SYLLABUS / FIȘA DISCIPLINEI

1. Information on the study programme / Date despre programul de studii

1.1. Institution / Instituția de învățământ superior	Universitatea de Vest din Timişoara
1.2. Faculty / Facultatea	Matematică și Informatică
1.3. Department / Departamentul	Computer Science (Informatică)
1.4. Study program field	Computer Science (Informatică)
1.5. Study cycle/ Ciclul de studii	Bachelor / licență
1.6. Study programme / Programul de studii / calificarea*	Computer Science / Informatică în limba engleză / Database administration / *Administrator baze de date - 252101; Computer network administration / Administrator de retea de calculatoare - 252301; Analyst / Analist - 251201; Research assistant in computer science / Asistent de cercetare în informatica - 214918; Teacher in secondary schools / Profesor în învatamântul gimnazial - 233002; Programmer / Programator - 251202; Software systems designers / Proiectant sisteme informatice - 251101*

2. Information on the course / Date despre disciplină

2.1. Title of the course / Denumirea disciplinei	Programming II						
2.2. Lecture instructor / Titularul activităților de curs	Daniel Pop						
2.3. Seminar / laboratory instructor / Titularul activităților de seminar	Isabela Dramnesc, Daniel Pop						
2.4. Study year / Anul de studii	1	2.5. Semester / Semestrul	2	2.6. Examination type / Tipul de evaluare: E(xam)/C(olloquim)	E	2.7. Course type / Regimul disciplinei: M(andatory)/ E(lective)/ F(acultative)	DI

3. Estimated study time (number of hours per semester) /Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Attendance hours per week /	4	out of which din	2	3.3. seminar/laborator	2

Număr de ore pe săptămână		care: 3.2 lecture/ curs			
3.4. Attendance hours per semester / Total ore din planul de învățământ	56	out of which: 3.5 lecture / curs	28	3.6. seminar/laborator	28
Distribution of the allocated amount of time / Distribuția fondului de timp*					**hours/ ore**
Individual study /Studiu după manual, suport de curs, bibliografie și notițe					35
Supplementary documentation at library or using electronic repositories / Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate					15
Preparing for laboratories, homework, reports etc. /Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					40
Exams / Examinări					6
Tutoring / Tutorat					8

3.7. Total number of hours of individual study / Total ore studiu individual	104
3.8. Total number of hours per semester / Total ore pe semestru	160
3.9. Number of credits (ECTS) / Număr de credite	6

4. Prerequisites (if it is the case) / Precondiții (acolo unde e cazul)

4.1. curriculum / de curriculum	Programming I, Algorithms and Data structures I
4.2. Competences / de competențe	Problem solving abilities, Proficiency in English

5. Requirements (if it is the case) / Condiții (acolo unde e cazul)

5.1. for the lecture / de desfășurare a cursului	Room equiped with beamer and whiteboard
5.2. for the seminar, laboratory / de desfășurare a seminarului/laboratorului	Room equiped with computers running one of the following IDEs: MS Visual Studio (Academic licence), Code Blocks, Eclipse (with C++ plugin), IntelliJ IDEA (with C++ plugin)

6. Acquired skills / Competențe specifice acumulate

Professional skills / Competențe profesionale	• Good knowledge of object-oriented concepts • Learn about principles of object oriented design • Good knowledge of generic programming • Ability to implement small-sized object-oriented projects in C++ • Usage of Code Blocks / Microsoft Visual Studio IDEs
Transversal skills / Competențe transversale	• Ability to build complex systems based on elementary building blocks • Develop an analytical spirit and curisosity about how computer software works • Conceptual modelling, i.e. ability to represent real-life problems using abstract models

7. Objectives of the course / Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1. General objective / Obiectivul general al disciplinei	Assimilate and ability to operate with object-oriented paradigm; develop an object-oriented mindset
7.2. Specfic objectives / Obiectivele specifice	*Knowledge wise objectives (KO):* (1) Good knwoledge of object-oriented paradigm (2) Good knowledge of generic programming concepts (3) Basic conceptual modelling and object-oriented design; (4) Represent OOP and generic programming concepts in C++ programming language *Ability wise objectives (AO):* (1) Ability to design simple problems using OOP concepts; (2) Ability to implement and test simple problems in C++ programming language; (3) Ability to debug running programms *Skills wise objectives (SO):* (1) Root cause analysis; (2) Abstractization and conceptualization of real-life problems

8. Content / Conținuturi*

8.1. Lecture / Curs	**Teaching strategies / Metode de predare**	**Remarks, details / Observații**

C1. (2h) Introduction. Evolution of programming paradigms. Short history of OOP and C++	Lecture, discussions, active student participation	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 3rd Edition, Addison Wesley, 1997 [Section 4.9.4]
C2. (2h) Object oriented programming concepts. Object. Class. Message.	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 4th Edition, Addison Wesley, 2013 [Chapter 16]
C3. (2h) Classes. Access control. Constructor. Destructor. Self-reference. Modifiers	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 4th Edition, Addison Wesley, 2013 [Chapter 16]
C4. (2h) Classes. Class declaration and definition. Object instantiation. Class members. Scopes	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 4th Edition, Addison Wesley, 2013 [Chapter 16]
C5. (2h) Objects. Class relationships. Object construction and destruction.	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 4th Edition, Addison Wesley, 2013 [Chapter 16]
C6. (2h) Association. Composition. Aggregation	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++

		Programming Language 4th Edition, Addison Wesley, 2013 [Chapter 16]
C7 (2h) Inheritance. Derived classes. Accessing base class members. Constructors. Destructor. Virtual functions.	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 4th Edition, Addison Wesley, 2013 [Chapter 20]
C8. (2h) Inheritance. Abstract classes. Polymorphism. Class hierarchies. Multiple inheritance	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 4th Edition, Addison Wesley, 2013 [Chapter 16]
C9 (2h) Exception handling. Definition. Throw-try-catch mechanism. Exception	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 4th Edition, Addison Wesley, 2013 [Chapter 13]
C10. (2h) Generic programming. Introduction. Template classes. Template functions. Specializations. Inheritance	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 4th Edition, Addison Wesley, 2013 [Chapter 23]
C11. (2h) C++ Standard Library. Introduction. Containers. Algorithms.	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 4th Edition, Addison Wesley, 2013 [Chapter 30]

C12. (2h) C++ Standard Library. Introduction. Algorithms. Iterators. Strings. Streams. Numerics	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 4th Edition, Addison Wesley, 2013 [Chapter 30]
C13. (2h) Object oriented analysis. Overview. Activities. Use cases. Case study.	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 4th Edition, Addison Wesley, 2013
C14. (2h) Object oriented design: Overview. OOD principles. Case study	Idem	Lecture notes http://web.info.uvt.ro/~danielpop Bjarne Stroustrup – The C++ Programming Language 4th Edition, Addison Wesley, 2013
Recommended bibliography / Bibliografie [1] Bjarne Stroustrup – The C++ Programming Language 4th Edition. Addison Wesley, 2013. [2] Brett McLaughlin, Gary Pollice, David West – Head First Object-Oriented Analysis and Design, O'Reilly, 2006		
8.2. Seminar, lab / Seminar, laborator	**Teaching/learning strategies / Metode de predare/ invățare**	
L1 (2h) Recap on C programming language	Hands-on, discussions, homework	http://web.info.uvt.ro/~idramnesc/ProgramareC++_en.html
L2. (2h) Improvements in C++	Idem	Idem
L3. (2h) Basic class definition	Idem	Idem
L4. (2h) Modifiers: static, const, friend	Idem	Idem

L5. (2h) Object instantiation	Idem	Idem
L6. (2h) Class relationships	Idem	Idem
L7. (2h) Operator overloading	Idem	Idem
L8.(2h) Inheritance and polymorphism (I)	Idem	Idem
L9.(2h) Inheritance and polymorphism (II)	Idem	Idem
L10.(2h) Exception handling	Idem	Idem
L11.(2h) Template classes	Idem	Idem
L12. (2h) Template functions	Idem	Idem
L13. (2h) Standard Template Library (STL)	Idem	Idem
L14. (2h) Final test		

Recommended bibliography / Bibliografie

[1] Bjarne Stroustrup – The C++ Programming Language 4th Edition. Addison Wesley, 2013

9. Correlations between the content of the course and the requirements of the IT field / Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului

It is nearly impossible to develop applications nowdays that are not written in an object-oriented language. Object-oriented modelling and implementation is the de-facto approach used to implement complex systems across multiple businesses, such as financial, commercial, industrial or online commerce. The local, national and internationall workforce market is continuously looking for highly-skilled personnel to develop complex systems using object-oriented languages and environments.

10. Evaluation / Evaluare*

Activity / Tip de activitate	10.1. Evaluation criteria / Criterii de evaluare**	10.2. Evaluation methods / Metode de evaluare***	10.3. Weight in the averaged mark / Pondere din nota finală
10.4. Lecture / Curs	• (KO1) Good knwoledge of object-oriented paradigm • (KO2) Good knowledge of generic programming concepts • (KO3) Basic conceptual modelling and object-oriented design; • (KO4) Represent OOP and generic programming concepts in C++ programming language • (AO1) Ability to design simple problems using OOP concepts;	Written test at exam	50%
10.5. Seminar/ lab	• (AO1) Ability to design simple problems using OOP concepts; • (AO2) Ability to implement and test simple problems in C++ programming language; • (AO3) Ability to debug running programms • (SO2) Abstractization and conceptualization of real-life problems	Practical test (last laboratory)	50%
10.6. Minimal knowledge for passing / Standard minim de performanță			
Minmal knowledge for passing this subject: • Good knowledge of basic concepts of object-oriented paradigm • Ability to model a simple problem using object-oriented concepts (classes, relationships) • Ability to understand and model simple problems using template classes and functions • Ability to write a simple programm (2-3 relatd classes) in C++ programming language, to compile it and run it successfully			

- Ability to use basic collections and algorithms from STL (Standard Template Library)

The final grade is computed as a weighted average of grades obtained for components described in 10.4 and 10.5. The exam is passed if each individual grade obtained at components 10.4 and 10.5 (i.e. both lecture and lab evaluations) are greater or equal to 5. This rule is inforce for all exam periods. The student need to re-take only the failed component (course or lab, respectively), unless the student wishes to re-take both evaluations.

Final remark: All students all welcome to tutoring meetings as scheduled by the department.

Date/ Data completării	Signature (lecture) / Semnătura titularului de curs	Signature (seminar) Semnătura titularului de seminar
1.10.2016		

Signature (director of the department)
Semnătura directorului de departament
Conf.dr. Victoria Iordan