NOVOtrenutno

Univerza v Mariboru

Fakulteta za naravoslovje in matematiko

UČNI NAČRT PREDMETA / COURSE SYLLABUS	
Predmet:	**Programiranje II**
Course title:	**Programming II**

Študijski program in stopnja Study programme and level	Študijska smer Study field	Letnik Academic year	Semester Semester
Enovit magistrski študijski program druge stopnje Predmetni učitelj	**/**	**1.**	**2**
Five-year master's degree program Subject Teacher	**/**		

Vrsta predmeta / Course type	Obvezni/ Obligatory
Univerzitetna koda predmeta / University course code:	

Predavanja Lectures	Seminar Seminar	Vaje Tutorial	Lab. vaje Laboratory work	Terenske vaje Field work	Samost. delo Individ. work	ECTS
30		**3**	**42**		**135**	**7**

Nosilec predmeta / Lecturer:	Mernik Marjan

Jeziki / Languages:	**Predavanja / Lectures:**	slovenski / slovene
	Vaje / Tutorial:	slovenski / slovene

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

Vsebina:	**Content (Syllabus outline):**
• Uvod v objektno usmerjeno programiranje: slabosti strukturiranega programiranja, prednosti objektno usmerjenega programiranja. • Razredi in objekti: razred in komponente razreda, ustvarjanje objektov, konstruktorji in destruktorji, konstantni objekti, statični elementi razreda, skrivanje informacij. • Kazalci: kazalci na objekte, kazalec this,	• Introduction to object-oriented programming: shortcomings of structured programming, advantages of object-oriented programming. • Classes and objects: class and class members, object creation, constructors and destructors, constant objects, static class members, information hiding. • Pointers: pointer to objects, this pointer,

dinamične podatkovne strukture. • Prekrivanje: prekrivanje metod, prekrivanje operatorjev, operatorji kot prijateljske funkcije. • Dedovanje: nadrazredi in izpeljani razredi, zaščiteni elementi, virtualne funkcije, abstraktni razredi, dinamično povezovanje, večkratno dedovanje. • Uvod v objektno usmerjeno analizo in načrtovanje: analiza, načrtovanje, osnove UML. • Šablone: šablone funkcij, šablone razredov, knjižnica STL. • Izjeme: zbujanje izjem, lovljenje izjem. • Podatkovni tokovi: vhodni tokovi, izhodni tokovi, datoteke, vhod/izhod za uporabniško definirane tipe. • Implementacija objektov v jeziku C++.	dynamic data structures. • Overloading: method overloading, operator overloading, operators as friend functions. • Inheritance: super classes, derived classes, protected members, virtual functions, abstract classes, dynamic binding, multiple inheritance. • Introduction to object-oriented analysis and design: analysis, design, UML fundamentals. • Templates: function templates, class templates, Standard Template Library (STL). • Exceptions: throwing exceptions, exception handling. • Data streams: input streams, output streams, files, I/O streams for user defined types. • Implementation of objects in C++ programming language.

Temeljni literatura in viri / Readings:

- B. Stroustrup: *Programming: Principles and Practice Using C++*, Addison-Wesley, Reading, 2014.
- B. Stroustrup: The C++ Programming Language, Second Edition, Addison-Wesley, Reading, 1991.
- H. M. Deitel, P. J. Deitel: *C++ How to Program*, Fourth Edition, Prentice Hall, London, 2002.
- F. M. Carrano: *Data Abstraction and Problem Solving with C++: Walls and Mirrors*, Fourth Edition, Addison Wesley, Reading, 2004.
- F. L. Friedman, E. B. Koffman: Problem Solving, Abstraction, and Design Using C++, Addison-Wesley, Reading, 1994.

Cilji in kompetence:	**Objectives and competences:**
Cilj tega predmeta je seznaniti študente z osnovami objektno usmerjenega programiranja in jih naučiti načrtovanja, kodiranja, razhroščevanja, testiranja in dokumentiranja programov.	The objective of this course is to acquaint students with the fundamentals of object-oriented programming and to teach them how to design, implement, debug, test, and document programs.

Predvideni študijski rezultati:	**Intended learning outcomes:**
Znanje in razumevanje: Po zaključku tega predmeta bo študent sposoben • razumeti tehnike objektno usmerjenega programiranja, • razumeti razlike med strukturiranim in objektno usmerjenim programiranjem,	**Knowledge and understanding:** On completion of this course the student will be able to • understand object-oriented programming techniques, • understand differences between structured and object-oriented programming,

• definirati razrede in podrazrede, ustvarjati objekte, definirati polimorfne metode, • razumeti kako so objekti implementirani v jeziku C++, • uporabljati podatkovne tokove, • definirati šablone in obravnavati izjeme, • uporabljati knjižnico STL, • načrtovanja, implementiranja, razhroščevanja in dokumentiranja preprostejših programov. **Prenosljive/ključne spretnosti in drugi atributi:** • *Spretnosti komuniciranja:* ustni zagovor laboratorijskih vaj, pisno izražanje pri pisnem izpitu. • *Uporaba informacijske tehnologije:* uporaba programskih orodij za načrtovanje, implementacijo, razhroščevanje in testiranje programov. • *Reševanje problemov:* načrtovanje in implementacija programov.	• define classes and subclasses, create objects and define polymorphic methods, • understand how objects are implemented in C++ language, • use data streams, • define templates and exceptions, • use Standard Template Library, • design, implement, debug and document simple programs. **Transferable/Key Skills and other attributes:** • *Communication skills:* oral lab work defence, manner of expression at written examination. • *Use of information technology:* use of different tools for program design, implementation, debugging and testing. • *Problem solving:* design and implementation of programs.

Metode poučevanja in učenja:	**Learning and teaching methods:**
• predavanja, • laboratorijske vaje.	• lectures, • lab work.

Načini ocenjevanja:	Delež (v %) / Weight (in %)	**Assessment:**
• laboratorijske vaje, • 1. vmesni pisni izpit, • 2. vmesni pisni izpit, • 3. vmesni pisni izpit. **Klasični način** • laboratorijske vaje, • pisni izpit.	50 % 16 % 17 % 17 % **Delež (Pogoj)** 50 (25) 50 (25)	• lab work, • 1st midterm written exam, • 2nd midterm written exam, • 3rd midterm written exam. **Classical assessment methods** • lab work, • written examination.

~~Opomba: Če študent ni uspešno opravil vseh treh vmesnih izpitov, jih nadomesti s pisnim izpitom v deležu 50%.~~

~~Note: If a student has not completed all three midterm exams, he replaces them with a written exam in the weight of 50%.~~

Reference nosilca / Lecturer's references:

- KOSAR, Tomaž, MERNIK, Marjan, GRAY, Jeffrey G., KOS, Tomaž. Debugging measurement systems using a domain-specific modeling language. *Computers in industry*, ISSN 0166-3615. [Print ed.], 2014, vol. 65, iss. 4, str. 622-635.
- MERNIK, Marjan. An object-oriented approach to language compositions for software language engineering. The Journal of Systems and Software, ISSN 0164-1212. [Print ed.], 2013, vol. 86, iss. 9, str. 2451-2464.
- KOSAR, Tomaž, MERNIK, Marjan, CARVER, Jeffrey C. Program comprehension of domain-specific and general-purpose languages : comparison using a family of experiments. Empirical software engineering, ISSN 1382-3256, 2012, vol. 17, no. 3, str. 276-304.
- ARORA, Ritu, BANGALORE, Purushotham, MERNIK, Marjan. Raising the level of abstraction for developing message passing applications. The journal of supercomputing, ISSN 0920-8542, 2012, vol. 59, no. 2, str. 1079-1100.
- KOS, Tomaž, KOSAR, Tomaž, MERNIK, Marjan. Development of data acquisition systems by using a domain-specific modeling language. Computers in industry, ISSN 0166-3615.[Print ed.], Apr. 2012, vol. 63, no. 3, str. 181-192.