



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

2. Information on the course / Date despre disciplină

2.1. Title of the course / Denumirea disciplinei	Programming III				
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, Monica Tirea				
2.4. Study year / Anul de studii	2	2.5. Semester / Semestrul	1	2.6. Examination type / Tipul de evaluare: E(xam)/C(olloquim)	2.7. Course type / Regimul disciplinei: M(andatory)/ E(lective)/ F(acultative)

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	4	out of which din care: 3.2 lecture/ curs	2	3.3. seminar/laborator	2
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					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	10
Tutoring / Tutorat	20
3.7. Total number of hours of individual study / Total ore studiu individual	50
3.8. Total number of hours per semester / Total ore pe semestru	176
3.9. Number of credits (ECTS) / Număr de credite	5

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

4.1. curriculum / de curriculum	C Language Programming, Oriented Object Programming
4.2. skills / de competențe	Basic knowledge of computer operation

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

5.1. for the lecture / de desfășurare a cursului	Room with blackboard and projector
5.2. for the seminar, laboratory / de desfășurare a seminarului/laboratorului	Room with computers that have installed Java programming language and Eclipse IDE

6. Acquired skills / Competențe specifice acumulate

Professional skills / Competențe profesionale	<ul style="list-style-type: none"> • Consolidate the learned concepts of object oriented programming • Usage of Eclipse IDE • Creation of beginner level programs using object oriented concepts
Transversal skills / Competențe transversale	<ul style="list-style-type: none"> • The ability to split a problem in sub-problems • The ability to build complex structures starting from elementary things • The improvement of computer usage

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

7.1. General objective / Obiectivul general al disciplinei	The familiarization with Java language, usage of some Java libraries like Swing, IO. The ability to develop object oriented programs.
7.2. Specific objectives / Obiectivele specifice	<p><i>Knowledge objectives (KO):</i>(1) Java language familiarization; (2) Java libraries familiarization; (3) oriented object concepts study</p> <p><i>Ability objectives (AO):</i>(1) to identify the usage of proper libraries based on the context of the problem; (2) to design object oriented application</p> <p><i>Attitudinal objectives (AtO):</i>(1) to argue the proposed solution for a problem by identifying advantages/disadvantages of the proposed solution</p>

8. Content / Conținuturi*



8.1. Lecture / Curs	Teaching strategies / Metode de predare	Remarks, details / Observații
Course 1: Basic concepts of object oriented programming. First Java programm. (AO2)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia
Course 2: Classes. Creation and initialization of objects. Static members. Access specifiers (KO, AO1, AO2)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia
Course 3: Objects comparisons. Inheritance. instanceof keyword. Packages (KO, AO1,AO2)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia
Course 4: Interfaces.Comparison interfaces. Relations between objects interfaces and classes (KO, AO1,AO2)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia
Course 5: Exceptions in Java. Java Language Exception hierarchy (KO, AO1,AO2)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia
Course 6: Collections in Java. generics (KO, AO1,AO2, AtO)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia
Course 7: Graphical user interfaces. AWT. Events handling (KO, AO1,AO2, AtO)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia
Course 8: Graphical user interfaces. SWING (KO, AO1,AO2, AtO)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia
Course 9: Database connection using JAVA. JDBC API (KO, AO1,AO2, AtO)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia
Course 10: Java I/O (KO, AO1,AO2, AtO)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia
Course 11: Data Serialization	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia



Course 12: NIO. New Java IO (KO, AO1,AO2, AtO)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia
Course 13: Threads in Java (KO, AO1,AO2, AtO)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia
Course 14: Threads in Java (KO, AO1,AO2, AtO)	Lecture, conversation, illustration	References: web.info.uvt.ro/~zflavia

Recommended bibliography / Bibliografie

- 1.K. Arnold, J. Gosling - "The Java Programming Language. Second Edition", Addison-Wesley, 1997
- 2.D. Arnow, G. Weiss - "Introduction to Programming Using Java. An Object-Oriented Approach", Addison-Wesley, 1998
- 3.G. Booch - "Object-Oriented Analysis and Design with Applications. Second Edition", Addison-Wesley, 1994
- 4.Ștefan Tănasă – Java de la 0 la expert, Ed. Polirom, Iași, 2005
- 5.Horia Georgescu – Introducere in universul Java, Ed Tehnică, 2002
- 6.Cristian Frăsinaru – Curs practic de Java, Ed. MatrixRom, 2005
- 7.Victoria Iordan, Flavia Micota – Introducere în Java, Ed. Eurostampa, 2010

8.2. Seminar, lab / Seminar, laborator	Teaching/learning strategies / Metode de predare/ învățare	Remarks, details / Observații
1.Creation of simple Java applications. Familiarization with Eclipse environment (OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
2.Arrays in Java. Command line arguments. String class (OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
3.Class and objects definitions (OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
4. Collections (OA1, OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
5. Simple inheritance in Java (OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
6.Inheritance and Interfaces (OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia



7.GUI. Swing. Simple components (OA1, OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
8.GUI. Swing. Complex components (OA1, OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
9.GUI. Swing. Canvas (OA1, OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
10. Database access using JDBC (OA1, OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
11. Java I/O examples (OA1, OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
12. Java NIO examples (OA1, OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
13. Objects serialization (OA1, OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
14. Threads (OA1, OA2)	Dialogue, collaborative learning, problem solving	References: web.info.uvt.ro/~zflavia
Recommended bibliography / Bibliografie 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

Object-oriented programming paradigms are commonly used for development of software application for medium to large size industries projects. The labor market local, national or European is always looking for graduates with good knowledge of programming in general and in particular object-oriented programming languages such as C ++ or Java.

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	<ul style="list-style-type: none"> • knowledge of object oriented programming (KO, AO) • Java language specific implementation of OO (KO, AO) • Java language libraries (KO, AO) 	Written exam in exams session	50%
	<ul style="list-style-type: none"> • the capacity of implementation of a problem into a OO (KO, AO) 	Practical test in exams session	30%
10.5. Seminar/ lab	<ul style="list-style-type: none"> • Capacity to follow the subjects proposed on courses (KO, AO) 	Homeworks during semester	10%
		Laboratory activity	5%
		Course activity	5%
10.6. Minimal knowledge for passing / Standard minim de performanță (Knowledge for 5 mark) <ul style="list-style-type: none"> • creation of an abstract data type (class) and make simple operation on a collection of objects • implement relation between classes: inheritance, aggregation • knowledge of object orient programming terminology • knowledge of specific Java language subjects presented at course 			
The final mark is calculated like ponder mean of the marks presented in section 10. The exam is promovated if the minimum mark is 5. At each exam sessions any of previous obtain marks is let in consideration if is bigger than 5.			

Date/ Data completării

10.10.2016

Signature (lecture) /
Semnătura titularului de curs
Flavia Micota

Signature (seminar)
Semnătura titularului de seminar
Flavia Micota
Monica Tirea

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