



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

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:

Jeziki / Predavanja / Lectures:
Languages: Vaje / Tutorial:

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

Vsebina:

- Uvod v objektno usmerjeno programiranje: slabosti strukturiranega programiranja, prednosti objektno usmerjenega programiranja.
- Razredi in objekti: razred in komponente razreda, ustvarjanje objektov, posebne metode, delo z viri, konstantni objekti, statični elementi razreda, skrivanje informacij.

Content (Syllabus outline):

- Introduction to object-oriented programming: shortcomings of structured programming, advantages of object-oriented programming.
- Classes and objects: class and class members, object creation, special member functions, resource management, constant objects, static class members, information hiding.

- Kazalci: kazalci na objekte, kazalec this, 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.
- Novi konstrukti v jeziku C++11 in C++14.
-

- Pointers: pointer to objects, this pointer, 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.
- New language features of C++11 and C++14.

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:

Cilj tega predmeta je razumeti osnove objektno usmerjenega programiranja in jih naučiti načrtovanja, kodiranja, razhroščevanja, testiranja in dokumentiranja programov.

Objectives and competences:

The objective of this course is to understand the fundamentals of object-oriented programming and to teach them how to design, implement, debug, test, and document programs.

Predvideni študijski rezultati:

Znanje in razumevanje:

Po zaključku tega predmeta bo študent sposoben

- razumeti tehnike objektno usmerjenega programiranja,
- razumeti razlike med strukturiranim in objektno usmerjenim programiranjem,

Intended learning outcomes:

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,

<ul style="list-style-type: none"> • definirati razrede in podrazrede, ustvarjati objekte, definirati polimorfne metode, • razumeti, kako so objekti implementirani v jeziku C++, • uporabljati knjižnico STL, • načrtovanja, implementiranja, razhroščevanja in dokumentiranja preprostejših programov. <p>Prenosljive/ključne spretnosti in drugi atributi:</p> <ul style="list-style-type: none"> • <i>Spretnosti komuniciranja:</i> ustni zagovor laboratorijskih vaj, pisno izražanje pri pisnem izpitu. • <i>Uporaba informacijske tehnologije:</i> uporaba programskih orodij za načrtovanje, implementacijo, razhroščevanje in testiranje programov. • <i>Reševanje problemov:</i> načrtovanje in implementacija programov.

<ul style="list-style-type: none"> • define classes and subclasses, create objects and define polymorphic methods, • understand how objects are implemented in C++ language, • use Standard Template Library, • design, implement, debug and document simple programs. <p>Transferable/Key Skills and other attributes:</p> <ul style="list-style-type: none"> • <i>Communication skills:</i> oral lab work defence, manner of expression at written examination. • <i>Use of information technology:</i> use of different tools for program design, implementation, debugging and testing. • <i>Problem solving:</i> design and implementation of programs.
--

Metode poučevanja in učenja:

Learning and teaching methods:

<ul style="list-style-type: none"> • predavanja, • laboratorijske vaje.

<ul style="list-style-type: none"> • lectures, • lab work.
--

Načini ocenjevanja:	Delež (v %) / Weight (in %)	Assessment:
<ul style="list-style-type: none"> • laboratorijske vaje • 1. kolokvij, • 2. kolokvij. 	50 % 25 % 25 %	<ul style="list-style-type: none"> • lab work • 1st midterm written exam, • 2nd midterm written exam. – •

Opomba:

Kolokvija se lahko nadomestita s pisnim izpitom v deležu 50 %.

Note:

The midterm examinations may be replaced by written exam in the weight of 50%.

Reference nosilca / Lecturer's references:

- KOS, Tomaž, MERNIK, Marjan, KOSAR, Tomaž. A tool support for model-driven development : an industrial case study from a measurement domain. Applied sciences, ISSN 2076-3417, 2019, vol. 9, iss. 21, str. 1-20, ilustr., doi: 10.3390/app9214553. [COBISS.SI-ID 22712342]
- KOSAR, Tomaž, GABERC, Sašo, CARVER, Jeffrey C., MERNIK, Marjan. Program comprehension of domain-specific and general-purpose languages: replication of a family of experiments using integrated development environments. Empirical software engineering, ISSN 1382-3256, 2018, vol. 23, iss. 5, str. 2734-2763, doi: 10.1007/s10664-017-9593-2. [COBISS.SI-ID 21123606]
- KOSAR, Tomaž, BOHRA, Sudev, MERNIK, Marjan. A systematic mapping study driven by the margin of error. The Journal of Systems and Software, ISSN 0164-1212. [Print ed.], Oct. 2018, vol. 144, str. 439-449, doi: 10.1016/j.jss.2018.06.078. [COBISS.SI-ID 21580566]