UČNI NAČRT PREDMETA / COURSE SYLLABUS	
Predmet:	**Programski vzorci**
Course title:	Programming Paradigms

Študijski program in stopnja Study programme and level	Študijska smer Study field	Letnik Academic year	Semester Semester
Izobraževano računalništvo, 1. stopnja			
Educational computer science – Double-major, 1st degree			

Vrsta predmeta / Course type Izbirni predmet

Univerzitetna koda predmeta / University course code:

Predavanja Lectures	Seminar Seminar	Sem. vaje Tutorial	Lab. vaje Laboratory work	Teren. vaje Field work	Samost. delo Individ. work	ECTS
30			**30**		**120**	**6**

Nosilec predmeta / Lecturer: Janez Brest

Jeziki / Languages:		
	Predavanja / Lectures:	**slovenščina / Slovenian**
	Vaje / Tutorial:	**slovenščina / Slovenian**

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

Znanje, vključeno v predmete Programiranje I, Programiranje II in Diskretne strukture.

Prerequisits:

Knowledge included in the courses Programming 1, Programming 2, and Discrete Mathematics.

Vsebina:

- Uvod: neformalna definicija programskih jezikov, delitve in zgodovina programskih jezikov.
- Vrednosti in tipi: delitev tipov, preverjanje tipov, ekvivalenca tipov, vrste izrazov, vrednosti prvega in drugega razreda, princip polnosti tipa.
- Pomnilnik: spremenljivka, model pomnilnika, selektivno ali popolno ažuriranje, shranljive vrednosti, življenjska doba, vrste ukazov, izrazi s stranskimi učinki.
- Povezovanje: povezovanje, okolje in blok, povezljive vrednosti, vrste deklaracij, statični in dinamični doseg, bločni izrazi in bločni ukazi, kvalifikacijski princip.
- Abstrakcije: princip abstrakcije, funkcijske abstrakcije proceduralne abstrakcije, izbirne abstrakcije in generične abstrakcije, kopirni mehanizmi parametrov, korespondenčni princip, dosledni in normalni izračun.

Content (Syllabus outline):

- Introduction: definition of programming languages, classification and history of programming languages.
- Values and types: type classification, type checking, type equivalence, types of expressions, first-class values, second-class values, type completeness principle.
- Storage: variable, storage, total updating or selective updating, storable value, lifetime, types of commands, and expressions with side effects.
- Binding: binding, environment and block, bindable values, types of declarations, static scope, dynamic scope, block expression, qualification principle.
- Abstraction: abstraction principle, functional abstraction, procedural abstraction, selector abstraction, generic abstraction, copy parameter mechanism, correspondence principle, applicative-order evaluation, normal-order evaluation.

- Pojem “kapsuliranja”, koncepti, ki podpirajo modularnost.
- Sistemi tipov: monomorfni in polimorfni sistemi tipov, vrste polimorfizma, pojem “sekvencer”, vrste sekvencerjev.
- Osnovni koncepti objektno usmerjenega programiranja, sistemi prvega, drugega, tretjega in petega nivoja, hierarhija objektnih jezikov, vrste dedovanja.
- Funkcijsko programiranje, programski jezik Lisp/Haskell.

- Term “encapsulate”, concepts of modularity.
- Type systems: monomorphic and polymorphic type systems, types of polymorphism, term “sequencer”, types of sequencers.
- Basic concepts of object oriented programming, first, second, third, and fifth level systems, hierarchy of object languages, types of inheritance.
- Functional programming, programming language Lisp/Haskell.

Temeljni literatura in viri / Readings:

- V. Žumer, M. Mernik, *Principi programskih jezikov*, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor 2003.
- R. Sethi: *Programming Languages: Concepts and Constructions*, Second Edition, Addison-Wesley, Reading, 1996.
- D. A. Watt, *Programming Language Concepts and Paradigms*. Prentice-Hall International, New York, 1990.
- D. A. Watt, Programming Language Design Concepts, John Wiley, Chichester, 2004.

Cilji in kompetence:

Cilj predmeta je seznaniti študente s koncepti programskih jezikov in spoznati programske vzorce funkcijskega, logičnega in objektno usmerjeno programiranja.

Objectives and competences:

The objective of this course is to acquaint students with concepts of programming languages and programming paradigms of functional, logical, imperative and object oriented programming.

Predvideni študijski rezultati:

Znanje in razumevanje:

Po zaključku tega predmeta bo študent sposoben
- identificirati prednosti in slabosti posameznega vzorca,
- izbrati primeren programski jezik za rešitev dane naloge,
- razumeti koncepte programskih jezikov s pomočjo katerih se bodo hitreje naučili novega programskega jezika,
- razumeti razlike med statičnim in dinamičnim tipiziranjem,
- razumeti različne oblike dodeljevanja pomnilnika, dosega,
- razumeti različne oblike prenosa parametrov,
- razumeti pomen abstrakcij,
- razumeti različne oblike polimorfizma.

Prenosljive/ključne spretnosti in drugi atributi:

- *Spretnosti komuniciranja:* ustni zagovor laboratorijskih vaj, pisno izražanje pri pisnem izpitu.
- *Uporaba informacijske tehnologije:* uporaba objektno usmerjenih, funkcijskih in logičnih programskih jezikov.
- *Reševanje nalog:* uporaba programskih vzorcev pri načrtovanju in implementaciji programov.

Intended learning outcomes:

Knowledge and understanding:

On completion of this course the student will be able to
- identify advantages and disadvantages of some paradigms,
- to choose suitable program language for solving defined problem,
- understand concepts of programming languages for quickly learning a new programming language,
- understand differences between static and dynamic typing
- understand different storage and scope model,
- understand parameter mechanisms,
- understand term abstraction,
- understand different types of polymorphism.

Transferable/Key skills and other attributes:

- *Communication skills:* oral lab work defence, manner of expression at written examination.
- *Use of information technology:* use object oriented, functional and logic programming languages.
- *Problem solving:* using programming paradigms for program design and implementation.

Metode poučevanja in učenja:

Learning and teaching methods:

• predavanja, • laboratorijske vaje, • reševanje domačih nalog.	• lectures, • lab work, • homework assignments.

Načini ocenjevanja:	Delež (v %) / Weight (in %)	**Assessment:**
• domače naloge, • kvizi, • laboratorijske vaje, • pisni izpit.	10 % 30 % 30 % 30 %	• homeworks, • quizzes, • lab work, • written examination.

Reference nosilca / Lecturer's references:

1.01 Izvirni znanstveni članek

1. FISTER, Iztok, FISTER, Iztok, MERNIK, Marjan, BREST, Janez. Design and implementation of domain-specific language easytime. *Comput. syst. struct.*, Oct. 2011, vol. 37, iss. 4, str. 151-167, doi: 10.1016/j.cl.2011.04.001. [COBISS.SI-ID 14934550], [JCR, WoS do 1. 5. 2013: št. citatov (TC): 2, čistih citatov (CI): 1, normirano št. čistih citatov (NC): 1, Scopus do 18. 9. 2013: št. citatov (TC): 8, čistih citatov (CI): 4, normirano št. čistih citatov (NC): 4]

2. GREINER, Sašo, BREST, Janez, ŽUMER, Viljem. Zero - A blend of static typing and dynamic metaprogramming. *Comput. syst. struct.*, Oct. 2009, vol. 35, no. 3, str. 241-251, doi: doi:10.1016/j.cl.2008.04.001. [COBISS.SI-ID 12284694