

UČNI NAČRT PREDMETA / COURSE SYLLABUS (leto / year 2016/17)						
Predmet:		Uvod v programiranje				
Course title:		Introduction to programming				
Študijski program in stopnja Study programme and level		Študijska smer Study field		Letnik Academic year		Semester Semester
Univerzitetni študijski program Matematika		ni smeri		1		drugi
First cycle academic study programme Mathematics		none		1		second
Vrsta predmeta / Course type				obvezni / compulsory		
Univerzitetna koda predmeta / University course code:				M0204		
Predavanja Lectures	Seminar Seminar	Vaje Tutorial	Klinične vaje work	Druge oblike študija	Samost. delo Individ. work	ECTS
30		45			105	6
Nosilec predmeta / Lecturer:				prof. dr. Andrej Bauer, prof. dr. Marko Petkovšek		
Jeziki / Languages:		Predavanja / Lectures:		slovenski / Slovene		
		Vaje / Tutorial:		slovenski / Slovene		
Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:				Prerequisites:		
Vpis v letnik študija.				Enrolment in the programme.		
Opravljen predmet Računalniški praktikum.				Completed course Computer practical.		
Vsebina:				Content (Syllabus outline):		

Pojem algoritma. Osnovni koncepti v programiranju: spremenljivke, osnovni in številski tipi, aritmetika, pogojni stavki in Boolove vrednosti, zanke, tabele, datoteke, podprogrami.

Uvod v strukturirano programiranje: funkcijsko programiranje (funkcije, funkcije višjih redov, rekurzivne funkcije, paralelizacija), podatkovni tipi (uporabniški podatkovni tipi, skladi, vrste, sezname, drevesa), objektno programiranje.

The concept of algorithm. Basic programming concepts: variables, basic and numeric types, arithmetic, conditional statements and Boolean values, loops, arrays, files, subroutines.

Introduction to structured programming: functional programming (functions, higher-order functions, recursive functions, parallelization), data types (user-defined data types, stack, queue, list, tree), object-oriented programming.

Temeljni literatura in viri / Readings:

Priročniki in učbeniki za programske jezike, ki jih študenti spoznajo.

Manuals and textbooks for programming languages that students learn.

Cilji in kompetence:

Študent spozna osnovne tehnike programiranja.

Objectives and competences:

Student learns the basic programming techniques.

Predvideni študijski rezultati:

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.

Intended learning outcomes:

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

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

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

Načini ocenjevanja:	Delež (v %) / Weight (in %)	Assessment:
Možna sta dva načina ali kombinacija obeh po odločitvi izvajalca:	50%	Two approaches are possible or a combination of both (teacher's decision):
1. način (sprotno ustno izpraševanje na vajah, končni ustni izpit):	50%	
Študent na vsakih vajah odda izdelek. Asistent ga pregleda in na naslednjih vajah študenta še na osnovi izdelka ustno izpraša ter oceni. Na osnovi ocen iz vaj se določi končna ocena iz vaj. Zahteva se vsaj 85% prisotnost na vajah.	50%	The student submits the result of his/her work after each exercise (lab) session. Teaching assistant reviews the work, examines (orally) the student during the next exercise session and grades him/her. The final grade is determined from the grades of each exercise. The required presence at the exercise sessions is 85%
Izpit iz teorije	50%	
2. način (pisni izpit, ustno izpraševanje, naloge, projekt):		(final) oral exam
2 kolokvija namesto izpita iz vaj, izpit iz vaj, opsijski manjši projekt (seminarska naloga)		Approach 2: (examination, oral, coursework, project):
izpit iz teorije.		2 midterm exams instead of written exam, written exam
Splošno:		oral exam
ocene: 1-5 (negativno), 6-10 (pozitivno)		General:

(po Statutu UL)		grading: 1-5 (fail), 6-10 (pass) (according to the Statute of UL)
-----------------	--	---

Reference nosilca / Lecturer's references:

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]

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]

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, 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. 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]