

UČNI NAČRT PREDMETA / COURSE SYLLABUS (leto / year 2017/18)						
Predmet:	Programiranje 1					
Course title:	Programming 1					
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	prvi	
Interdisciplinary first cycle academic study programme Computer Science and Mathematics	none			1	first	
Vrsta predmeta / Course type				obvezni / compulsory		
Univerzitetna koda predmeta / University course code:				63277		
Predavanja Lectures	Seminar Seminar	Vaje Tutorial	Klinične vaje work	Druge oblike študija	Samost. delo Individ. work	ECTS
45		30			105	6
Nosilec predmeta / Lecturer:						
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):		

predavanja:

Pregled programskih jezikov s poudarkom na sodobnih programskih jezikih 3. generacije

Primer preprostega programa, postopek priprave programa, prevajanje in izvajanje

Osnovni podatkovni tipi

Predstavitev celih in realnih števil, znakov ter logičnih vrednosti

Deklaracije konstant in spremenljivk

Stavki

Prireditveni stavek, pisanje izrazov, operatorji in njihova prioriteta

Izbirni stavki

Ponavljalni stavki

Metode

Vloga in vrste metod

Formalni in dejanski parametri

Rekurzija

Razredi in objekti

Spremenljivke in metode objekta

Konstruktorji, večkratno definiranje konstruktorjev

Dostopna določila

Vnaprej deklarirani razredi

lectures:

Overview of programming languages with emphasis on 3rd generation

Writing, compiling and executing a simple computer program

Basic data types

Integer and real numbers, characters and logical values

Constants and variables declaration

Programming statements

The assignment statement, expressions, operators, operator precedence

Conditional statements

Loops

Methods

The role and classification of methods

Formal and actual parameters

Recursion

Classes and objects

Instance variables and methods

Constructors, overloading constructors

Access modifiers

Using predefined classes

Arrays

Tabele	One-dimensional and multidimensional arrays
Enodimenzionalne in večdimenzionalne tabele	Arrays of objects
Tabele objektov	Strings
Nizi	Inheritance
Dedovanje	Overriding superclass methods
Redefinicija metod	Using superclass constructors in subclasses
Konstruktorji v podrazredih	Dynamic method binding
Dinamično povezovanje podprogramov	Abstract classes and abstract methods
Abstraktni razredi in abstraktne metode	The Object class
Razred Object	Creating and using interfaces
Vmesnik (interface)	Graphics and GUI widget toolkits, components of GUI
Grafične knjižnice, komponente grafičnega uporabniškega vmesnika	Event driven programming
Dogodkovno vodeno programiranje	lab practice:
vaje:	Students solve practical problems to reinforce the understanding of topics covered during lectures. Individual work under the guidance of teaching assistants is emphasized.
Na vajah študenti rešujejo praktične probleme, s katerimi utrjujejo snov, ki so jo obravnavali na predavanjih. Poudarek je na samostojnem delu ob pomoči asistentov.	homework:
domače naloge:	Students are given a list of programs that must be developed outside contact hours and submitted for evaluation within prescribed deadlines, thus preventing them from procrastinating and encouraging self-reliance. Completion of these assignments is a prerequisite for entering the exam.
Študenti dobijo seznam nalog (programov), ki jih morajo izdelati doma in zagovarjati na vajah v vnaprej predpisanih rokih. S tem jih vzpodbujamo k sprotnemu študiju in samostojnemu delu. Študent, ki nima pozitivno ocenjenih domačih nalog, ne more pristopiti k izpitu.	

Temeljni literatura in viri / Readings:

V. Mahnič, L. Fürst, I. Rožanc: Java skozi primere, Bi-TIM, 2008.

J. Farrell: Java Programming, Seventh Edition, Course Technology, Cengage Learning, 2014.

Dodatna literatura:

I. Horton: Beginning Java, Java 7 Edition, John Wiley & Sons, Inc., 2011

Uroš Mesojedec, Borut Fabjan: Java 2: Temelji programiranja, Pasadena, 2004.

Cilji in kompetence:

Cilj predmeta je študentom predstaviti osnovne koncepte objektno usmerjenega programiranja v enem izmed splošno namenskih programskih jezikov 3. generacije in jih usposobiti za samostojen razvoj enostavnih računalniških programov.

Predvidene kompetence:

razvoj sposobnosti za kritično, analitično in sintetično razmišljanje,

sposobnost razumevanja in reševanja strokovnih izzivov na področju računalništva in informatike,

sposobnost uporabe pridobljenega znanja pri samostojnem delu za reševanje tehničnih in znanstvenih problemov na področju računalništva in informatike, sposobnost nadgradnje pridobljenega znanja,

temeljna znanja na področju računalništva in informatike, ki vključujejo temeljna teoretična znanja, praktična znanja in znanja, ki so bistvena za področje računalništva in informatike,

temeljna znanja na področju računalništva in informatike, ki so pomembna za nadaljevanje

Objectives and competences:

The main objective is to teach students basic concepts of object-oriented programming in a general-purpose 3rd generation programming language, thus making them able to develop computer programs of low complexity.

The competences students gain are:

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.

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.

študija na drugi stopnji.

Predvideni študijski rezultati:

Znanje in razumevanje:

Razumevanje delovanja enostavnih digitalnih vezij. Sposobnost minimizacije logičnih vezij. Razumevanje delovanja sekvenčnih vezij in avtomatov.

Uporaba:

Uporaba osnovnih orodij za načrtovanje vezij in izdelava enostavnih logičnih sklopov.

Refleksija:

Spoznavanje in razumevanje uglasenosti med teorijo in njeno aplikacijo na konkretnih primerih s enostavnih logičnih in sekvenčnih vezij.

Prenosljive spretnosti - niso vezane le na en

predmet:

Uporaba binarne logike.

Načrtovanje in izgradnja enostavnih digitalnih vezij.

Metode poučevanja in učenja:

Intended learning outcomes:

Knowledge and understanding of:

the process of writing, compiling, and running a computer program

basic programming constructs and data structures

object-oriented programming concepts

basic algorithms for typical program problems solving

event-driven programming and basic components of GUI

Application:

Application of concepts learned in development of simple computer programs.

Reflection:

Understanding of the role of a programmer in solving problems of different end-users.

Transferable skills:

The ability of using Slovenian and foreign literature and user manuals. Capability for self-reliant problem solving.

Learning and teaching methods:

<p>Predavanja z aktivno udeležbo študentov (razlaga, diskusija, vprašanja, primeri, reševanje problemov),</p> <p>Laboratorijske vaje (refleksija izkušenj, praktično reševanje več tipičnih problemov na računalniku, predstavitev in zagovor programskih rešitev, diskusija, sporočanje povratne informacije),</p> <p>Domače naloge (samostojna izdelava računalniških programov)</p> <p>Individualne konsultacije (diskusija, dodatna razlaga, obravnava specifičnih vprašanj)</p>	<p>Lectures with active participation of students (explanation, discussion, questions, examples, problem solving),</p> <p>Lab practice (reflection of experience, practical problem solving, presentation of solutions, discussion, communication of feedback information)</p> <p>Homework (individual development of simple computer programs)</p> <p>Individual consultation hours (discussion, additional explanation, specific problems solving)</p>
--	--

		Delež (v %) / Weight (in %)	Assessment:
Načini ocenjevanja:			
Način (pisni izpit, ustno izpraševanje, naloge, projekt):			Type (examination, oral, coursework, project):
Sprotno preverjanje (3 seminarske naloge, , kratki testi v obliki kvizov)			Continuing (homework, midterm exams, project work)
Končno preverjanje (izpit)	50%		Final (written and oral exam)
Ocene: 6-10 pozitivno, 1-5 negativno (v skladu s Statutom UL)	50%		Grading: 6-10 pass, 1-5 fail.

Reference nosilca / Lecturer's references:

MAHNIČ, Viljan. A capstone course on agile software development using Scrum. IEEE transactions on education, ISSN 0018-9359, Feb. 2012, vol. 55, no. 1, str. 99-106, tabele. [COBISS.SI-ID 8390740]

MAHNIČ, Viljan, HOVELJA, Tomaž. On using planning poker for estimating user stories. The Journal of Systems and Software, ISSN 0164-1212. [Print ed.], Sep. 2012, vol. 85, no. 9, str. 2086-2095, tabele. [COBISS.SI-ID 9127764]

MAHNIČ, Viljan. Teaching Scrum through team-project work : students' perceptions and teacher's observations. International journal of engineering education, ISSN 0949-149X, 2010, vol. 26, no. 1,

str. 96-110, tabele. [COBISS.SI-ID 7613012]

FÜRST, Luka, MAHNIČ, Viljan. Introductory programming course : motivating students with prior knowledge. World transactions on engineering and technology education, ISSN 1446-2257, 2013, vol. 11, no. 4, str. 400-405. [COBISS.SI-ID 10371668]

MAHNIČ, Viljan, FÜRST, Luka, ROŽANC, Igor. Java skozi primere. Šenčur: Bi-tim, 2008. XIV, 478 str., ilustr., tabele. ISBN 978-961-6046-10-7. [COBISS.SI-ID 242338304]