

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 Timișoara
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 / <i>Administrator baze de date - 252101; Computer network administration / Administrator de rețea de calculatoare - 252301; Analyst / Analist - 251201; Research assistant in computer science / Asistent de cercetare în informatică - 214918; Teacher in secondary schools / Profesor în învățământul gimnazial - 233002; Programmer / Programator - 251202; Software systems designers / Proiectant sisteme informatice - 251101</i>

2. Information on the course / Date despre disciplină

2.1. Title of the course / Denumirea disciplinei		Design Patterns					
2.2. Teacher in charge of the course / Titularul activităților de curs		Flavia Micota					
2.3. Teacher in charge of the seminar / Titularul activităților de seminar		Flavia Micota					
2.4. Study year / Anul de studii	3	2.5. Semester / Semestrul	1	2.6. Examination type / Tipul de evaluare: E(xam)/C(olloquim)	C	2.7. Course type / Regimul disciplinei: M(andatory)/ E(lective)/ F(acultative)	DO

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

3.1. Attendance hours per week / Număr de ore pe săptămână	3	out of which din care: 3.2 lecture/ curs	2	3.3. seminar/laborator	1
3.4. Attendance hours per semester / Total ore din planul de învățământ	42	out of which: 3.5 lecture / curs	28	3.6. seminar/laborator	14
Distribution of the allocated amount of time / Distribuția fondului de timp*					hours/ ore
Individual study /Studiu după manual, suport de curs, bibliografie și notițe					20
Supplementary documentation at library or using electronic repositories / Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate					30
Preparing for laboratories, homework, reports etc. /Pregătire seminarii/laboratoare, teme,					40

referate, portofolii și eseuri	
Exams / Examinări	3
Tutoring / Tutorat	10
3.7. Total number of hours of individual study / Total ore studiu individual	90
3.8. Total number of hours per semester / Total ore pe semestru	145
3.9. Number of credits (ECTS) / Număr de credite	4

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

4.1. curriculum / de curriculum	Object Oriented Programming skills, Java language knowledge, Unified Modeling Language knowledge
4.2. skills / de competențe	Ability to analyze a software project code

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

5.1. for the lecture / de desfășurare a cursului	Course room with blackboard and projector
5.2. for the seminar, laboratory / de desfășurare a seminarului/laboratorului	Laboratory room with necessary facilities (computers with Java language installed)

6. Acquired skills / Competențe specifice acumulate

Professional skills / Competențe profesionale	<ul style="list-style-type: none"> • Knowledge of analysis and design methodologies for software projects • Usage of the knowledge to design applications • Ability to analyze an activity domain and design computer applications for it • Development of a project within a defined framework
Transversal skills / Competențe transversale	<p><i>Knowledge objective (KO):</i> (1) Execution of some specific tasks in the activity of code implementation and document design; (2) Identification and applying patterns in software projects</p> <p><i>Ability objectives (AO):</i> (1) can read and implement UML diagram of a pattern in a programming language (2) identify possible places in software projects where patterns should be applied</p> <p><i>Behaviour objectives (BO):</i> (1) to argue the importance of using patterns in process of software development; (2) to argue the proposed solution</p>

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

7.1. General objective / Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Designing an application using certain design patterns • Identification in existing applications of the possibility to use design patterns • Acquiring knowledge in order to use design patterns in object-oriented application development
7.2. Specific objectives / Obiectivele specifice	<ul style="list-style-type: none"> • Changing the existing of a project • Identify where you can apply templates in projects • Communication and collaboration within the project team • The use of specific software development tools

8. Content / Conținuturi*

8.1. Lecture / Curs	Teaching strategies / Metode de predare	Remarks, details / Observații
1. Design Patterns. Introductions (KO, AO)	Lecture, conversation, illustration	References: F. Micota - course - web.info.uvt.ro/zflavia
2. Fundamental patterns (KO, AO)	Lecture, conversation, illustration	Idem
3. Creational patterns (KO, AO)	Lecture, conversation, illustration	Idem
4. Structural patterns (KO, AO)	Lecture, conversation, illustration	Idem
5. Structural- behavioral patterns (KO, AO)	Lecture, conversation, illustration	Idem
6. Behavioral patterns (KO, AO)	Lecture, conversation, illustration	Idem
7. Behavioral. Partitioning patterns (KO, AO)	Lecture, conversation, illustration	Idem
8. GUI patterns (KO, AO2)	Lecture, conversation, illustration	Idem
9. Refactoring (KO, AO2)	Lecture, conversation, illustration	Idem
10. Antipatterns (KO, AO2)	Lecture, conversation, illustration	Idem
11. Tiered Applications. J2EE – design patterns I (KO)	Lecture, conversation, illustration	Idem
12. Tiered Applications. J2EE – design patterns II (KO)	Lecture, conversation, illustration	Idem
13. A study case (KO, AO, BO)	Conversation, illustration	Idem

14. Exam	Examination	Idem
Recommended bibliography / Bibliografie 1. Design patterns: elements of reusable object-oriented software, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides (1994) 2. Head First Design Patterns, Eric Freeman, Bert Bates, Kathy Sierra, Elisabeth Robson (2004) 3. Refactoring: Improving the design of existing code, Fowler Martin (1999). 4. Refactoring to Patterns. Kerievsky Joshua (2004). 5. Core J2EE Patterns: Best Practices and Design Strategies, Deepak Alur (2001)		
8.2. Seminar, lab / Seminar, laborator	Teaching/learning strategies / Metode de predare/ învățare	Remarks, details / Observații
1. Present working example. Modify existing example to respect fundamental patterns. Identify weak points (AO1, BO)	Questioning, dialogue, collaborative learning	
2. Add creational patterns at working example (AO1, BO)	Questioning, dialogue, collaborative learning	
3. Add structural patterns at working example (AO1, BO)	Questioning, dialogue, collaborative learning	
4. Add behavior patterns to working example (AO1, BO)	Questioning, dialogue, collaborative learning	
5. GUI creation for the proposed example (AO1, BO)	Questioning, dialogue, collaborative learning	
6. Finishing the project (AO1, BO)	Questioning, dialogue, collaborative learning	
7. Project presentation	Examination	
Recommended bibliography / Bibliografie 1. web.info.uvt.ro/~zflavia		

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

The content of this course is consistent with structure similar courses from other universities and covers the fundamental aspects necessary familiarity with issues of design and identification of patterns in software projects. The ability to identify, design, implement and analyze software projects and software projects requirements is essential for any activity in the field of informatics. Skills offered by this discipline for an IT specialist is needed to identify effective solutions for solving concrete problems.

10. Evaluation / Evaluare*

Activity / Tip de	10.1. Evaluation criteria / Criterii de	10.2. Evaluation	10.3. Weight in
-------------------	---	------------------	-----------------

activitate	evaluare**	methods / Metode de evaluare***	the averaged mark / Pondere din nota finală
10.4. Lecture / Curs	The ability to describe existing design patterns and refactor applications in order to respect a pattern	Written exam	50%
	The ability to identify design patterns in given examples	Course activity	1%
10.5. Seminar/ lab	The implementation of design patterns	Laboratory activity	24%
	A presentation and an implementation example for a design pattern	Pattern presentation	10%
	The laboratory requirements full implementation	Project Presentation	15%
10.6. Minimal knowledge for passing / Standard minim de performanță			
Minimal knowledge for passing (knowledge for 5 mark) <ul style="list-style-type: none"> ● description of a creational pattern ● knowledge acquisition about the cases where the presented patterns are applicable ● identification of possible patterns application in a given example The mark is calculated like ponder mean of the marks from section 10. At each session previous obtain marks are taken into account.			

Date/ Data completării

9.10.2016

Signature (lecture) /

Semnătura titularului de curs
Flavia Micota

Signature (seminar)

Semnătura titularului de seminar
Flavia Micota

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