Óbuda University John von Neumann Faculty of Informatics	Institute of Software Engineering

Name and code: Advanced programming techniques (NSXHF1EBNE) **Credits:** 4

Computer Science BSc *Daytime 2020/21 year II. semester*

Subject lecturers: Albert Áron, Balázs Elemér, Benkő Gábor, Dr. Erdélyi Krisztina, Haydu Lénárt, Dr. Kertész Gábor, Nagy Dávid, Romhányi Ármin, Röhberg Péter, Simon-Nagy Gabriella, Sipos Miklós, Szabó-Resch Zsolt, Tóth Norbert

Prerequisites: (with code)				
Weekly hours:	Lecture: 0	Seminar: 0	Lab. hours: 3	Consultation: 0
Way of assessment:	Midyear grade			

Course description

Goal: Familiarize the students with the advanced topics of C# programming. One lesson from the weekly three is held as a lecture.

Course description: Advanced techniques of the C# language (Lambda expressions, LINQ, Entity Framework, Attributes, Reflection, DLL, Unit tests, Mock, Processes and threads);

Lecture schedule

Education week	Topic
1	*Lecture*: Rules, Delegate/event, Func/Action, Lambda, Logger example *CSharp*: Simple FeedbackProcessor
2	*Lecture*: Extension methods, LINQ, XML/JSON, XLINQ/JsonLINQ, People.xml *CSharp*: XLINQ War on westeros.xml
3	*Lecture*: Dotnet execution/DLL, Reflection, XmlSerializer *CSharp*: Reflection / Validator
4	*Lecture*: Layering, ORM, SQL-first vs Code-first, EF EMP demo (sql first) *CSharp*: DbContext, Entities, Db Seed, Select/NonCrud example (code first) *Project*: Bitbucket/atlassian account, git repository, readme.md
5	*Lecture*: Project expectations, Layering example, Repository pattern *CSharp*: Project example: One SLN, Multiple CSPROJ, DB access, Menu
6	*Lecture*: GIT quickstart + DLL (chaser) via GIT *CSharp*: ZH practice *Project*: Solution with empty projects
7	*Lecture*: Dotnet GC, Dotnet versions *CSharp*: ZH (together, outside regular schedule) *Project*: Database + dbcontext + entities
8	*Lecture*: Unit tests, theory, simple example *CSharp*: Unit tests using nUnit *Project*: Menu + All list operations
9	*Lecture*: Unit tests using MOQ *CSharp*: Project review
10	*Lecture*: Processes, IPC (theory + examples) *CSharp*: Project review *Project*: All crud+noncrud functionality
11	*Lecture*: Threads 1 (Thread+Task syntax and examples) *CSharp*: Thread/Task usage
12	*Lecture*: Threads 2 (Task extras, synchronization) *CSharp*: Thread synchronization
13	*Lecture*: Layered architectures closing thoughts *CSharp*: Project work consultation *Project*: Gitstat deadline
14	*CSharp*: ZH retakes (together, outside regular schedule) *Project*: Project work defense

Midterm requirements

Attendance on the practices is obligatory. Before the practices the students must watch the lecture videos.

The students will write one mid-semester test, writing the mid-semester test is obligatory. If a student doesn't write or doesn't pass the mid-semester test, a re-test is possible on the last week (the re-test can contain materials from the full semester). Successful test means achieving minimum 50% of the points.

If a student doesn't have a passed test at the end of the regular semester, a last re-test is possible in the exam season.

The students have to create a project work on their own, that shows their advanced C# programming skills:

- The expectations must be met that are mentioned in the prog3_layers_requirements document:
- Gitstats: True
- Stylecop/NetAnalyzer-valid code, Doxygen developer documentation
- Single-user, single-branch GIT repository
- Usage of a database + Entity Framework to access it
- Usage of LINQ
- Layered architecture (minimum 5 sub-projects: Console app + business logic DLL + repository DLL + data access DLL + Unit test DLL)
- Unit tests (typically for the logic classes)

The project work has to be submitted until the deadline (the project schedule can found in the requirements pdf). If that deadline is not met, or the teacher doesn't accept the quality of the project work or it doesn't fulfil the bare minimum expectations listed above, then the student will have to present their project work in the exam season.

Midterm Test Scheduling

Education week	Topic
7	FIRST MIDTERM
14	RE-TEST if necessary

Midterm grade calculation methods

Mid-semester grade can only be given to a student who passed the test and who submitted an accepted project work. The grade will be the test grade, that can be changed with one grade by the project work. In case of the second, third, and fourth milestone: every single missed Project Work milestone deadline decreases the maximum possible end-semester grade that the student can receive.

“Signature refused” entry will be given to any student who misses more than 30% of the practice sessions. (TVSZ 23.§). Also “Signature refused” entry will be given to any student who did not submit a project work (missing GIT repository).

“Failed” grade will be given to any student who doesn't have the successful practice test or the project grade is “Fail”. These students can participate in the signature retake.

Method of replacement

In a mid-semester re-test, the test can be rewritten.

In the exam season re-test, ALL students have to solve a combined task from the FULL contents of the semester.

Presentation of the project works will be done afterwards, if necessary.

Type of exam

Exam grade calculation methods

References

Obligatory:

Lab presentations, practice materials
All materials listed in the course page inside the University e-learning system
https://users.nik.uni-obuda.hu/prog3/
prog_tools_en.pdf, prog3_layers_requirements_en.pdf

Recommended:

Others: