

PROGRAMIRANJE 2

UČNI NAČRT PREDMETA/COURSE SYLLABUS

Predmet:	PROGRAMIRANJE 2
Course title:	PROGRAMMING 2
Članica nosilka/UL	
Member:	

Študijski programi in stopnja	Študijska smer	Letnik	Semestri	Izbirnost
Upravna informatika, prva stopnja, univerzitetni	Upravna informatika (študijski program)	1. letnik	2. semester	obvezen

Univerzitetna koda predmeta/University course code:	0045526
Koda učne enote na članici/UL Member course code:	0863

Predavanja /Lectures	Seminar /Seminar	Vaje /Tutorials	Klinične vaje /Clinical tutorials	Druge oblike študija /Other forms of study	Samostojno delo /Individual student work	ECTS
45	0	30	0	0	105	6

Nosilec predmeta/Lecturer: Tomaž Dobravec

Vrsta predmeta/Course type: obvezni/core

Jeziki/Languages:

Predavanja/Lectures:	Slovenščina
Vaje/Tutorial:	Slovenščina

Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:

Opravljanje študijskih obveznosti je opredeljeno v Študijskih pravilih FRI.

Prerequisites:

As specified by internal acts of the University of Ljubljana and Faculty of Computer and Information Science.

Vsebina:

- Osnovno o programiranju
 - uporaba programskih jezikov za reševanje praktičnih problemov;
 - predstavitev dobrih praks in pasti programiranja;
 - predstavite uporabnih programerskih "receptov".
- Osnovni konstrukti programskih jezikov
 - podatkovni tipi, spremenljivke in konstante;
 - izrazi, operatorji, strukturiranje kode;
 - kazalci, dodeljevanje pomnilnika in nizi;
 - delo z datotekami.
- Osnove objektnega programiranja
 - spremenljivke, metode, člani, dostopnostna določila;
 - razredi, objekti, dedovanje, nadomeščanje;
 - abstraktni razredi, vmesniki;
 - dogodkovno vodeno programiranje;

Content (Syllabus outline):

- Basic programming skills
 - using programming languages to solve practical problems;
 - presentation of good practices and pitfalls of programming;
 - useful programming "recipes".
- The building blocks of programming languages
 - data types, variables and constants;
 - expressions, operators, structured code;
 - pointers, memory allocation and arrays;
 - work with files.
- Object-oriented programming
 - variables, methods, members, access modifiers;
 - classes, objects, inheritance, overriding;
 - abstract classes, interfaces;
 - event-driven programming;

• komentarji in dokumentacija.	• comments and documentation.
--------------------------------	-------------------------------

Temeljna literatura in viri/Readings:

<ul style="list-style-type: none"> • Brian W. Kernighan, Dennis M. Ritchie. The C Programming Language (2nd edition). Prentice Hall, 1988. • T. Dobravec. abC. Založba Fe&FRI, 2010. • K. Arnold, J. Gosling, D. Holmes. The Java Programming Language (3th edition). Addison-Wesley, 2000. • R. Morelli, R. Walde. Java, Java, Java Object-oriented problem solving (3th edition). Pearson Prentice Hall, 2006.
--

Cilji in kompetence:

<p>Cilj predmeta je študentom računalništva in informatike predstaviti osnovne principe imperativnega programiranja s poudarki na proceduralnem in objektno usmerjenem jeziku. Poleg pojmov, ki se nanašajo na konkretne programske konstrukte (funkcija, metoda, spremenljivka, operator, delo s pomnilnikom, kazalci, objekt, razred, dedovanje, ...) bodo študentje spoznali tudi pomen splošnih programerskih pojmov, kot so prenosljivost, prevajalnik, navidezni stroj, testiranje, razhroščevanje, ... Vsak študent, ki bo opravil ta predmet, naj bi bil sposoben samostojno reševati srednje zahtevne programerske probleme iz prakse.</p>	<h3>Objectives and competences:</h3> <p>The main goal of the course is to present the basic principles of imperative programming with emphasis on the procedural and object-oriented languages. In addition to concepts related to specific programming constructs (function, method, variable, operator, memory, pointers, object, class, inheritance, ...) students will learn the general programming concepts such as portability, compiler, virtual machine, testing, debugging, ... Each student that will conduct this course should be able to independently solve semi complex programming problems.</p>
---	---

Predvideni študijski rezultati:

<p>Znanje in razumevanje: Poznavanje osnovnih konstruktor programskih jezikov: podatkovni tip, spremenljivka, konstanta, izraz, operator, funkcija, kazalec, delo s pomnilnikom, nizi, datoteke. Poznavanje osnovnih pojmov objektnega programiranja: objekt, razred, član, metoda, dedovanje, enkapsulacija, nadomeščanje, prekrivanje, vmesniki. Razumevanje pomena splošnih programerskih pojmov, kot so: prenosljivost, prevajalnik, navidezni stroj, testiranje, razhroščevanje.</p> <p>Uporaba: Znanja, pridobljena pri tem predmetu, spadajo med osnovna računalniška znanja. Sposobnost samostojnega reševanja praktičnih problemov je večšina, ki je izredno uporabna ne samo pri velikem številu predmetov, s katerimi se študent sreča med študijem, pač pa tudi v praksi. Poznavanje sodobnih programskih jezikov omogoča sorazmerno lahek prehod na druge programske jezike.</p> <p>Refleksija: Poznavanje osnovnih programskih jezikov pomaga pri razumevanju delovanja računalnika ter pri mnogih predmetih, s katerimi se sreča študent računalništva in informatike. Programski jeziki so učinkovito orodje za reševanje industrijskih problemov ter pisanje poslovnih aplikacij. Prenosljive spretnosti - niso vezane le na en predmet:</p>	<h3>Intended learning outcomes:</h3> <p>Knowledge and understanding: Basic building blocks of programming languages: data type, variable, constant expression, operator, function, pointer, memory, arrays, files. and object-oriented programming (object, class, member, method, inheritance, encapsulation, replacement, overlay, interfaces). Understanding the importance of general programming concepts, such as portability, compiler, virtual machine, testing, debugging.</p> <p>Application: Students will gain the basic computer skills and the ability to independently solve practical problems, which is a skill that is extremely useful not only in large numbers of courses with which the student encounters during the study, but also in practice. A good knowledge of the most commonly used languages of modern industry allows relatively easy transition to other languages.</p> <p>Reflection: Knowledge of basic programming languages helps in understanding how the computer works. Modern programming languages are a powerful tool for solving industrial problems, and writing business applications.</p> <p>Transferable skills: Good knowledge of programming skills are very important for understanding how the computer system works.</p>
---	---

Dobro poznavanje programerskih spretnosti omogoča razumevanje delovanja celotnega računalniškega sistema.	
---	--

Metode poučevanja in učenja: Predavanja, laboratorijske vaje, delavnice.	Learning and teaching methods: Lectures, exercises and home work.
--	---

Načini ocenjevanja:	Delež/Weight	Assessment:
Način (pisni izpit, ustno izpraševanje, naloge, projekt): Sprotno preverjanje (domače naloge, kolokviji in projektno delo) Končno preverjanje (pisni in ustni izpit). Ocene: 6-10 pozitivno, 1-5 negativno (v skladu s Statutom UL).	50,00 %	Type (examination, oral, coursework, project): Continuing (homework, midterm exams, project work) Final: (written and oral exam). Grading: 6-10 pass, 1-5 fail.
Način (pisni izpit, ustno izpraševanje, naloge, projekt): Sprotno preverjanje (domače naloge, kolokviji in projektno delo) Končno preverjanje (pisni in ustni izpit). Ocene: 6-10 pozitivno, 1-5 negativno (v skladu s Statutom UL).	50,00 %	Type (examination, oral, coursework, project): Continuing (homework, midterm exams, project work) Final: (written and oral exam). Grading: 6-10 pass, 1-5 fail.

Reference nosilca/Lecturer's references:

<p>Pet najpomembnejših del: DOBRAVEC, Tomaž, ŽEROVNIK, Janez, ROBIČ, Borut. An optimal message routing algorithm for circulant networks. <i>J. systems archit.</i>. [Print ed.], 2006, vol. 52, no. 5, str. [298]-306 DOBRAVEC, Tomaž, ROBIČ, Borut. Restricted shortest paths in 2-circulant graphs. <i>Comput. commun.</i>. [Print ed.], March 2009, vol. 32, no. 4, str. 685-690 BULIĆ, Patricio, DOBRAVEC, Tomaž. Identifying data dependencies with a sufficiently large distance between memory references in a multimedia vectorizing compiler. <i>PDPTA 2008, Worldcomp'08, July 14-17, 2008, Las Vegas Nevada, USA.</i> str. 393-397 BULIĆ, Patricio, DOBRAVEC, Tomaž. An approximate method for filtering out data dependencies with a sufficiently large distance between memory references. <i>J. supercomput.</i>, 2009, doi:10.1007/s11227-009-0364-8. DOBRAVEC, Tomaž, BULIĆ, Patricio. Strojni in programski vidiki arhitekture CUDA. <i>Elektrotehniški vestnik</i>. [Slovenska tiskana izd.], 2010, letn. 77, št. 5, str. 267-272 Celotna bibliografija doc. dr. Dobravca je dostopna na SICRISu: http://sicris.izum.si/search/rsr.aspx?lang=slv&id=10416. Celotna bibliografija pred. dr. Slivnika je dostopna na SICRISu: http://sicris.izum.si/search/rsr.aspx?lang=slv&id=7849.</p>
--