UČNI NAČRT PREDMETA / COURSE SYLLABUS (leto / year 2017/18)			
Predmet:	Prevajalniki		
Course title:	Compilers		
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	3	drugi
Interdisciplinary first cycle academic study programme Computer Science and Mathematics	none	3	second

Vrsta predmeta / Course type	izbirni / elective
Univerzitetna koda predmeta / University course code:	63265

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:razbitje prevajalnika na prednji in zadnji del, zgradba prevajalnika kot cevovoda, izbira prevajanega programskega jezika in ciljnega zbirnika. Leksikalna analiza:opis simbolov programskega jezika z regularnimi izrazi in razbitje prevajanega programa na osnovne simbole,- domača naloga: izdelava leksikalnega analizatorja na osnovi končnih avtomatov. Sintaksna analiza:opis sintakse s kontekstno neodvisno gramatiko, postopek sintaksne analize in reševanje iz napak med sintaksno analizo,- domača naloga: izdelava sintaksnega analizatorja na osnovi skladovnega avtomata po algoritmu LR. Abstraktna sintaksa:poenostavljena interna predstavitev prevajanega programa,- domača naloga: generiranje abstraktnega sintaksnega drevesa prevajanega programa. Semantična analiza:analiza podatkovnih tipov, (ne)dosegljivosti kode,…,- domača naloga: izdelava semantičnega analizatorja za preverjanje tipov. Klicni zapisi:klicni zapisi za aktivacijo (gnezdenih, rekurzivnih) podprogramov, uporaba sklada ali kopice za realizacijo klicnih zapisov,- domača naloga: načrt klicnih zapisov. Vmesna koda:drevesna ali ukazna vmesna koda, uporaba začasnih spremenljivk, nivoji vmesne kode, prevod v vmesno kodo,- domača naloga: izdelava generatorja vmesne kode. Osnovni bloki:kanonizacija klicev in skokov v vmesni kodi, oblikovanje osnovnih blokov, permutacija osnovnih blokov,- domača naloga: izračun osnovnih blokov. Izbira strojnih ukazov:prevod vmesne kode v	Introduction:Decomposition of a compiler into front end and back end. Compiler as a staged pipeline. Choosing the source program language and the target assembler. Lexical analysis:describing programming language symbols with regular expressions, breaking the compiled program into lexical tokens Homework: construction of lexical analyzer based on finite automata. Parsing:describing syntax with a context-free grammar, parsing procedure and error recoveryHomework: construction of stack-based LR(k) syntax analyzer Abstract syntax:simplified internal representation of the compiled programHomework: generating an abstract syntax tree of the compiled program. Semantic analysis:type checking, unreachable code detection,…Homework: construction of semantic analyzer for type-checking. Activation records:description of records for activation of nested or recursive functions, and their implementation with stack or heap.Homework: activation records design Intermediate code:tree- or instruction-based intermediate code, temporary variables, translation to intermediate code.Homework: construction of intermediate code generator Basic blocks:canonization of calls and jumps in intermediate code, grouping of statements into basic blocks, permutation of basic blocksHomework: formation of basic blocks Instruction selection:translation of intermediate code to target assembler using only temporary variablesHomework: target code generator

ukaze zbirnika z uporabo začasnih spremenljivk,- domača naloga: generator strojne kode brez registrov. Analiza aktivnosti začasnih spremenljivk:analiza aktivnosti začasnih spremenljivk na osnovi grafov poteka in podatkovnih enačb,- domača naloga: izračun interferenčnega grafa spremenljivk. Izbira registrov:barvanje interferenčnega grafa in izračun preliva začasnih spremenljivk v klicni zapis,- domača naloga: izračun preslikave začasnih spremenljivk v registre in preliv. Zaključek:domača naloga: združitev prvih desetih domačih nalog v delujoč prevajalnik.	(without register allocation) Liveness analysis:activity analysis of temporary variables based on flow graphs and dataflow equations.Homework: construction of a flow graph. Register allocation:coloring of inference graphs, spilling temporary variables into activation records.Homework: allocation of registers to temporary variables and spilling. Conclusion: Homework: integration of earlier homework into a working compiler.

Temeljni literatura in viri / Readings:

Andrew W. Appel, Modern Compiler Implementation in Java, Cambridge University Press, 2002.

Boštjan Vilfan, Prevajanje programskih jezikov, 1. del, Fakulteta za elektrotehniko in računalništvo, 1991.

Steven Muchnick, Advanced Compiler Design and Implementation, Morgan Kaufmann, 1997.

Cilji in kompetence:	**Objectives and competences:**
Predstavitev zgradbe, delovanja in izdelave prevajalnika za prevajanje programskih jezikov v zbirnik. Splošne kompetence: Sposobnost razumevanja in reševanja strokovnih izzivov v računalništvu in informatiki	Presentation of compiler architecture and functional parts, as well as construction and implementation of a working compiler from a chosen programming language into assembler. General competences: The ability to understand and solve professional challenges in computer and information science The ability to define, understand and solve creative professional challenges in computer and

Sposobnost definiranja, 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: Praktično znanje in veščine s področja strojen in programske opreme ter informacijske tehnologije, ki so potrebne za uspešno strokovno delo v računalništvu in informatiki Sposobnost samostojnega izvajanja enostavnih in zahtevnih opravil v določenih ožjih področjih in samostojno reševanje specifičnih dobro definiranih opravil v računalništvu in informatiki Osnovne veščine v računalništvu in informatiki, ki omogočajo nadaljevanje študija na drugi stopnji	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: Practical knowledge and skills of computer hardware, software and information technology necessary for successful professional work in computer and information science The ability to independently perform both less demanding and complex engineering and organisational tasks in certain narrow areas and independently solve specific well-defined tasks in 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: Poznavanje sodobnih postopkov razvoja programske opreme in razumevanje njihovega izvora ter medsebojne povezanosti. Uporaba: Uporaba inženirskih metod pri razvoju programske opreme. Refleksija: Razumevanje primernosti uporabe določenih postopkov razvoja programske opreme glede na tip in zahteve.	Knowledge and understanding: Understanding the workings of a modern compiler implies familiarity with algorithms for syntax and semantic program analysis, generation of intermediate and target machine code, as well as awareness of compilers' limitations. By knowing all this, one also knows and understands how compiled programs work. Application: Compiler is a fundamental software development tool, and therefore the acquired knowledge is (explicitly or implicitly) useful in all programming tasks.

Prenosljive spretnosti - niso vezane le na en predmet: Poznavanje in uporaba metod za delo v skupini, ki rešuje intelektualno zahtevne naloge, trening učinkovitega pisnega in ustnega sporazumevanja s sodelavci.	Reflection: Understanding of relations between writing programs and their execution. Transferable skills: Algorithms for analysis of structured texts, writing efficiently coded programs.

Metode poučevanja in učenja:	**Learning and teaching methods:**
Predavanja in domače naloge (seminarski način dela). Poseben poudarek je na sprotnem oddajanju domačih nalog.	Lectures and homework with explicit focus on simultaneous studies (for homeworks).

Načini ocenjevanja:	Delež (v %) / Weight (in %)	**Assessment:**
Način (pisni izpit, ustno izpraševanje, naloge, projekt): Sprotno preverjanje (domače naloge) Končno preverjanje (pisni in ustni izpit) Ocene: 6-10 pozitivno, 1-5 negativno (v skladu s Statutom UL)	 50% 50%	Type (examination, oral, coursework, project): Continuing (homework, midterm exams, project work) Final (written exam) Grading: 6-10 pass, 1-5 fail.

Reference nosilca / Lecturer's references:

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]

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]

SLIVNIK, Boštjan. The embedded left LR parser. V: GANZHA, Maria (ur.), MACIASZEK, Leszek (ur.), PAPRZYCKI, Marcin (ur.). FedCSIS : proceedings of the Federated Conference on Computer Science and Information Systems, September 18-21, 2011, Szczecin, Poland. Los Alamitos: IEEE Computer Society Press, 2011, str. 871-878, graf. prikazi. [COBISS.SI-ID 8628564]

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]

SLIVNIK, Boštjan, VILFAN, Boštjan. Improved error recovery in generated LR parsers. Informatica, ISSN 0350-5596, 2004, vol. 28, no. 3, str. 257-263, ilustr. [COBISS.SI-ID 4902484]