad, vrsta, drevo, ...				Use of complex data types and implementation of data structures: list, stack, queue, tree, ...		

<p>Algoritmi za iskanje in urejanje. Računska zahtevnost in preverjanje pravilnosti algoritmov. Osnovne metode razvoja algoritmov: deli in vladaj, dinamično programiranje, sestopanje, ...</p> <p>Zajem in čiščenje podatkov. Poizvedbe, analiza in prikaz podatkov.</p>	<p>Search and sorting algorithms. Computational complexity and correctness analysis of programs. Basic methods of algorithm design: divide and conquer, dynamic programming, backtracking, ...</p> <p>Obtaining and cleaning data. Queries, analysis and presentation of data.</p>
---	--

**Temeljni literatura in viri / Readings:**

<p>Priročniki in učbeniki o programiranju v programskem jeziku, ki ga študenti spoznajo. Manuals and textbooks on programming in appropriate programming language.</p>
--

**Cilji in kompetence:**

<p>Študent spozna osnove strukturiranega programiranja in nekatere osnovne podatkovne strukture in algoritme.</p>
---

**Objectives and competences:**

<p>A student gets familiar with the basics of structured programming and certain basic data structures and algorithms.</p>
--

**Predvideni študijski rezultati:**

<p>Znanje in razumevanje: Osnovno znanje vsaj enega programskega jezika in osnovnih programerskih tehnik.</p> <p>Uporaba: Programerske veščine študent uporabi pri ostalih predmetih, ki zahtevajo reševanje problemov z računalnikom.</p> <p>Refleksija: Strukturirano programiranje omogoča abstrakten in konceptualno prečiščen</p>
--

**Intended learning outcomes:**

<p>Knowledge and understanding: Basic knowledge of at least one programming language and basic programming techniques.</p> <p>Application: A student can use the skills of programming at other courses that require problem solving using a computer.</p> <p>Reflection: Structured programming enables abstract and conceptually clean approach to</p>
--

pristop k programiranju, ki sloni na diskretnih strukturah.

Prenosljive spretnosti – niso vezane le na en predmet: Programerske sposobnosti so uporabne pri nadaljnjih računalniških predmetih, kakor tudi pri numeričnih metodah.

programming that is based on discrete structures.

Transferable skills: Programming capabilities are useful in further computer science courses as well as the courses on numerical methods.

**Metode poučevanja in učenja:**

Predavanja, vaje, domače naloge, konzultacije

**Learning and teaching methods:**

Lectures, exercises, homework, consultations

**Načini ocenjevanja:**

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

Delež (v %) /  
Weight (in %)

100%

**Assessment:**

homework, midterm exams, projects, written exam, oral exam

grading: 5 (fail), 6-10 (pass) (according to the Statute of UL)

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

Andrej Bauer:

- BAUER, Andrej, BIRKEDAL, Lars. Continuous functionals of dependent types and equilogical spaces. V: CLOTE, Peter G. (ur.). Computer science logic : 14th international workshop, CSL 2000, annual conference of the EACSL, Fischbachau, Germany, August 21-26, 2000 : proceedings, (Lecture notes in computer science, ISSN 0302-9743, 1862). Berlin [etc.]: Springer, 2000, vol. 1862, str. 202-216 [COBISS.SI-ID 10606681]
- BAUER, Andrej. Uvod v programiranje v Javi. Ljubljana: [A. Bauer], 2008. 1 optični disk (CD-ROM) [COBISS.SI-ID 14629977]

- BAUER, Andrej. Teorija programskih jezikov. Ljubljana: [A. Bauer], 2007. 100 str [COBISS.SI-ID 14630489]
- BAUER, Andrej, TAYLOR, Paul. The Dedekind reals in abstract Stone duality. Mathematical structures in computer science, ISSN 0960-1295, 2009, vol. 19, iss. 4, str. 757-838 [COBISS.SI-ID 15322201]
- BAUER, Andrej, STONE, Christopher A. RZ: a tool for bringing constructive and computable mathematics closer to programming practice. Journal of logic and computation, ISSN 0955-792X, 2009, vol. 19, no. 1, str. 17-43 [COBISS.SI-ID 15325785]

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]