

.NET

Syllabus

Topic: Basics of Software Development in C#

Lecturer: Miloš Kosanović

Course objectives:

The course main objective is to teach students the basics of software development. Other goals are to teach and enable students to think algorithmically, to enable them to analyze and breakdown real-life problem into algorithmic steps, and then implement these steps in some programming language (in this case C#). Student will also learn to recognize and use basic and advanced data structures like arrays, strings, multidimensional arrays, hash tables, stack and queues. They will learn to write, compile, build and debug program in Visual Studio 2015. Prior knowledge is not needed. Special attention will be given to practical work, coding, and the usage of available online services that enable student to learn programming concepts through practical examples and games.

Course outcomes:

The participants will acquire:

- problem solving skills – to analyze real life problem, find and develop algorithmic steps to solve it and then implement these steps in C# programming language
- basic knowledge of programming in C#
- experience with developing and debugging software in Visual Studio
- knowledge about basic and advanced data structures
- ability to use version controls tools like GIT/SVN

Theoretical teaching topics:

1. Algorithms and the elements of programming languages. Program control flow.
2. Basic data types and data structures
3. Modular programming, Functions, recursion
4. Advanced data structures: Strings, Hash tables, Stack and Queue
5. Version control and Git

Practical exercises:

The number of practical work will be 70%. The practical exercises will be implemented in Visual Studio 2015.

Evaluation:

Students will be evaluated based on:

1. Homework (20%) – There will be 2 homework assignments.
2. Activity (30%) – will be assessed by professor during lectures and will be based on student activity, speed and skill shown during lecture assignments.
3. Final exam (50%)– will consist of ABCD test and practical test. Student will get assignments for practical test that he needs to solve by using a computer.

Student will pass the exam if he has more than 50% of points.

Course duration:

It will last for 40 school classes (36 for lectures and 4 for tests). Lectures will be completed in a block system in one-month time (10 days, 4 lectures per day)

Course content:

Day	Topic	Practical	Homework And Test
1	<ul style="list-style-type: none">•Introduction to Computer Science and IT.•Jobs in IT•Computer SW and HW architecture.•Introduction to algorithms.	<ul style="list-style-type: none">•Introduction to Visual Studio 2015•Cloud storage services and Dropbox.	
2	<ul style="list-style-type: none">•Introduction to C#.•Arithmetic, logic and relation operators.•Variables and data types.•Variable scope. Local and global variables.•Input and output statements. If statement.	<ul style="list-style-type: none">•Coding and debugging a computer program.	
3	<ul style="list-style-type: none">•Switch statement.•Algorithms with Loops.•Infinite Loop and break and continue statements.•Introduction to data structures and Arrays.	<ul style="list-style-type: none">•Assignment with For and while loops.	
4	<ul style="list-style-type: none">•Operations with arrays.•Different array sorting algorithms•Introduction to Multidimensional arrays.•Operations with Multidimensional arrays.	<ul style="list-style-type: none">•Assignment with Arrays and Lists.	HW 1
5	<ul style="list-style-type: none">•Introduction to Classes, objects and methods.•Introduction to standard library classes.•Data type conversion.	<ul style="list-style-type: none">•Type conversion.•Using Math and Date library classes.	
6	<ul style="list-style-type: none">•Introduction to modular programming.•Declaration and definition of functions.•Referent data types.	<ul style="list-style-type: none">•Assignments with functions.	

	•Passing arguments to functions.		
7	•Recursion. •Array as function parameter. •Introduction to String data type. •Methods of String class and operations with strings	•Strings Assignments.	HW2
8	•Introduction to generic classes. •Advanced data structures. •Hash tables, Linked list, Stack, Queue.	•Hash tables, Stack, Queue	
9	•Introduction to version control. •Introduction to Git.	•Git Assignment.	
10	•Summary of the course. •The analysis of the final test.		Final Test

Topic: Object-oriented programming

Lecturer: Suzana Stojković

Course objectives:

The course main objective is to teach students the basics of the object-oriented programming technique. Other goals are to teach and enable students to design object-oriented software. Student will also learn to develop Windows application in Visual Studio 2015. Prior knowledge is the procedural programming in some C-like language. Special attention will be given to practical work.

Course outcomes:

Mastering the object oriented programming and skills to independently design solutions to problems using the object-oriented concept and to learn how to develop applications in C# programs. Students will acquire:

- strong understanding of OOP concepts
- Good understanding of standard .NET libraries
- Further develop real life problem solving skills

Theoretical teaching topics:

- The concepts of object oriented programming. Classes and object. Constructors.
- Programming in C#.
- Static class members. Operator overloading. Inheritance. Polymorphism. Abstract classes and interfaces.
- Exception handling.

- Working with files and file-system in C#.
- Windows programming in C#.

Practical exercises:

The number of practical work will be 60%. The practical exercises will be implemented in Visual Studio 2015.

Evaluation:

Students will be evaluated based on:

4. Homework (20%) – There will be 2 homework assignments.
5. Activity (30%) – will be assessed by professor during lectures and will be based on student activity, speed and skill shown during lecture assignments.
6. Final exam (50%)– will consist of ABCD test and practical test. Student will get assignments for practical test that he needs to solve by using a computer.

Student will pass the exam if he has more than 50% of points.

Course duration:

It will last for 40 school classes (36 for lectures and 4 for tests). Lectures will be completed in a block system in one-month time (10 days, 4 lectures per day)

Course content:

Day	Topic	Practical	Homework and Testing
1	1. class: Procedural, modular and object-oriented programming techniques. Class definition, attributes and methodes. Class member access. Objects creation. 2 class: Properties in C#, reference this, static class members.	1. class: Develop object-oriented application in VS 2015. Adding class by using class wizard. Creating class diagram. 2. class: Adding some properties and static attributes and methods in the created application.	
2	1. class: Value and reference types. Structures and enumerations. 2 class: Pass parameters by value and by reference. Methods with variable number of parameters.	1, 2. class: C# application with many classes, enumerations and structures.	
3	1. class: Constructors. 2 class: Operator overloading. Indexers.	1, 2 class: C# application containing classes with constructors and operator methods.	

4	1. class: Inheritance. Definition of derived class. Constructors of derived classes. 2 class: Polymorphism. Virtual and override methods in C#.	1, 2 class: Application with base and derived classes.	
5	1. class: Abstract methods and abstract classes. 2 class: Interfaces in C#.	1, 2 class: Application containing abstract classes or interfaces.	HW 1
6	1. class: Exception handling: exception throwing, exception catching. Uncaught exceptions. 2 class: Generic methods and classes.	1, 2 class: Application with generic classes and exceptions.	
7	1. class: Delegates and events. 2 class: Event driven programming. Windows forms and windows controls	1, 2. class: Simple Windows applications. 2 class:	
8	1. class: New controls, their properties and events.	1, 2, 3 class: Windows application with many forms.	HW2
9	1. class: Input-output. Working with binary and textual files in C#. 2 class: Working with file-system in C#.	1, 2: Application containing reading data from the files, and writing data in the files.	
10	Summary of the course.		Final Test

Topic: Introduction to relational databases

Lecturer: Milan Zdravković

Course objectives: To achieve understanding of designing and manipulating relational data structures

Course outcomes: Main desired outcomes are:

- Ability of participants to independently design relational data structures of moderate complexity.
- Ability of participants to independently develop SQL queries for the web projects of moderate complexity

Theoretical teaching topics: Different techniques and approaches for information modeling. Relational Database Management Systems – main concepts, definition, architecture. Conceptual design of databases – Entity-Relationship (ER) modeling. Logical database design: datatypes, transformation of conceptual to logical database design, Implementation of database by using SQL language, Data constraints, Referential integrity. Physical database design: schema normalization and denormalization, SQL query optimization and indexing, application and security aspects of the database design (access control, data audit), different functions of the RDBMS (views, stored procedures, triggers, transactions). Data view and manipulation by using SQL: data inserting, data deletion, data updates, viewing data (sorting, viewing data from multiple tables, basic data processing, working with time and date functions, subqueries, aggregate queries). SQL query optimization – continuation.

Practical exercises: Exercising basic information modeling: for the given form, define concepts, their characteristics, their constraints and their relationships. Exercising hierarchical data modeling: for the given website, develop XML schema. Using draw.io service for drawing ER models. Exercising basic understanding of the ER concepts: for the number of given short and longer sentences, identify concepts and relationships and draw basic ER models. Exercising understanding of data and relation properties: for the given narrative text, design ER models. Exercising advanced ER concepts, namely, identifiers, complex and multi-valued attributes, weak entities, associative entities, cardinality and modality, relationships with optional and mandatory entity participation, identifying and non-identifying relationships, recursive relationships: for the given narrative text, design ER models. Exercising transformation of conceptual to logical database design: for the given conceptual ER model, design database schema, by using WWW SQL designer web service. MySQL software installation and customization. SQL exercises: for the given texts and given database designs, develop the set of INSERT, UPDATE and SELECT queries.

Evaluation: The overall grade is composed of the partial assessments of homework (20% of overall grade, total of 3 homework assignments), final exam (50% of overall grade, practical test with the narrative form of requirements which need to be transposed to a logical and physical database design + narrative functional requirements for the given database structure which need to be transposed to SQL queries) and participants engagement during class assignments (30% of overall grade, include trainers personal assessment of the participants skills).

Course duration: The course is 40 classes long (36 for lectures and practical work, 4 for a final exam). The 4 classes blocks will be implemented by day.

Course content:

Day	Topic	Practice	Homework and testing
1	<p>Class 1. Client-server architecture DIKTW pyramid: Data, Information, Knowledge, Wisdom Meta-data</p> <p>Class 2. Hierarchical models – XML Data model features, data features</p> <p>Class 3. Data modeling approaches and tools Relational databases, RDBMS</p>	<p>Define basic functional requirements for the given website idea</p> <p>Identify concepts, their features, constraints and relationships for the given examples</p> <p>Based on the identified functional</p>	

	Class 4. Process of database design	requirements, design XML schema Summary discussion	
2	Class 1. Conceptual, logical and physical design Key elements of the conceptual design Entity types, weak entities Class 2. Associative entities, category entities Attributes Class 3. ER notation Class 4. Cardinality	Using draw.io web service for drawing ER schemas Design simple ER schemas for the given short and longer sentences – identification of entities, entity types and relationships and their cardinalities Demonstration of conceptual design process on example Independent design of the conceptual model – 4 assignments	
3	Class 1. Advanced ER concepts: identifiers, complex and multi-valued attributes Class 2. Weak entities, associative entities, modality, relationship with optional and mandatory entity participation, identifying and non-identifying relationships, recursive relationships Class 3-4. Exercises	Identify modalities for the given short examples For the previously done assignments, implement advanced ER concepts above – 4 assignments Independent design of ER models of moderate complexity – 3 assignments	Homework – design conceptual model for 2 given narrative descriptions of websites
4	Class 1. Transformation of conceptual to logical design Data types (textual, numeric, date and time) Class 2-3. WWW SQL Designer online tool Class 4. Transformation of subtypes Creating constraints	Working with WWW SQL Designer online tool Demonstration – transformation of conceptual to logical design – reference example Saving, opening and generating SQL code in WWW SQL Designer MySQL software installation and customization	Homework – transform conceptual to logical model for two given models – 2 assignments

5	<p>Class 1. Introduction to SQL. Key principles</p> <p>Class 2. CREATE TABLE command</p> <p>Class 3. Application and security aspects of database design.</p> <p>Class 4. INSERT command</p>	<p>Creating databases with the code generated from WWW SQL Designer</p> <p>Inserting data in the created database – demonstration and assignment</p>	
6	<p>Class 1. Basic structure of SELECT, UPDATE and DELETE SQL queries</p> <p>Operators</p> <p>Class 2. Criteria for data manipulation, using joker characters</p> <p>Sorting</p> <p>Class 3. Joining data (INNER, LEFT, RIGHT, FULL)</p> <p>Using aliases</p> <p>DISTINCT clause</p> <p>Class 4. Basic data processing</p>	<p>Assignments – design of SELECT, UPDATE and DELETE queries based on the given narrative – functional requirements</p> <p>Assignments – joining data and basic data processing</p>	<p>Homework – for the given sentences (functional requirements) design SELECT, UPDATE and DELETE queries</p>
7	<p>Class 1. Basic structure of aggregate queries</p> <p>Aggregate functions: MIN, MAX, SUM, AVG, COUNT</p> <p>Class 2-3. GROUP BY, HAVING clauses</p> <p>Class 4. SQL date and time functions</p>	<p>Demonstration of examples of aggregate queries</p> <p>Assignments – design of aggregate queries</p> <p>Demonstration of examples of using date and time functions</p>	
8	<p>Class 1-2. Functions of RDBM systems</p> <p>Views, Stored procedures, Triggers, Transactions</p> <p>Class 3. Indexing</p> <p>Class 4. Schema normalization</p>	<p>Demonstration of RDBMS functions</p> <p>Discussion on schema normalization approaches</p>	
9	<p>Class 1-2. Pre-exam</p> <p>Class 3-4. Discussion and self-evaluation</p>	<p>Participants are given the example of exam. They work on the solutions for the given problems. After the completion, teacher presents the solution, summary discussion takes place, based on which participants are self-evaluated.</p>	<p>Self-evaluation</p>
10	<p>Class 1-4. Final exam</p>		<p>Final exam</p>

Topic: Web programming

Lecturer: Željko Kalezić

Course objectives:

The course main objective is to teach students the basics of web software development. Other goals are to teach and enable students to think algorithmically, to enable them to analyze and breakdown real-life problem into algorithmic steps. The course will introduce advanced server side programming techniques in ASP.NET, which is open-source server-side web application framework designed for web development to produce dynamic web pages. The participants will gain the ability to independently develop web application in ASP.NET – MVC for web pages and ASP.NET - WebAPI for service oriented applications. They will learn to write, compile, build and debug web applications in Visual Studio 2017. Prior knowledge is not needed. Special attention will be given to practical work, coding, and teamwork through team projects.

Course outcomes:

The participants will acquire:

- Basic understanding of HTML/CSS
- Experience with server-side .NET development
- Experience with .NET, C#, ASP .NET MVC, ASP .NET Web API, JSON, REST
- Experience with securing Web applications (authorization and authentication)
- Knowledge of web-application architecture (multi-module, multi-layered systems)
- Experience with OR Mapping - Entity Framework (Code First, Database First)
- Basic Javascript understanding

Theoretical teaching topics:

1. Architecture of a Web applications
2. Server-Side Programming Concepts
3. ASP.NET and IIS Overview
4. Working with databases
5. Model-View-Controller pattern
6. Web Services and Ajax
7. Security

Practical exercises:

40% to 60% of working hours. Introduction to Visual Studio development environment and Web projects. The implementation and publishing of web applications.

Evaluation:

Students will be evaluated based on:

1. Homework (20%) – There will be 4 homework assignments.

2. Activity (30%) – will be assessed by professor during lectures and will be based on student activity, speed and skill shown during lecture assignments.
3. Final exam (50%) – will consist of ABCD test and team project. Students will get assignment for practical team project (eg. to build a web shop) that they need to solve by using a computer in teams and they also need to present the solution.

Student will pass the exam if he has more than 50% of points.

Course duration:

It will last for 108 school classes (104 for lectures and 4 for tests). Lectures will be completed in a block system in two-months time (27 days, 4 lectures per day)

Course content:

Day	Topic	Practical	Homework and Testing
1	Explaining course content and structure Web application architecture fundamentals Client server architecture Three tier architecture MVC pattern Object Relational Mapper Entity Framework	Create default ASP.NET MVC application	
2	HTTP Protocol fundamentals Development Tools (Visual Studio 2017) Version control systems Git Flow repository organization methodology Cloud Fundamentals Windows Azure	Create Github repository Commit code to Github Deploy code to Azure platform	
3	Object Relational Mapper Fundamentals Entity Framework Introduction DB Context Entity Lifecycle	Create ORM Model, create migration and add the model to DB Context	
4	Basic operations with DB data through ORM (Create, Read, Update, Delete - CRUD)	Create CRUD controllers using scaffolding and execute basic operations	
5	Database relations through ORM (1:1, 1:N, M:N) Validation of model data Model inheritance and polymorphism	Add relations using ORM Add validation to the models	
6	Request processing in MVC .NET Routing in MVC .NET Routing engine fundamentals Route configuration:	Adding routes through attributes and using convention Adding custom routes with	

	<ul style="list-style-type: none"> - Attribute based - Convention based Route registration Route defaults, priorities and constraints	constraints	
7	Controllers in MVC .NET IController interface example Controller base class Action Methods Action Results Filters Request object HttpContext object User object Passing parameters to controller (query string, route attributes, model binder)	Passing parameters through query string, route and using model binder	
8	Returning data from controllers Action Results Examples (Content, Json, File, View) ASP .NET Views Fundamentals Intro to HTML Intro to Razor syntax	Test Content, JSON and File ActionResults	
9	Passing data to views ViewBag and ViewData objects View Models Strongly and weakly typed views	Create two views, one strongly typed and one weakly typed. Pass data to the views using Models and ViewBag	
10	Advanced ASP .NET View topics Layout views Partial views View Composition Adding a Bootstrap theme Resource bundling Sections Javascript basics	Add Bootstrap theme Move navigation bar to partial view Add section and add some basic JS code to it	HW1
11	Creating Forms from scratch HTML Helpers introduction Strongly typed HTML helpers Advanced validation Working with models and relations in forms	Create a form that has ability to change multiple objects	
12	Working with data through Entity Framework LINQ Introduction Package Manager Introduction Creating datagrid with search, sorting and	Create a datagrid with sorting, filtering and pagination	

	pagination capabilities		
13	ASP .NET Filter Introduction and Fundamentals Authorization Filters Action Filters Response Filters Exception Filters Filter registration Filter execution priority IIS & SQL Express introduction Connection strings	Add authorization filters to desired action on a controller Deploy website on IIS & SQL Express	
14	Web application security fundamentals ASP .NET Identity Introduction Authentication Authorization based on Roles ASP .NET Identity Configuration (Confirmation email, password parameters, ...) Creating System Roles Adding additional user information to identity model Web.config fundamentals	Analyze ASP .NET Identity DB structure Create account and login to the system, explore manage account page ASP .NET Identity configuration: setup confirmation email and password parameters Add additional user information (eg. birthday) Add default system roles to Seed method and add authorization based on Roles to a desired controller action	HW2
15	Social network login fundamentals OAUTH Protocol Introduction Adding Facebook login to web application	Create Facebook application Setup ASP .NET Identity login through Facebook	
16	REST Fundamentals Creating, using and testing Web API REST services	- Creating RESTful web service using Web API	
17	jQuery Datatables - HTML and Ajax datasource Usage of OData and SOAP/WSDL services	Consuming webservice using jQuery DataTables plugin Using OData and WSDL/SOAP web services using service references	
18	Team Project Workshops	Implementing defined DB	HW3

- 25	- Defining team project themes - Defining functionalities to be implemented - Defining DB (ORM Model) structure - Defining authorization scheme (Roles) - Defining controllers and supporting views - Styling the project with Bootstrap themes	(ORM Model) structure Implementing Role based authorization Implementing controllers and supporting views Implementing project themes	HW4
26	Advanced topics: MVC Core and Websocket (SignalR) Introduction	/	
27	Final exam and team project presentation Future career advices from industry professional	/	Final exam Team project presentation

Topic: Methodologies and tools for software development

Lecturer: Miloš Kosanović

Course objectives:

The course will introduce the concepts and methodologies of the agile software development, life-cycle, development processes and Project Management frameworks.

Course outcomes:

The participants will understand all processes and activities in software development and will be able to participate in all phases of software project cycle.

Theoretical teaching topics:

1. Methodologies overview (frameworks and approaches). Software development life cycle and software development process. User interface design.
2. Requirements Engineering. Agile methodology (SCRUM).
3. Task planning. Task estimation and implementation.
4. Architecture design. UML language. Testing and documenting.
5. Deployment and Maintenance. Project Management frameworks.

Practical exercises:

20% to 50% of working hours. The practical exercises will include scrum meeting simulation, sprint planning meeting simulation, use case analysis and discussion, software tools for fast

wireframe and UI development, overview and demonstration of tools for project management and issue tracking.

Course content: Lecturer will need to complete a course from the selected field with its own course curriculum.

Course duration: The course will last for **24 school classes (20 for lectures and 4 for tests)**. Lectures will be completed in a block system (6 days with 4 classes per day).

Curriculum

The first day should be the general introduction into the course topics and the software development cycle in general. The content and the learning pace will depend on the candidate's previous knowledge and experience. The remaining 5 days will follow the rule: 2 school classes of lecture, 2 school classes of practical work. The goal of the course will be to follow the development of one real life project example from its inception to the first production release of the product. It will be requested from the candidates to create a team for the project, analyze the requirements, specify high level requirements, create the tasks or user stories, describe the testing and quality procedures and describe their solution to the real life problems.

Grading

1. Homework assignments will be graded with 20%
2. Lecturer will grade each student based on general impression with 30%.
3. Final ABCD test and project (which will contain requirements specification document) will be graded with 50%

The student will pass each test if he has correctly answered more than 50% of the questions. The student will pass the course if he has more than 50% of points.

Tests

The test questions and the topic of the project assignment will not be disclosed due to confidentiality reasons.

Course content:

Day	Topic	Practical	Homework And Test
1	<ul style="list-style-type: none">•Introduction, Course overview, Methodologies overview (framework and approaches)•SDLC - Software Development Life Cycle.	<ul style="list-style-type: none">•Create wireframes (use Balsamiq)	HW 1

	<ul style="list-style-type: none"> •Waterfall method, V model •Incremental model, Iterative model (RUP, EUP), Prototype model, Spiral model •UI Design - Wire-Frames and Mock-ups 		
2	<ul style="list-style-type: none"> •Requirements Analysis & Specification. Functional and non-functional requirements •User stories •Introduction to project assignment •Agile methodology (SCRUM, Kanban, XP), Comparison between standard and agile approach. •Team roles in agile development •Test Driven Development TDD, Behaviour Driven Development BDD •Continuous Integration •Agile software development process. 	<ul style="list-style-type: none"> •Test project explanation and proposal •SCRUM meeting simulation, • Case study I – Requirements meeting 	
3	<ul style="list-style-type: none"> •SCRUM •KANBAN •Planning and estimating tasks •The Vision or Long term planning, Scope change. Time constraints and other problems. 	<ul style="list-style-type: none"> •WBS – Work breakdown structure •Sprint planning meeting simulation 	HW 2
4	<ul style="list-style-type: none"> •Technical design (specification). UML diagrams. Use Case, Activity and class diagram, Sequence diagrams, •System design, DataBase design •Coding - good coding practice, •Software Testing and Quality Assurance. •Software Maintenance and end user support 	<ul style="list-style-type: none"> •UML diagrams (starUML, glyphy)) •Questions and discussion •Writing unit tests. 	
5	<ul style="list-style-type: none"> •Project Management •Project Management frameworks overview (JIRA, redmine, microsoft project, gunter) •Other tools for project management, development and issue tracking, Reporting bugs 	<ul style="list-style-type: none"> •Introduction to some Project management tool •Write tasks in some project management tool 	
6	<ul style="list-style-type: none"> •Coding - good coding practice. •Software Maintenance and end user support •Documenting projects 		Final Test