UČNI NAČRT PREDMETA / COURSE SYLLABUS

Predmet:	Programiranje 1
Course title:	Programming 1

Študijski program in stopnja **Study programme and level**	**Študijska smer** **Study field**	**Letnik** **Academic year**	**Semester** **Semester**
Enoviti magistrski študijski program Pedagoška matematika	ni smeri	3 ali 4	prvi
Integrated Master's study programme Pedagogical Mathematics	none	3 or 4	first

Vrsta predmeta / Course type	izbirni

Univerzitetna koda predmeta / University course code:	M0534

Predavanja **Lectures**	**Seminar** **Seminar**	**Vaje** **Tutorial**	**Klinične vaje** **work**	**Druge oblike študija**	**Samost. delo** **Individ. work**	**ECTS**
30		**30**			**90**	**5**

Nosilec predmeta / Lecturer:	doc. Matija Pretnar, prof. Andrej Bauer

Jeziki / **Languages:**	**Predavanja /** **Lectures:**	slovenski/Slovene
	Vaje / Tutorial:	slovenski/Slovene

Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:	**Prerequisites:**
Opravljen predmet Uvod v programiranje.	Completed course Introduction to programming.

Vsebina:	**Content (Syllabus outline):**

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

Temeljni literatura in viri / Readings:

Priročniki in učbeniki o programiranju v programskem jeziku, ki ga študenti spoznajo.

Manuals and textbooks on programming in appropriate programming language.

Cilji in kompetence:	**Objectives and competences:**
Študent spozna osnove strukturiranega programiranja in nekatere osnovne podatkovne strukture in algoritme.	A student gets familiar with the basics of structured programming and certain basic data structures and algorithms.

Predvideni študijski rezultati:	**Intended learning outcomes:**
Znanje in razumevanje: Osnovno znanje vsaj enega programskega jezika in osnovnih programerskih tehnik.	Knowledge and understanding: Basic knowledge of at least one programming language and basic programming techniques.
Uporaba: Programerske veščine študent uporabi pri ostalih predmetih, ki zahtevajo reševanje problemov z računalnikom.	Application: A student can use the skills of programming at other courses that require problem solving using a computer.

Refleksija: Strukturirano programiranje omogoča abstrakten in konceptualno prečiščen pristop k programiranju, ki sloni na diskretnih strukturah.

Reflection: Structured programming enables abstract and conceptually clean approach to programming that is based on discrete structures.

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

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:	Delež (v %) / Weight (in %)	**Assessment:**
domače naloge, kolokviji, projekti, pisni izpit, ustni izpit ocene: 5 (negativno), 6-10 (pozitivno) (po Statutu UL)		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, 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]