1.1 Higher education	Babes-Bolyai University
institution	
1.2 Faculty	Faculty of Mathematics and Computer Science
1.3 Department	Department of Computer Science
1.4 Field of study	Mathematics
1.5 Study cycle	Undergraduate
1.6 Study programme /	Mathematics and Computer Science
Qualification	

1. Information regarding the programme

2. Information regarding the discipline

2.1 Name of the discipline (en)			Algorithms and Programming			
(ro)			Algoritmi şi Programare			
2.2 Course coordinator			Conf. dr. Camelia Chira			
2.3 Seminar coordinator			Conf. dr. Camelia Chira			
2.4. Year of study 1	2.5 Semester	1	2.6. Type of evaluation	C	2.7 Type of discipline	Compulsory
2.8 Code of the discipline	MLE5115					

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	6	Of which: 3.2 course	2	3.3	2 sem
				seminar/laboratory	2 lab
3.4 Total hours in the curriculum	84	Of which: 3.5 course	28	3.6	56
				seminar/laboratory	
Time allotment:				·	hours
Learning using manual, course suppor	rt, bit	oliography, course note	S		14
Additional documentation (in libraries, on electronic platforms, field documentation)					12
Preparation for seminars/labs, homework, papers, portfolios and essays					14
Tutorship					8
Evaluations					18
Other activities:					
3.7 Total individual study hours 66					
3.8 Total hours per semester		150			
3.9 Number of ECTS credits		6			

4. Prerequisites (if necessary)

4.1. curriculum	•
4.2. competencies	•

5. Conditions (if necessary)

5.1. for the course	•	Projector
5.2. for the seminar /lab	•	Computers, Python programming language and environment
activities		

6. Specific competencies acquired

Professional competencies	 C1.1 Definition and description of programming paradigms and of language specific mechanisms, as well as identification of syntactic and semantic differences. C1.2 Description of existing software applications, on different levels of abstraction (architecture, classes, methods) using adequate basic knowledge. C1.3 Elaboration of adequate source code and testing of components in a well-known programming language, based on given specifications. C1.4 Testing applications based on testing plans. C1.5 Development of units of programs and corresponding documentation
Transversal	TC1 Application of efficient and rigorous working rules, manifest responsible attitudes towards the scientific and didactic fields, underlying the individual potential and respecting professional and ethical principles.
competencies	TC2 Use of efficient methods and techniques for learning, information, research and development of abilities for knowledge exploitation, for adapting to the needs of a dynamic society and for communication in a widely used foreign language.

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• To know the basic concepts of software engineering (design, implementation and maintenance) and to learn Python programming language
7.2 Specific objective of the discipline	 To know the key concepts of programming To know the basic concepts of software engineering To gain understanding of basic software tools used in development of programs To learn Python programming language and tools to develop, run, test and debug programs To acquire and improve a programming style according to the best practical recommendations

8. Content

8.1 Course	Teaching methods	Remarks
 Introduction to software development processes What is programming: algorithm, program, basic elements of the Python language, Python interpreter, basic roles in software engineering How to write programs: problem statement, requirements, feature driven development process Example: calculator 	 Interactive exposure Explanation Conversation Examples Didactical demonstration 	
2. Procedural programmingCompound types: list, tuple, dictionary	Interactive exposureExplanationConversation	

a. Test-driven development (TDD), refactoring • Didactical 3. Modular programming • Interactive exposure • What is a module. Packages, standard • Interactive exposure • Explanation • Conversation • Extraples • Didactical • Extraples • Didactical • How to define new data types: encapsulation, data hiding in Python, guidelines • Interactive exposure • Introduction to object-oriented programming • Examples • Abstract data types • Interactive exposure • Abstract data types • Interactive exposure • Introduction to object-oriented programming • Interactive exposure • Abstract data types • Interactive exposure • Interactive exposure • Examples • Didactical • Interactive exposure • Abstract data types • Didactical • Interactive exposure • Explanation • Object-oriented programming • Examples • Layered architecture: UI layer, application layer, infrastructure layer • Interactive exposure • Testing methods: exhausive testing, black box testing, white box testing, white box testing, white box testing, top on testing, white box testing, top on testing, white box testing, top on testing, top on testing, white box testing, top on			
• Test-driven development (TDD), refactoring demonstration 3. Modular programming • Interactive exposure • What is a module: Python module definition, variable scope in a module, packages, standard module libraries, deployment • Interactive exposure • Eclipse + PyDev • Didactical demonstration • Interactive exposure • How to define new data types: encapsulation, data hiding in Python, guidelines • Interactive exposure • Introduction to object-oriented programming • Examples • Exceptions • Interactive exposure • Abstract data types • Didactical demonstration • Implementation of classes in Python • Interactive exposure • Layrend architecture: UI layer, application layer, domain layer, infrastructure layer • Interactive exposure • How to organize source code: responsibility single responsibility principle, separation of concerns, dependency, coupling, cohesion • Interactive exposure • Faying methods: exkhaustive testing, black box testing, white box testing • Interactive exposure • File operations in Python • Interactive exposure		• Functions: test cases, definition, variable scope,	• Examples
3. Modular programming • Interactive exposure • What is a module: Python module definition, variable scope in a module, packages, standard module libraries, deployment • Interactive exposure • Excipace + PyDev • Explanation 4. User defined types • Interactive exposure • How to define new data types: encapsulation, data hiding in Python, guidelines • Interactive exposure • Introduction to object-oriented programming • Interactive exposure • Exceptions • Interactive exposure • Abstract data types • Interactive exposure • Implementation of classes in Python • Didactical demonstration • Object-oriented programming • Didactical demonstration • Conversation • Examples • Didactical demonstration • Conversation • Layered architecture: UI layer, application layer, idnomin layer, idnomin layer, idnomin layer, idnomin layer, idnomina			
 What is a module: Python module definition, variable scope in a module, packages, standard module libraries, deployment Eclipse + PyDev User defined types How to define new data types: encapsulation, data hiding in Python, guidelines Introduction to object-oriented programming Exceptions Object-oriented programming Abstract data types Implementation of classes in Python Object-oriented programming Abstract data types Implementation of classes; classes, objects, fields, methods, Python scope and namespace Software design guidelines Layered architecture: Ull ayer, application layer, domain layer, infrastructure layer How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion Program testing and inspection Automade testing, TDD File operations in Python Direct and indirect recursion Examples Complexity of algorithms Problem definition Search algorithms Problem definition Sorting algorithms Problem definition Sorting algorithms Complexity of algorithms Complexity of algorithms 		• Test-driven development (TDD), refactoring	demonstration
variable scope in a module, packages, standard module libraries, deployment • Conversation • Eclipse + PyDev • Examples • User defined types • Interactive exposure • How to define new data types: encapsulation, data hiding in Python, guidelines • Interactive exposure • Introduction to object-oriented programming • Examples • Didactical demonstration • Didactical demonstration • Object-oriented programming • Abstract data types • Interactive exposure • Implementation of classes in Python • Didactical demonstration • Object-oriented programming • Abstract data types • Interactive exposure • Layered architecture: UI layer, application layer, domain layer, infrastructure layer • Interactive exposure • Layered architecture: UI layer, application layer, domain layer, infrastructure layer • Interactive exposure • Testing methods: exhaustive testing, black box testing, white box testing • Automated testing, TDD • Interactive exposure • File operations in Python • Interactive exposure • Notion of recursion • Notion of recursion • Notion of recursion • Complexity of algorithms • Interactive exposure • Problem definition • Search halgorithms • Problem definition • Search methods: sequential, binary • Complexity of algorithms • Interactive exposure • Drodactical demonstra	3.	Modular programming	• Interactive exposure
module libraries, deploymentExamples• Eclipse + PyDev• Didactical demonstration4. User defined types• Interactive exposure• How to define new data types: encapsulation, data hiding in Python, guidelines• Interactive exposure• Introduction to object-oriented programming• Examples• Diject-oriented programming• Extramples• Abstrat data types• Didactical demonstration• Object-oriented programming• Explanation• Object-oriented programming• Interactive exposure• Abstrat data types• Didactical demonstration• Object-oriented programming• Explanation• Object-oriented programming• Interactive exposure• Abstrat data types• Didactical demonstration• Interactive exposure• Explanation• Conversation• Examples• Layered architecture: UI layer, application layer, domain layer, infrastructure layer • How to organize source coder responsibilities, single responsibility principle, separation of concers, dependency, coupling, cohesion • Testing methods: exhaustive testing, black box testing, white box testing • Notion of recursion • Direct and indirect recursion • Didactical demonstration• Interactive exposure • Explanation • Conversation • Conversation9. Search algorithms • Problem definition • Search methods: sequential, binary • Complexity of algorithms• Interactive exposure • Explanation • Conversation10. Sorting algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithms• Inter		• What is a module: Python module definition,	• Explanation
Eclipse + PyDev Didactical demonstration Lore defined types How to define new data types: encapsulation, data hiding in Python, guidelines Introduction to object-oriented programming Exceptions Didactical demonstration Lexamples Didactical demonstration Conversation Conversation Introductive exposure Explanation Conversation Conversation Introductive exposure Explanation Conversation Conversation Introductive exposure Explanation Conversation Conversatio		variable scope in a module, packages, standard	Conversation
 Eclipse + PyDev Didactical demonstration User defined types How to define new data types: encapsulation, data hiding in Python, guidelines Intractive exposure Exceptions Exceptions Object-oriented programming Exceptions Object-oriented programming Abstract data types Implementation of classes in Python Objects and classes: classes, objects, fields, methods, Python scope and namespace Software design guidelines Layered architecture: UI layer, application layer, domain layer, infrastructure layer How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion Program testing and inspection Testing methods: exhaustive testing, black box testing, white b		module libraries, deployment	• Examples
4. User defined types demonstration 4. User define two dafine new data types: encapsulation, data hiding in Python, guidelines intractive exposure introduction to object-oriented programming Explanation 2. Object-oriented programming Didactical demonstration 3. Object-oriented programming Didactical demonstration 4. May an encode the system of the system		• Eclipse + PyDev	-
 How to define new data types: encapsulation, data hiding in Python, guidelines Intraduction to object-oriented programming Exceptions Object-oriented programming Abstract data types Inplementation of classes in Python Objects and classes: classes, objects, fields, methods, Python scope and namespace Software design guidelines Layered architecture: UI layer, application layer, domain layer, infrastructure layer How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion Program testing and inspection Tretractive exposure Examples Didactical demonstration Conversation Examples Didactical demonstration Search methods: sequential, binary Complexity of algorithms Problem defi			demonstration
 How to define new data types: encapsulation, data hiding in Python, guidelines Introduction to object-oriented programming Exceptions Object-oriented programming Abstract data types Inplementation of classes in Python Objects and classes: classes, objects, fields, methods, Python scope and namespace Software design guidelines Layered architecture: UI layer, application layer, infrastructure layer How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion Program testing and inspection Theractive exposure Examples Didactical demonstration Conversation Explanation Conversation Explanation Conversation Interactive exposure Examples Didactical demonstration Conversation Explanation Conversation Examples Didactical demonstration <li< th=""><th>4.</th><th>User defined types</th><th>• Interactive exposure</th></li<>	4.	User defined types	• Interactive exposure
data hiding in Python, guidelines • Introduction to object-oriented programming • Exceptions• Conversation • Examples 		• 1	
 Introduction to object-oriented programming Exceptions Object-oriented programming Abstract data types Implementation of classes in Python Objects and classes: classes, objects, fields, methods, Python scope and namespace Software design guidelines Layered architecture: UI layer, application layer, domain layer, infrastructure layer How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion Program testing and inspection Testing methods: exhaustive testing, black box testing, white box testing Automated testing, TDD File operations in Python Notion of recursion Notion of recursion Notion of necursion Search algorithms Problem definition Search methods: sequential, binary Complexity of algorithms Problem definition Sort methods: Bubble Sort, Insertion Sort, Quick Sort Complexity of algorithms Problem definition Sort methods: Bubble Sort, Insertion Sort, Quick Sort Complexity of algorithms Problem definition Sort methods: Subble Sort, Insertion Sort, Quick Sort Complexity of algorithms 			-
 Exceptions Didactical demonstration Object-oriented programming Abstract data types Implementation of classes in Python Objects and classes: classes, objects, fields, methods, Python scope and namespace Software design guidelines Layered architecture: UI layer, application layer, domain layer, infrastructure layer How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion Program testing and inspection Testing methods: exhaustive testing, black box testing, white box testing Automated testing, TDD File operations in Python Recursion Notion of recursion Examples Didactical demonstration Conversation Explanation Conversation Examples Didactical demonstration Search algorithms Problem definition Search methods: sequential, binary Complexity of algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Problem definition Sort methods			
demonstration5. Object-oriented programming • Abstract data types • Implementation of classes in Python • Objects and classes: classes, objects, fields, methods, Python scope and namespace• Interactive exposure • Explanation • Conversation6. Software design guidelines • Layered architecture: UI layer, application layer, domain layer, infrastructure layer • How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion• Interactive exposure • Examples • Didactical demonstration7. Program testing and inspection • Tresting methods: exhaustive testing, black box testing, white box testing • Automated testing, TDD • File operations in Python• Interactive exposure • Explanation • Conversation • Examples • Didactical demonstration8. Recursion • Notion of recursion • Direct and indirect recursion • Examples • Computational complexity• Interactive exposure • Explanation • Conversation • Examples • Didactical demonstration9. Search algorithms • Problem definition • Soart methods: sequential, binary • Complexity of algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithms• Interactive exposure • Explanation • Conversation • Examples • Didactical demonstration10. Sorting algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithms• Interactive exposure • Explanation • Conversation • Examples 		• • • •	-
5. Object-oriented programming Interactive exposure • Abstract data types Explanation • Implementation of classes in Python • Conversation • Objects and classes: classes, objects, fields, methods, Python scope and namespace • Examples • Layered architecture: UI layer, application layer, domain layer, infrastructure layer • Interactive exposure • How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion • Interactive exposure 7. Program testing and inspection • Didactical demonstration • Automated testing, TDD • File operations in Python • File operations in Python • Interactive exposure • Notion of recursion • Interactive exposure • Notion of recursion • Interactive exposure • Notion of recursion • Examples • Computational complexity • Didactical demonstration 9. Search algorithms • Interactive exposure • Problem definition • Explanation • Complexity of algorithms • Interactive exposure • Problem definition • Conversation • Search methods: sequential, binary • Comversation • Problem definition • Interactive exposure			
 Abstract data types Implementation of classes in Python Objects and classes: classes, objects, fields, methods, Python scope and namespace Software design guidelines Layered architecture: UI layer, application layer, domain layer, infrastructure layer How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion Program testing and inspection Testing methods: exhaustive testing, black box testing, white box testing Automated testing, TDD File operations in Python Notion of recursion Direct and indirect recursion Examples Computational complexity Search algorithms Problem definition Search methods: sequential, binary Complexity of algorithms Problem definition Soft methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Complexity of algorithms Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Didactical demonstration 	5	Object-oriented programming	
 Implementation of classes in Python Objects and classes: classes, objects, fields, methods, Python scope and namespace Conversation Software design guidelines Layered architecture: UI layer, application layer, infrastructure layer How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion Program testing and inspection Tersting methods: exhaustive testing, black box testing, white box testing Automated testing, TDD File operations in Python Brecursion Direct and indirect recursion Examples Conversation Conversation Conversation Conversation Interactive exposure Explanation Conversation Conversation Interactive exposure Explanation Conversation Examples Didactical demonstration Interactive exposure Examples Didactical demonstration Interactive exposure Explanation Examples Conversation Conversation Conversation Examples Conversation Examples Conversation Interactive exposure Examples Didactical demonstration Interactive exposure Examples Conversation Examples Didactical demonstration Interactive exposure Examples Didactical demonstration Interactive exposure Examples Didactical demonstration Examples Didactical demonstration Examples Didactical demonstration Examples Didactical demonst		5 1 6 6	
 Objects and classes: classes, objects, fields, methods, Python scope and namespace Didactical demonstration Software design guidelines Layered architecture: UI layer, application layer, infrastructure layer How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion Program testing and inspection Testing methods: exhaustive testing, black box testing, white box testing Automated testing, TDD File operations in Python Direct and indirect recursion Examples Computational complexity Search algorithms Complexity of algorithms Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms 		• •	1
methods, Python scope and namespace• Didactical demonstration6. Software design guidelines • Layered architecture: UI layer, application layer, domain layer, infrastructure layer • How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion• Interactive exposure • Explanation • Didactical demonstration7. Program testing and inspection • Testing methods: exhaustive testing, black box testing, white box testing • Automated testing, TDD • File operations in Python• Interactive exposure • Explanation • Conversation • Examples • Didactical demonstration8. Recursion • Notion of recursion • Direct and indirect recursion • Examples • Computational complexity• Interactive exposure • Explanation • Conversation • Examples • Didactical demonstration9. Search algorithms • Problem definition • Search methods: sequential, binary • Complexity of algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithms• Interactive exposure • Explanation • Conversation 			
demonstration6. Software design guidelines• Layered architecture: UI layer, application layer, domain layer, infrastructure layer• How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion7. Program testing and inspection • Testing methods: exhaustive testing, black box testing, white box testing • Automated testing, TDD • File operations in Python8. Recursion • Notion of recursion • Direct and indirect recursion • Direct and indirect recursion • Problem definition • Search methods: sequential, binary • Complexity of algorithms9. Search algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithms10. Sorting algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithms10. Sorting algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithms10. Sorting algorithms10. Sorting algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithms			-
6. Software design guidelines Layered architecture: UI layer, application layer, domain layer, infrastructure layer How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion Interactive exposure Examples Didactical demonstration Automated testing, TDD File operations in Python Interactive exposure Examples Didactical demonstration Examples Didactical demonstration Examples Didactical demonstration Search algorithms Problem definition Search methods: sequential, binary Complexity of algorithms Problem definition Sorting algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Interactive exposure Examples Didactical demonstration 		incurous, i yulon scope and numespace	
 Layered architecture: UI layer, application layer, domain layer, infrastructure layer How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion Program testing and inspection Testing methods: exhaustive testing, black box testing, white box testing Automated testing, TDD File operations in Python Interactive exposure Examples Didactical demonstration Interactive exposure Explanation Conversation Examples Didactical demonstration Interactive exposure Explanation Conversation Examples Didactical demonstration Interactive exposure Explanation Conversation Examples Conversation Examples Computational complexity Search algorithms Problem definition Search methods: sequential, binary Complexity of algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Interactive exposure Examples Didactical demonstration Conversation Examples Didactical demonstration Conversation Examples Didactical demonstration Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Interactive exposure Explanation Conversation Examples Didactical demonstration 	6	Software design guidelines	
layer, domain layer, infrastructure layer• Conversation• How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion• Examples • Didactical demonstration7. Program testing and inspection • Testing methods: exhaustive testing, black box testing, white box testing • Automated testing, TDD • File operations in Python• Interactive exposure • Examples • Didactical demonstration8. Recursion • Notion of recursion • Examples • Computational complexity• Interactive exposure • Examples • Didactical demonstration9. Search algorithms • Problem definition • Search methods: sequential, binary • Complexity of algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithms• Interactive exposure • Examples • Interactive exposure • Examples • Didactical demonstration10. Sorting algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithms• Interactive exposure • Examples • Interactive exposure • Examples • Didactical demonstration	0.		-
 How to organize source code: responsibilities, single responsibility principle, separation of concerns, dependency, coupling, cohesion Program testing and inspection Testing methods: exhaustive testing, black box testing, white box testing Automated testing, TDD File operations in Python Recursion Didactical demonstration Conversation Conversation Examples Didactical demonstration Conversation Examples Didactical demonstration Recursion Notion of recursion Examples Didactical demonstration Conversation Conversation Examples Didactical demonstration Search algorithms Search methods: sequential, binary Complexity of algorithms Interactive exposure Examples Didactical demonstration Sorting algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Interactive exposure Examples Didactical demonstration Conversation Examples Didactical demonstration Conversation Examples Didactical demonstration 			-
single responsibility principle, separation of concerns, dependency, coupling, cohesionDidactical demonstration7. Program testing and inspection • Testing methods: exhaustive testing, black box testing, white box testing • Automated testing, TDD • File operations in PythonInteractive exposure • Exalanation • Conversation • Didactical demonstration8. Recursion • Notion of recursion • Direct and indirect recursion • Computational complexityInteractive exposure • Examples • Didactical demonstration9. Search algorithms • Problem definition • Complexity of algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithmsInteractive exposure • Examples • Didactical demonstration10. Sorting algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithmsInteractive exposure • Examples • Didactical demonstration10. Sorting algorithms • Complexity of algorithms • Complexity of algorithmsInteractive exposure • Examples • Didactical demonstration10. Sorting algorithms • Complexity of algorithms • Complexity of algorithmsInteractive exposure • Examples • Didactical demonstration			
concerns, dependency, coupling, cohesiondemonstration7. Program testing and inspection • Testing methods: exhaustive testing, black box testing, white box testing • Automated testing, TDD • File operations in Python• Interactive exposure • Examples • Didactical demonstration8. Recursion • Notion of recursion • Direct and indirect recursion • Computational complexity• Interactive exposure • Examples • Didactical demonstration9. Search algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, • Problem definition • Sort methods: Bubble Sort, Selection Sort, • Complexity of algorithms • Complexity of algorithms• Interactive exposure • Explanation • Conversation • Examples • Didactical demonstration10. Sorting algorithms • Complexity of algorithms • Complexity of algorithms • Complexity of algorithms • Complexity of algorithms• Interactive exposure • Explanation • Conversation • Examples • Didactical demonstration		U	-
7. Program testing and inspection Interactive exposure • Testing methods: exhaustive testing, black box Explanation • Automated testing, TDD Examples • File operations in Python Didactical 8. Recursion Interactive exposure • Notion of recursion Examples • Computational complexity Interactive exposure • Examples Didactical • Computational complexity Examples • Problem definition Examples • Problem definition Examples • Didactical demonstration • Didactical demonstration • Complexity of algorithms Interactive exposure • Problem definition Examples • Didactical demonstration • Conversation Examples • Didactical demonstration • Complexity of algorithms Interactive exposure • Problem definition Examples • Didactical Interactive exposure • Problem definition Explanation • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Interactive exposure • Complexity of algo		• • • • • •	
 Testing methods: exhaustive testing, black box testing, white box testing Automated testing, TDD File operations in Python File operations in Python Examples Didactical demonstration Interactive exposure Examples Conversation Examples Conversation Examples Conversation Examples Conversation Examples Conversation Examples Computational complexity Didactical demonstration Search algorithms Complexity of algorithms Problem definition Sorting algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Conversation Examples Didactical demonstration Interactive exposure Examples Didactical demonstration Sorting algorithms Conversation Explanation Conversation Interactive exposure Examples Didactical demonstration 	7		
testing, white box testing • Automated testing, TDD • File operations in Python• Conversation • Examples • Didactical demonstration8. Recursion • Notion of recursion • Direct and indirect recursion • Examples • Computational complexity• Interactive exposure • Explanation • Conversation • Examples • Didactical demonstration9. Search algorithms • Problem definition • Search methods: sequential, binary • Complexity of algorithms• Interactive exposure • Examples • Didactical demonstration10. Sorting algorithms • Problem definition 	/.	• • •	
 Automated testing, TDD File operations in Python File operations in Python File operations in Python Didactical demonstration Interactive exposure Examples Computational complexity Search algorithms Search methods: sequential, binary Complexity of algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Conversation Examples Didactical demonstration Interactive exposure Explanation Conversation Interactive exposure Examples Didactical demonstration Conversation Examples Didactical demonstration Conversation Examples Didactical demonstration 			-
 File operations in Python File operations in Python Didactical demonstration Interactive exposure Explanation Examples Computational complexity Search algorithms Complexity of algorithms Sorting algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Complexity of algorithms Didactical demonstration Interactive exposure Examples Didactical demonstration Interactive exposure Explanation Conversation Examples Didactical demonstration Interactive exposure Examples Didactical demonstration Examples Didactical demonstration 			
demonstration8. Recursion• Interactive exposure• Notion of recursion• Explanation• Direct and indirect recursion• Conversation• Examples• Didactical• Computational complexity• Didactical9. Search algorithms• Interactive exposure• Problem definition• Examples• Complexity of algorithms• Interactive exposure• Complexity of algorithms• Interactive exposure• Didactical• Examples• Didactical• Explanation• Sort methods: Bubble Sort, Selection Sort,• Interactive exposure• Complexity of algorithms• Interactive exposure• Complexity of algorithms• Didactical• Complexity of algorithms• Didactical• Didactical• Didactical• Complexity of algorithms• Didactical• Didactical• Didactical• demonstration• Didactical• Didactical• Didactical• Didactical• Didactical• Complexity of algorithms• Didactical• Didactica			÷
8. Recursion • Interactive exposure • Notion of recursion • Explanation • Direct and indirect recursion • Conversation • Examples • Didactical demonstration • Computational complexity • Interactive exposure 9. Search algorithms • Interactive exposure • Problem definition • Examples • Complexity of algorithms • Interactive exposure • Conversation • Examples • Didactical demonstration • Conversation • Complexity of algorithms • Interactive exposure • Problem definition • Examples • Didactical demonstration • Didactical demonstration 10. Sorting algorithms • Interactive exposure • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithms • Interactive exposure • Examples • Didactical demonstration		• File operations in Python	
 Notion of recursion Direct and indirect recursion Examples Computational complexity Search algorithms Problem definition Search methods: sequential, binary Complexity of algorithms Interactive exposure Examples Didactical demonstration Sorting algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Didactical Didactical demonstration 	0		
 Direct and indirect recursion Examples Conversation Examples Computational complexity Didactical demonstration Search algorithms Problem definition Search methods: sequential, binary Complexity of algorithms Interactive exposure Examples Didactical demonstration Sorting algorithms Interactive exposure Examples Didactical demonstration Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Didactical demonstration Didactical demonstration Didactical demonstration Didactical demonstration Didactical demonstration 	8.		-
 Examples Computational complexity Search algorithms Problem definition Search methods: sequential, binary Complexity of algorithms Interactive exposure Examples Didactical Conversation Examples Didactical demonstration Sorting algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Didactical Didactical Explanation Explanation Explanation Didactical demonstration 			1
 Computational complexity Didactical demonstration Search algorithms Problem definition Search methods: sequential, binary Complexity of algorithms Complexity of algorithms Examples Didactical demonstration Sorting algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Didactical demonstration 			
demonstration 9. Search algorithms • Problem definition • Search methods: sequential, binary • Complexity of algorithms • Complexity of algorithms • Didactical demonstration 10. Sorting algorithms • Problem definition • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Complexity of algorithms • Didactical demonstration		•	-
9. Search algorithms • Interactive exposure • Problem definition • Explanation • Search methods: sequential, binary • Conversation • Complexity of algorithms • Examples • Didactical • Interactive exposure • Problem definition • Examples • Problem definition • Interactive exposure • Problem definition • Interactive exposure • Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort • Examples • Complexity of algorithms • Didactical • Complexity of algorithms • Didactical		Computational complexity	
 Problem definition Search methods: sequential, binary Complexity of algorithms Explanation Conversation Examples Didactical demonstration Interactive exposure Explanation Explanation Examples Didactical demonstration Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Didactical demonstration Didactical demonstration 		a	
 Search methods: sequential, binary Complexity of algorithms Conversation Examples Didactical demonstration Sorting algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Conversation Explanation Examples Didactical demonstration 	9.	5	-
 Complexity of algorithms Examples Didactical demonstration Sorting algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Examples Didactical demonstration 			-
 Didactical demonstration 10. Sorting algorithms Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Didactical demonstration Explanation Conversation Examples Didactical demonstration 			
demonstration10. Sorting algorithms• Interactive exposure• Problem definition• Explanation• Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort• Conversation• Complexity of algorithms• Didactical demonstration		• Complexity of algorithms	-
10. Sorting algorithms• Interactive exposure• Problem definition• Explanation• Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort• Conversation• Complexity of algorithms• Didactical demonstration			
 Problem definition Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Examples Didactical demonstration 			
 Sort methods: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort Complexity of algorithms Complexity of algorithms Complexity of algorithms 	10.		-
Insertion Sort, Quick Sort• Examples• Complexity of algorithms• Didactical demonstration			• Explanation
Complexity of algorithms Didactical demonstration			Conversation
demonstration		Insertion Sort, Quick Sort	• Examples
		• Complexity of algorithms	• Didactical
11. Problem solving methods (I) • Interactive exposure			demonstration
	11.	Problem solving methods (I)	Interactive exposure

		1
 General presentation of the Backtracking, 	• Explanation	
Divide & Conquer methods	Conversation	
• Algorithms and complexity	• Examples	
• Examples	Didactical	
	demonstration	
12. Problem solving methods (II)	• Interactive exposure	
 General presentation of the Greedy and 	• Explanation	
Dynamic Programming methods	Conversation	
 Algorithms and complexity 	• Examples	
• Examples	Didactical	
	demonstration	
13. Revision	• Interactive exposure	
• Revision of most important topics covered by	Explanation	
the course	Conversation	
	• Examples	
	Didactical	
	demonstration	
14. Evaluation		
Bibliography		•
1. M.L. Hetland, Beginning Python: From Novice	e to Professional, Apress,	2005.
2. M. Frentiu, H.F. Pop, Fundamentals of Program	mming, Cluj University P	ress, 2006.
3. K. Beck, Test Driven Development: By Examp	ole. Addison-Wesley Lon	gman, 2002.
http://en.wikipedia.org/wiki/Test-driven_devel	lopment	
4. M. Fowler, Refactoring. Improving the Design	of Existing Code, Addise	on-Wesley, 1999.
http://refactoring.com/catalog/index.html		
5. The Python Programming Language - <u>https://v</u>		
6. The Python Standard Library - <u>https://docs.pyt</u>		<u>tml</u>
7. The Python Tutorial - <u>https://docs.python.org/</u>		
8.2 Seminar / laboratory	Teaching methods	Remarks
1. Simple Python programs	• Interactive exposure	
2. Procedural Programming	• Explanation	
3. Modular Programming	Conversation	
4. Feature-driven software development	Didactical	
5. Abstract data types	demonstration	
6. Design principles		
7. Object-oriented programming		
8. Program design. Layered architecture		
9. Inspection and testing		
10. Recursion. Complexity of algorithms		
11. Search and sorting algorithms		
12. Problem solving methods: Backtracking		
13. Problem solving methods: Greedy		
14. Practical test		
Bibliography		
1. M.L. Hetland, Beginning Python: From Novice	· · ·	
2. M. Frentiu, H.F. Pop, Fundamentals of Program	mming, Cluj University P	ress, 2006.
	mming, Cluj University P	ress, 2006.

- K. Beck, Test Driven Development: By Example. Addison-Wesley Longman, 200 <u>http://en.wikipedia.org/wiki/Test-driven_development</u>
- 4. M. Fowler, Refactoring. Improving the Design of Existing Code, Addison-Wesley, 1999. http://refactoring.com/catalog/index.html
- 5. The Python Programming Language <u>https://www.python.org/</u>
- 6. The Python Standard Library https://docs.python.org/3/library/index.html
- 7. The Python Tutorial <u>https://docs.python.org/3/tutorial/</u>

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- The course respects the IEEE and ACM Curricula Recommendations for Computer Science studies.
- The course exists in the studying program of all major universities in Romania and abroad.
- The content of the course is considered by the software companies as important for average programming skills.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)			
10.4 Course	The correctness and completeness of the accumulated knowledge and the capacity to design and implement correct Python programs	Written exam	40%			
10.5 Seminar/lab activities	Be able to design, implement and test a Python program	Practical exam	30%			
	Correctness of laboratory assignments and documentation delivered during the semester	Program and documentation	30%			
10.6 Minimum performance standards						
		e lab activity, practical test ar	nd written			

Date	Signature of course coordinator	Signature of seminar coordinator
27.04.2021	Conf. univ. dr. Camelia Chira	Conf. univ. dr. Camelia Chira

Date of approval

Signature of the head of department

.....

Prof. dr. Laura Dioșan