

SYLLABUS

Programming Environments and tools

1. Information on academic programme

1.1. University	„1 Decembrie 1918”
1.2. Faculty	Faculty of Informatics and Engineering
1.3. Department	Informatics, Mathematics and Electronics Department
1.4. Area	Computer Science
1.5. Level	undergraduate
1.6. Specialization	Computer Science/ ESCO: 2511/ Systems Analyst, 2512/ Software developers; Analyst-251201, Computer System Programmer -251204, Computer System Engineer – 251203.

2. Information of Course Matter

2.1. Course	<i>Programming Environments and tools</i>		2.2. Code	CSE 306			
2.3. Course Leader	Incze Arpad						
2.4. Seminar Tutor	Incze Arpad						
2.5. Academic Year	III	2.6. Semester	I	2.7. Type of Evaluation (E – final exam/ CE - colloquy examination / CA -continuous assessment)	CE	2.8. Type of course (C– Compulsory, Op – optional, F - Facultative)	Op

3. Course Structure (Weekly number of hours)

3.1. Weekly number of hours	6	3.2. course	2	3.3. seminar, laboratory	4
3.4. Total number of hours in the curriculum	84	3.5. course	28	3.6. seminar, laboratory	56
Allocation of time:					Hours
Individual study of readers					20
Documentation (library)					20
Home assignments, Essays, Portfolios					20
Tutorials					-
Assessment (examinations)					6
Other activities.....					-

3.7 Total number of hours for individual study	66
3.8 Total number of hours in the curriculum	84
3.9 Total number of hours per semester	150
3.10 Number of ECTS	6

4. Prerequisites (*where applicable*)

4.1. curriculum-based	<i>Mathematical Modelling and Simulation</i>
-----------------------	---

4.2. competence-based	<p>C3.3. The use of computer software environments to solve specific problems in the application field.</p> <p>C3.4. Data and model analysis.</p> <p>C3.5. The development of software components of interdisciplinary projects.</p>
-----------------------	--

5. Requisites (where applicable)

5.1. course-related	Classroom equipped with video projector / board. Microsoft Teams for online courses
5.2. seminar/laboratory-based	<i>Laboratory – computers, Software: Microsoft Visual Studio, Internet access.</i>

6. Specific competences to be acquired (chosen by the course leader from the programme general competences grid)

Professional competences	<p>C2. Development and maintenance of computer applications</p> <p><i>C2.4. The use of appropriate criteria and methods for the evaluation of computer applications.</i></p> <p><i>C2.5. The development of dedicated computer projects.</i></p> <p>C3. The use of computer tools in an interdisciplinary context</p> <p><i>C3.1. The description of concepts, theories and models used in the application field.</i></p> <p><i>C3.2 The identification and explanation of base computer models that are suitable for the application domain.</i></p> <p><i>C3.3. The use of computer and mathematical models and tools to solve specific problems in the application field.</i></p> <p><i>C3.4. Data and model analysis.</i></p> <p><i>C3.5. The development of software components of interdisciplinary projects.</i></p>
Transversal competences	CT3. The use of efficient methods and techniques for learning, scientific inquiry and development of the capacities of using knowledge, of adapting to a dynamic society and of communication in English.

7. Course objectives (as per the programme specific competences grid)

7.1 General objectives of the course	This course aims to introduce students to the basic concepts and features of C# programming. It is hoped that the knowledge would enhance the programming expertise of students to enable them develop C# based applications.
7.2 Specific objectives of the course	<p>By taking this course the students will be able to:</p> <p>Explain the term C# (C Sharp)</p> <ul style="list-style-type: none"> ● Clarify the origin of C# ● List the versions of C# ● Outline the basic features of C# ● Outline the design goals ● List the categories of C# Type system ● Explain the concept of boxing and unboxing ● Declare a variable in C#

	<ul style="list-style-type: none"> • Describe the naming conventions • Identify common variables in C# • Describe statements, statement blocks and comments • State the minimal requirement to use C# • Outline the steps involved in building console applications • State the procedure for building and running GUI applications • Outline the steps required to build a code library • Create a C# project in VisualStudio.NET • Identify C# expressions • List common operators used in C#
--	---

8. Course contents

8.1 Course (learning units)	Teaching methods	Remarks
1. <i>C# FUNDAMENTALS. C# and .NET Framework</i>	<i>Lecture, presentation, conversation, discussion. PowerPoint/Teams video tutorial</i>	2h
2. <i>C# TYPES. C# Type System. Boxing and unboxing. C# Data Types</i>	<i>Lecture, presentation, conversation, discussion. PowerPoint/Teams video tutorial</i>	2h
3. <i>LANGUAGE BASICS. Naming Conventions. C# Syntax.</i>	<i>Lecture, presentation, conversation, discussion. PowerPoint/Teams video tutorial</i>	2h
4. <i>LANGUAGE BASICS. Getting started with C#.</i>	<i>Lecture, presentation, conversation, discussion. PowerPoint/Teams video tutorial</i>	2h
5. <i>C# APPLICATIONS. Creating Console Assemblies.</i>	<i>Lecture, presentation, conversation, discussion. PowerPoint/Teams video tutorial</i>	2h
6. <i>C# APPLICATIONS. Creating GUI Assemblies</i>	<i>Lecture, presentation, conversation, discussion. PowerPoint/Teams video tutorial</i>	6h
7. <i>C# APPLICATIONS. Creating Code Library Assemblies</i>	<i>Lecture, presentation, conversation, discussion. PowerPoint/Teams video tutorial</i>	4h
8. <i>VISUAL STUDIO.NET. Creating a Project.</i>	<i>Lecture, presentation, conversation, discussion. PowerPoint/Teams video tutorial</i>	4h
9. <i>VISUAL STUDIO.NET. Language Concepts.</i>	<i>Lecture, presentation, conversation, discussion. PowerPoint/Teams video tutorial</i>	2h

10. <i>VISUAL STUDIO.NET. C# Expressions and Operators</i>	<i>Lecture, presentation, conversation, discussion. PowerPoint/Teams video tutorial</i>	2h
--	---	----

References

1. Abelson, H and Gerald J. S. (1997). *Structure and Interpretation of Computer Programs*. The MIT Press.
2. Armstrong, Deborah J. (2006). "The Quarks of Object-Oriented Development". *Communications of the ACM* **49** (2): 123–128. <http://portal.acm.org/citation.cfm?id=1113040>. Retrieved 2006-08-08.
3. Booch, Grady (1997). *Object-Oriented Analysis and Design with Applications*. Addison-Wesley.
4. Date, C. J and Hugh, D. (2006). *Foundation for Future Database Systems: The Third Manifesto* (2nd Edition)
5. Date, C. J and Hugh, D. (2007). *Introduction to Database Systems: The Sixth Manifesto* (6th Edition)
6. John C. Mitchell, *Concepts in programming languages*, Cambridge University Press, 2003, p.278
7. Joyce, F. (2006). *Microsoft Visual C#.NET with Visual Studio 2005*
8. Kay, Alan. *The Early History of Smalltalk*.
<http://gagne.homedns.org/%7etgagne/contrib/EarlyHistoryST.html>.
9. Martin, A and Luca, C. (2005). *A Theory of Objects*.
10. Michael Lee Scott (2006). *Programming language pragmatics*, (2nd Edition) p. 470

Seminars+laboratories	Teaching methods	remarc
<i>.NET Framework. Programs compilation</i>	<i>Project-work, computer-based activities, laboratory activities</i>	2+2
<i>C# language. Console applications.</i>	<i>Project-work, computer-based activities, laboratory activities</i>	2+2
<i>Structure of the C# program. Language syntax. Data types. Conversions. Applications.</i>	<i>Project-work, computer-based activities, laboratory activities</i>	2+2
<i>Constants. Variables. Expressions and operators. Applications.</i>	<i>Project-work, computer-based activities, laboratory activities</i>	2+2
<i>Collections and libraries. Applications.</i>	<i>Project-work, computer-based activities, laboratory activities</i>	2+2
<i>Foreach instruction. Applications.</i>	<i>Project-work, computer-based activities, laboratory activities</i>	2+2
<i>Try-catch-finally and throw. Applications.</i>	<i>Project-work, computer-based activities, laboratory activities</i>	2+2
<i>Visual programming environment C#. Applications.</i>	<i>Project-work, computer-based activities, laboratory activities</i>	4+4
<i>Windows. Controls. Applications.</i>	<i>Project-work, computer-based activities, laboratory activities</i>	6+6
<i>System. drawing. Applications.</i>	<i>Project-work, computer-based activities, laboratory activities</i>	4+4

9. Corroboration of course contents with the expectations of the epistemic community's significant representatives, professional associations and employers in the field of the academic programme

The knowledge C Sharp is increasingly valued in programming for various domains such as: the media, medicine, Web services, presentation of companies and organisations on the Internet. There are plenty of employment opportunities at local, regional and international level.

10. Assessment

Activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	<i>Final evaluation</i>	<i>Practic exam/project</i>	60%
10.5 Seminar/laboratory	<i>Continuous assessment</i>	<i>Laboratory activities portfolio</i>	40%
10.6 Minimum performance standard: minimum grade 5 at each criteria			
Modelling and solving problems of average complexity, using mathematics and computer science.			

Submission date

Course leader signature

Seminar tutor signature

Date of approval by Department members

Department director signature
