



COURSE DESCRIPTION

1. Program 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

2. Course Information

2.1 Discipline name	Introduction to .NET						
2.2 Course teacher	Lect.univ.dr. Arusoaie Andrei						
2.3 Seminar teacher	Drd.Olariu Florin						
2.4 Year of study	2	2.5 Semester	1	2.6 Evaluation type	EVP	2.7 Discipline status*	OP

* OB – Obligatoriu / OP – Opțional

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					60
Seminaries/laboratories preparation, homeworks, reports, portfolios, and essays					0
Tutoring					0
Evaluation					4
Other activities					0
3.7 Total hours per individual					64
3.8 Total hours per semester					120
3.9 Credits					4

4. Preconditions (in necessary)

4.1 Curriculum	-
4.2 Competencies	-

5. Conditions (if necessary)

5.1 For course operation	Lecture room
5.2 For seminary/laboratory operation	Laboratory room

6. Specific skills acquired



Professional skills	C1. Studying C# language C2. Using .NET Core Framework to build systems and scalable architectures C3. Integrating .NET with SGBS's and NoSQL
Transversal skills	CT1. Capacity to design and develop scalable architectural systems CT2. Capacity to create business applications, decoupled and layered CT3. Capacity to learn and understand easily object oriented principles and design patterns like: SOLID, Grasp, DI/loC, MVC and web API

7. Discipline objectives (din grila competențelor specifice acumulate)

7.1 General objective	Presenting in a simple and easy to understand the necessity of building scalable web applications based to OOP principles and understanding of an architectural pattern aplicable with .NET Core Framework.
7.2 Specific objectives	After taking this course, students will be able to : <ul style="list-style-type: none">▪ Explain concepts specific to high level programming language(C#)▪ Describe and build scalable applications▪ Use the best practices▪ Design an application based to decoupled functional blocks▪ Follow up code quality metrics by using unit testing techniques

8. Contents

8.1	Lecture	Teaching methods	Observations (hours and bibliography)
1.	Introduction to .NET Core – OOP, SOLID, GRASP, CQS	Slides, blackboard	2 hours
2.	C# fundamentals	Slides, blackboard	2 hours
3.	Managing collections in C#	Slides, blackboard	2 hours
4.	LINQ concepts and Lambda expressions	Slides, blackboard	2 hours
5.	Using unit testing and integration testing to increase code quality	Slides, blackboard	2 hours
6.	Entity Framework Core – ORM – Introduction and general concepts	Slides, blackboard	2 hours
7.	Entity Framework Core – ORM – advanced topics	Slides, blackboard	2 hours
8.	Building an API using .NET Core API	Slides, blackboard	2 hours



9.	Building an web application using .NET Core MVC	Slides, blackboard	2 hours
10.	Design Patterns in .NET Core - basic concepts	Slides, blackboard	2 hours, Factory pattern, Builder pattern, Strategy pattern
11.	Design Patterns in .NET Core - advanced concepts	Slides, blackboard	2 hours, IoC/DI /Adapter pattern, Repository Pattern
12.	Architectural patterns used by business and distributed applications	Slides, blackboard	2 hours Onion/CQRS/Event Sourcing
13.	Introduction to Domain Driven Design concept	Slides, blackboard	2 hours
14.	Applying best practices in business and distributed systems	Slides, blackboard	2 hours
Bibliography Main references: <ol style="list-style-type: none"> https://www.microsoft.com/net/learn/get-started/windows https://www.pluralsight.com https://www.asp.net/ http://www.udemy.com https://packtpub.com http://www.amazon.com https://mva.microsoft.com 			
8.2	Seminar / Laboratory	Teaching methods	Observations (hours and bibliography)
1.	Installation of .NET Core/ Visual Studio Community/ VS Code, SQL Server 2016 Main working setup.	Free discussions. Coding kata sessions. Bonus system.	2 hours, kframework.org
2.	Learning C#	Resolving the exercise sheet.	2 hours
3.	Defining the semantics of IMP: krun.	Resolving the exercise sheet.	2 hours
4.	Designing C# classes using SOLID principles	Resolving the exercise sheet.	2 hours
5.	Using LINQ in collection manipulation	Resolving the exercise sheet.	2 hours
6.	Using Unit tests and integration tests in Entity Framework Core	Resolving the exercise sheet.	2 hours
7.	Midterm	Midterm test	2 hours; live test in the lab.
8.	Presentation of the project proposals .	Demo and examples. Free discussions	2 hours
9.	Presentation of the project proposals.	The teacher presents all the project proposals. Students have to decide the project they will develop until the end of the semester.	2 hours



10.	Project: Requirements, Implementations and Diagrams. Collaboration	Demo. Evaluation. Free discussions.	2 hours
11.	Project: Requirements, Implementations and Diagrams. Collaboration	Demo. Evaluation. Free discussions.	2 hours
12.	Project: Requirements, Implementations and Diagrams. Collaboration	Demo. Evaluation. Free discussions.	2 hours
13.	Project: Requirements, Implementations and Diagrams. Collaboration	Demo. Evaluation. Free discussions.	2 hours
14.	Project: final presentation.	Final evaluation. Direct discussions.	2 hours
Bibliography 1. https://www.microsoft.com/net/learn/get-started/windows 2. https://www.pluralsight.com			

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, C#. This language is an alternative to Java. Given that .NET Core is portable it is now adopted by companies. .NET Core presentation and orientation to business applications gives the possibility for attendees to understand aspects like: design patterns and architectural matters. All of those offers adaptability and easy integration within the existing IT Companies. 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	Test	50%
10.5 Seminar/ Laboratory	20 + 30 points	Project	50%
10.6 Minimal requirements for graduation : 50 points			
Project points can be obtained in two stages: preliminary presentation (=20 points) and final presentation (=30 points). The points obtained at the preliminary presentation are valid only if the final presentation has been done.			

Proposal date,

Course teacher,
Lect.univ.dr. Arusoaie AndreiSeminary/Lab teacher,
Drd. Olariu Florin

Department acceptance date,

Department director,