



COURSE DESCRIPTION

1. General information

1.1 University	“Alexandru Ioan Cuza”, University of Iași
1.2 Faculty	Faculty of Computer Science
1.3 Department	Department of Computer Science
1.4 Study domain	Computer Science
1.5 Study cycle	Bachelor
1.6 Study program / Qualification	Computer Science (english)

2. Course Information

2.1 Discipline name	Principles of Programming Languages						
2.2 Course teacher	Arusoai Andrei						
2.3 Seminar teacher	Arusoai Andrei						
2.4 Year of study	2	2.5 Semester	1	2.6 Evaluation type	EVP	2.7 Discipline status*	OP

* OB – Mandatory / OP – Opțional / EVP – evaluation during the semester, no final exams

3. Total estimated hours (hours per semester and didactic activities)

3.1 Hours per week	4	In which: 3.2 course	2	3.3 seminar/laboratory	2
3.4 Hours in curriculum	56	In which: 3.5 course	28	3.6 seminar/laboratory	28
Time distribution					hours
Manual study, course support, bibliography, and others					56
Supplementary documentation in the library, electronic forums, and on the field					10
Seminaries/laboratories preparation, homeworks, reports, portfolios, and essays					30
Tutoring					0
Evaluation					4
Other activities					0
3.7 Total hours per individual					44
3.8 Total hours per semester					100
3.9 Credits					4

4. Preconditions (in necessary)

4.1 Curriculum	Logics for Computer Science, Data structures, Object-Oriented Programming
4.2 Competencies	Imperative programming

5. Conditions (if necessary)

5.1 For course operation	The course will mainly take place in physical format (face-to-face). Online operation will be partially considered depending on opportunity, necessity and specific means availability, and will not account for more than a maximum of 22% of the total course time
5.2 For seminary/laboratory operation	The laboratories will mainly take place in physical format (face-to-face). Online operation will be partially considered depending on opportunity, necessity and specific means availability, and will not account for more than a maximum of 22% of the total course time



6. Specific skills acquired

Professional skills	C1. Defining syntax for programming languages C2. Defining semantics for programming languages C3. Using a framework for defining programming languages
Transversal skills	CT1. Capacity to design a new programming language CT2. Capacity to define formal semantics for programming languages CT3. Capacity to create an interpreter for a programming language CT4. Capacity to learn fast a new programming language

7. Discipline objectives (din grila competențelor specifice acumulate)

7.1 General objective	Presenting in a simple and correct manner the main concepts of programming languages and a formal framework that allows defining programming languages and running programs written in that languages.
7.2 Specific objectives	After taking this course, students will be able to : <ul style="list-style-type: none">▪ Explain concepts specific to various programming language paradigms▪ Describe the syntax and the semantics of a programming language▪ Use a framework for defining a programming language▪ Design a programming language

8. Contents

8.1	Lecture	Teaching methods	Observations (hours and bibliography)
1.	Introduction to Programming languages. Overview.	Slides, blackboard	2 hours
2.	Coq: algebraic datatypes, inductive definitions, functions, demonstratii simple.	Slides, blackboard	2 hours
3.	Induction. Induction principles. Proofs by induction. Polymorphism. Higher-order functions. Logic in Coq.	Slides, blackboard	2 hours
4.	Concrete syntax vs abstract syntax: Backus-Naur Form, ambiguities, the syntax of IMP.	Slides, blackboard	2 hours
5.	An evaluator for IMP.	Slides, blackboard	2 hours
6.	Structural Operational Semantics: Big-step SOS.	Slides, blackboard	2 hours
7.	Structural Operational Semantics: Small-step SOS.	Slides, blackboard	2 hours
8.	Midterm lab exam.	Slides, blackboard	2 hours
9.	Type systems.	Slides, blackboard	2 hours
10.	Certified compilation.	Slides, blackboard	2 hours



11.	Paradigms: object-oriented programming	Slides, blackboard	2 hours
12.	Paradigms: functional programming(*)	Slides, blackboard	2 hours
13.	Paradigms: logical programming(*)	Slides, blackboard	2 hours
14.	Program verification: an introduction.(*) Evaluation.	Slides, blackboard	2 hours

References

1. **Practical Foundations of Programming Languages**, Robert Harper, Cambridge University Press 2016, <https://www.cs.cmu.edu/~rwh/pfpl/2nded.pdf>
2. **Software Foundations - Vol 2**, Benjamin C. Pierce, Arthur Azevedo de Amorim, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Cătălin Hrițcu, Vilhelm Sjöberg, Andrew Tolmach, Brent Yorgey, Online book: <https://softwarefoundations.cis.upenn.edu/current/index.html>
3. **Programming Languages: Principles and Paradigms**, Maurizio Gabbriellini, Simone Martini, Online: [http://websrv.dthu.edu.vn/attachments/newsevents/content2415/Programming_Languages - Principles and Paradigms thereads1106.pdf](http://websrv.dthu.edu.vn/attachments/newsevents/content2415/Programming_Languages_-_Principles_and_Paradigms_thereads1106.pdf)
4. **The formal semantics of Programming Languages – An Introduction**, Glynn Winskell. The MIT Press, Cambridge, Massachusetts, 1993. ISBN 978-0-262-23169-5. Online: <https://mitpress.mit.edu/books/formal-semantics-programming-languages>

8.2	Seminar / Laboratory	Teaching methods	Observations (hours and bibliography)
1.	Installing Coq. Main working setup.	Free discussions. Solve various issues.	2 hours
2.	Algebraic datatypes. Recursive functions. Tactics and proofs.	Resolving the exercise sheet.	2 hours
3.	Proof by induction. Polymorphism and higher-order functions.	Resolving the exercise sheet.	2 hours
4.	Abstract syntax in Coq. Notations. Improving IMP with new constructs.	Resolving the exercise sheet.	2 hours
5.	Defining an interpreter for IMP.	Resolving the exercise sheet.	2 hours
6.	Big-step SOS: new features for IMP.	Resolving the exercise sheet.	2 hours
7.	Small-step SOS: new features for IMP.	Resolving the exercise sheet.	2 hours
8.	Midterm exam.	Exam.	2 hours
9.	Type systems.	Resolving the exercise sheet.	2 hours
10.	Certified compilation.	Resolving the exercise sheet.	2 hours
11.	The REC language.	Resolving the exercise sheet.	2 hours
12.	Untyped lambda calculus.(*)	Resolving the exercise sheet.	2 hours



13.	Defining a stack language.(*)	Resolving the exercise sheet.	2 hours
14.	Program verification: an introduction.(*)	Resolving the exercise sheet.	2 hours
References <ol style="list-style-type: none">Software Foundations - Vol 2, Benjamin C. Pierce, Arthur Azevedo de Amorim, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Cătălin Hrițcu, Vilhelm Sjöberg, Andrew Tolmach, Brent Yorgey, Online book: https://softwarefoundations.cis.upenn.edu/current/index.htmlThe formal semantics of Programming Languages – An Introduction, Glynn Winskell. The MIT Press, Cambridge, Massachusetts, 1993. ISBN 978-0-262-23169-5. Online: https://mitpress.mit.edu/books/formal-semantics-programming-languages			

(*) Activities marked in this manner can take place online, according to conditions mentioned at point 5, using specific methods assisted by technology.

9. Course content synchronization with the expectations of the community representatives, professional associations and employers from the program domain

This discipline aims to develop the ability to learn quickly a new programming language, and to develop the necessary skills to design and implement a new programming language. Students will be able to easily adapt to whatever (domain specific) programming languages that software companies are using.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 The weight of each evaluation form (%)
10.4 Course	50 points	Exam	50%
10.5 Seminar/ Laboratory	50 points	Lab assignments	50%
10.6 Minimal requirements for graduation : <ul style="list-style-type: none">Define the syntax of a programming language using BNFDefine the semantics of a programming language using operational semanticsDefine an interpreter for a programming language using a dedicated framework			
There are 50 points in total for the lab assignments. For distinguished solutions and supplementary work the students can obtain special bonuses for their final grade.			

Issue date,

Course teacher,

Seminar teacher,

Andrei Arusoiaie

Andrei Arusoiaie

Date of approval,

Department director,

Dorel Lucanu