UČNI NAČRT PREDMETA / COURSE SYLLABUS (leto / year 2016/17)			
Predmet:	Programiranje 2		
Course title:	Programming 2		
Študijski program in stopnja **Study programme and level**	**Študijska smer** **Study field**	**Letnik** **Academic year**	**Semester** **Semester**
Interdisciplinarni univerzitetni študijski program Računalništvo in matematika	ni smeri	1	drugi
Interdisciplinary first cycle academic study programme Computer Science and Mathematics	none	1	second

Vrsta predmeta / Course type	obvezni / compulsory
Univerzitetna koda predmeta / University course code:	63278

Predavanja **Lectures**	**Seminar** **Seminar**	**Vaje** **Tutorial**	**Klinične vaje** **work**	**Druge oblike študija**	**Samost. delo** **Individ. work**	**ECTS**
45		**30**			**105**	**6**

Nosilec predmeta / Lecturer:	doc. dr. Boštjan Slivnik

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

Uvod v programski jezik C. Osnovni podatkovni tipi in osnovne kontrolne strukture. Razvoj programov in razhroščevanje. Kazalci in tabele (1. del). Kazalci in tabele (2. del). Funkcije in prenos argumentov. Dinamično dodeljevanje pomnilnika. Nizi. Vhodno-izhodne operacije. Strukture. Enostavni algoritmi urejanja. Rekurzija in algoritmi sestopanja (1. del). Rekurzija in algoritmi sestopanja (2. del). Predprocesor.	Introduction to C programming language. Basic data types and basic control structures. Program design and debugging. Pointers and arrays (part 1). Pointers and arrays (part 2). Functions and parameter passing. Dynamic memory allocation. Strings. Input-output operations. Structures. Simple sorting algorithms. Recursion and backtracking (part 1). Recursion and backtracking (part 2). Preprocessor.

Temeljni literatura in viri / Readings:

B. W. Kernighan, D. Ritchie: Programski jezik C, Fakulteta za računalništvo in informatiko, 1994.

T. Dobravec: abC, Fakulteta za računalništvo in informatiko, 2010.

A. Kavčič, M. Privošnik, C. Bohak, M. Marolt, S. Divjak: Programiranje in algoritmi skozi primere, Založba FE in FRI, 2010

Cilji in kompetence:	Objectives and competences:
Cilj predmeta je razširiti znanje programiranja skozi študij bazičnih in naprednejših tehnik programiranja. Splošne kompetence: Sposobnost kritičnega, analitičnega in sintetičnega razmišljanja Sposobnost razumevanja in reševanja strokovnih izzivov v računalništvu in informatiki Sposobnost uporabe pridobljenega znanja pri samostojnem reševanju tehničnih in znanstvenih problemov v računalništvu in informatiki, sposobnost razširjanja pridobljenega znanja Predmetno-specifične kompetence: Osnovne veščine v računalništvu in informatiki – osnovne teoretične veščine, praktično znanje, bistvene veščine za področje računalništva in informatiki Osnovne veščine v računalništvu in informatiki, ki omogočajo nadaljevanje študija na drugi stopnji	The goal of the course is to widen the programming skills by learning the most basic and advanced programming techniques. General competences: Developing skills in critical, analytical and synthetic thinking The ability to understand and solve professional challenges in computer and information science The ability to apply acquired knowledge in independent work for solving technical and scientific problems in computer and information science, the ability to upgrade acquired knowledge Subject-specific competences: Basic skills in computer and information science, which includes basic theoretical skills, practical knowledge and skills essential for the field of computer and information science Basic skills in computer and information science, allowing the continuation of studies in the second study cycle
Predvideni študijski rezultati:	**Intended learning outcomes:**
Znanje in razumevanje: postopka priprave in izvajanja programa, osnovnih programskih konstruktov in podatkovnih struktur, konceptov objektno usmerjenega programiranja, osnovnih algoritmov za reševanje tipičnih programerskih problemov,	Knowledge and understanding: The ability to independently develop programs, the familiarity with basic data structures and algorithms. Application: Writing simple programs for solving moderate programming problems.

dogodkovno vodenega programiranja in osnovnih komponent uporabniškega vmesnika. Uporaba: Uporaba naučenih konceptov pri samostojnem razvoju enostavnejših računalniških programov. Refleksija: Spoznavanje in razumevanje vloge programerja pri reševanju problemov različnih uporabnikov. Prenosljive spretnosti - niso vezane le na en predmet: Spretnost pri uporabi domače in tuje literature ter uporabniške dokumentacije. Pridobivanje sposobnosti za samostojno reševanje problemov.	Reflection: Understanding the basic principles of designing programs and algorithms and understanding their role in the development of large systems. Transferable skills: The ability to design the solution of different problems using programs and algorithms, the ability to use the presented programming concepts in an arbitrary programming language.

Metode poučevanja in učenja:	**Learning and teaching methods:**
Predavanja, domače naloge, seminarski način dela pri vajah. Poseben poudarek je na sprotnem študiju in domačih nalogah.	Lectures, home works, seminar works during tutorials. The emphasis is on continuous study and homeworks.

Načini ocenjevanja:	Delež (v %) / Weight (in %)	**Assessment:**
Način (pisni izpit, ustno izpraševanje): Sprotno preverjanje (kolokviji) Končno preverjanje (pisni izpit) Ocene: 6-10 pozitivno, 1-5 negativno (v skladu s Statutom UL)	 50% 50%	Type (examination, oral, coursework, project): Continuing (three homework programming projects, short tests) Final (exam) Grading: 6-10 pass, 1-5 fail.

Reference nosilca / Lecturer's references:

ROŽANC, Igor, SLIVNIK, Boštjan. Using reverse engineering to construct the platform independent model of a web application for student information systems. Computer science and information systems, ISSN 1820-0214. [Print ed.], 2013, vol. 10, no. 4, str. 1557-1583, ilustr. , doi: . [COBISS.SI-ID 10226516]

CANKAR, Matija, ARTAČ, Matej, ŠTERK, Marjan, LOTRIČ, Uroš, SLIVNIK, Boštjan. Co-allocation with collective requests in grid systems. Journal for universal computer science, ISSN 0948-6968, 2013, vol. 19, no. 3, str. 282-300, ilustr. [COBISS.SI-ID 9797972]

SLIVNIK, Boštjan. LL conflict resolution using the embedded left LR parser. Computer science and information systems, ISSN 1820-0214. [Print ed.], Sep. 2012, vol. 9, no. 3, str. 1105-1124, ilustr. [COBISS.SI-ID 9583700]

SLIVNIK, Boštjan, VILFAN, Boštjan. Producing the left parse during bottom-up parsing. Information processing letters, ISSN 0020-0190. [Print ed.], Dec. 2005, vol. 96, no. 6, str. [220]-224. [COBISS.SI-ID 5075284]

POTOČNIK, Matic, ČIBEJ, Uroš, SLIVNIK, Boštjan. Linter - a tool for finding bugs and potential problems in Scala code. V: Proceedings of the 29th Annual ACM Symposium on Applied Computing, Gyeongju, Korea, March 24-28, 2014, Proceedings of the 29th Annual ACM Symposium on Applied Computing, Gyeongju, Korea, March 24-28, 2014. [S. l.]: Association for Computing Machinery, cop. 2014, str. 1615-1616, graf. prikazi. [COBISS.SI-ID 10520660]

SLIVNIK, Boštjan. LLLR parsing. V: Proceedings of the 28th annual ACM Symposium on Applied Computing 2013, Coimbra, Portugal, March 18-22. [S. l.]: Association for Computing Machinery, 2013, str. 1698-1699. [COBISS.SI-ID 9735508]