.NET platform - course description

General information

Course name	.NET platform
Course ID	11.3-WE-INFP-Platf.NET-Er
Faculty	Faculty of Computer Science, Electrical Engineering and Automatics
Field of study	Computer Science
Education profile	academic
Level of studies	Erasmus programme
Beginning semester	winter term 2017/2018

Course information

Semester	6
ECTS credits to win	4
Course type	optional
Teaching language	english
Author of syllabus	• dr hab. inż. Marek Sawerwain, prof. UZ

Classes forms

The class form	Hours per semester (full-time)	Hours per week (full-time)	Hours per semester (part-time)	Hours per week (part-time)	Form of assignment
Lecture	30	2	-	-	Exam
Laboratory	30	2	-	-	Credit with grade
Project	15	1	-	-	Credit with grade

Aim of the course

- Familiarize students with concept and the role of the IT platform in the daily practice of IT developer.
- Familiarize students with basic information about the .NET platform as a development environment for the traditional desktop applications and for internet applications (ASP.NET).
- Learn basic skills in C# programming. Using database systems. Describing data using XML. Creation dynamic web pages in ASP.NET technology.
- Shaping the basic skills of creating network services including the security of .NET applications. Familiarize students with other .NET programming languages e.g. F#.

Prerequisites

Programming fundamentals, Object oriented programming, Algorithms and Data Structures, Databases

Scope

Introduction to .NET platform. Structure of the .NET platform. .NET distributions. Outline of .NET Framework environment.

Review of programming languages supported by .NET platform. Microsoft Visual Studio - environment characteristic.

Presentation of programming environment. Tool for form editing. Running application. Creating sample application.

Common Language Runtime. Elementary functions and services of CLR. Memory and other resources management. Thread management. Structure and configuration of metadata. Integration with Win32 DLL libraries. Methods of interaction between applications. Comparison between CLR and JVM.

Fundamentals of C# programming. Language syntax: instructions, variables, operators and data types. Design principles of classes, methods, constructors and objects. Arrays usage guidelines.

Software Development Kit – review of main programming tools.

Advanced C# programming. Preprocessor directives. Event handling. Handling errors using exceptions. Common string operations. Regular expressions reference. Remote object invocation. File access operations. Thread synchronization. Base Class Library – review. User interface components.

Introduction to functional programming in F#: Introduction to F#. Review of functional programming style. Operators and data structures.

Creating components in .NET. Principles of designing, implementing and testing components. COM and COM+ technology overview.

XML in .NET. Methods of information transfer using XML documents. Review of classes for XML documents manipulation and transformation.

Access data using ADO.NET. Review of ADO.NET objects. Database access methods.

Language Integrated Query – LINQ. Architecture of LINQ technology. LINQ queries to objects, databases, SQL databases and XML data. Parallel and serial LINQ queries.

ASP.NET technology. Base classes and main objects of ASP.NET. Using XML in ASP.NET. Designing web pages using ASP components.

Creating web services. SOAP and UDDI protocols.

Security features of ASP.NET applications: access control, authentication and data encoding and cryptography.

Teaching methods

Lecture: conventional lecture

Laboratory: laboratory exercises, group work

Project: project method, discussions and presentations

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
Student is oriented to the capabilities of .NET in the field of application security and security of information generated by users or other .NET applications.		• a test with score scale	• Lecture
Student is able to create an application for the .NET environment that uses its capabilities.		• a test with score scale • an observation and evaluation of activities during the classes	• Laboratory
Can create documentation for new projects and existing .NET libraries / packages / packages.		• a test with score scale • an observation and evaluation of activities during the classes	• Laboratory
Can analyze existing problems and identify ways and techniques for solving these problems using .NET platform.		• a project • sprawozdanie z projektu	• Project
Student is familiar with the basic components of the platform and is able to characterize the advantages and disadvantages of .NET platform. Student is also aware of dynamic development of .NET platform.		• a test with score scale	• Lecture
Student is able to work on IT system and is familiar with the team tools for workgroup.		• a project • carrying out laboratory reports	• Project
Knows ethical behavior and licensing issues for third-party packages / components / libraries used in .NET projects.		• a test with score scale	• Lecture
Can create new components (and develop existing components) for solving IT problems in a basic way. Student is also familiar with the structure of .NET components.		• a test with score scale	• Laboratory

Assignment conditions

Lecture - obtaining a positive grade in written exam.

Laboratory - the main condition to get a pass are sufficient marks for all exercises and tests conducted during the semester.

Project - a condition of pass is to obtain positive marks from all project tasks and preparation written report of project.

Calculation of the final grade: = lecture 40% + laboratory 30% + project 30%.

Recommended reading

1. Chappell D., Understanding .NET (2nd Edition), Addison-Wesley Professional, 2nd edition, 2006.
2. Duffy J.: Professional .NET Framework 2.0 (Programmer to Programmer), Wrox, 2006.
3. Michelsen K.: C# Primer Plus, Sams Publishing, 2007.
4. Chadwick J., Snyder T., Panda H.,Programming ASP.NET MVC 4: Developing Real-World Web Applications with ASP.NET MVC, O'Reilly Media, 2012.
5. Magennis T., LINQ to Objects Using C# 4.0: Using and Extending LINQ to Objects and Parallel LINQ (PLINQ), Addison-Wesley Microsoft Technology, 2010.
6. Burton K.: .NET Common Language Runtime Unleashed, Sams Publishing, 2002.
7. Solis D.M.: Illustrated C# 2010, A-Press, 2010.
8. Löwy J.,Programming WCF Services: Mastering WCF and the Azure AppFabric Service Bus, O'Reilly Media, 2010.

Further reading

1. Novák. I, Velvárt A., Granicz A., Balássy G., Hajdrik A., Sellers M., Hillar G.C., Molnár A., Kanjilal J.: Visual Studio 2010 and .NET 4 Six-in-One, Wiley Publishing, Inc., 2010.
2. Johnson B., Professional Visual Studio 2013, Wiley Publishing, Inc., 2014.
3. Nash T.: Accelerated C# 2010, A-Press, 2010.
4. Troelsen A.: Pro C# 5.0 and the .NET 4.5 Platform, 6th Ed., A-Press, 2012.
5. Freeman A., Rattz J.C. Jr.: Pro LINQ: Language Integrated Query in C#, A-Press, 2010.
6. Richter J., CLR via C#, 3rd edition, Microsoft Press, 2010.

Notes

Modified by dr hab. inż. Marek Sawerwain, prof. UZ (last modification: 08-05-2017 21:02)