

Introduction to the course

Sergey Efremov
Associate professor

Programming II

School of Business Informatics
Spring 2018

Boarding complete



During the lectures - English only!

Course topics

3rd module:

- Object-oriented programming in detail
- Dynamic collections and generic types
- File input-output and serialization
- Graphical user interface with WPF and XAML
- Libraries and packages. Version control systems
- Querying data with LINQ

4th module:

- Abstract programming
- Interfacing databases through ORMs
- Parallel programming
- Network interconnection

Self-study:

- XAML, Windows 10 and UWP

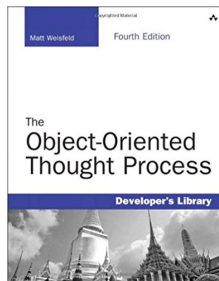
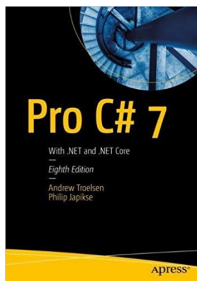
LMS: canvas.instructure.com

- Registration manual:
goo.gl/RAB1mb
- Mobile applications are available for Android and iOS

Software toolset: Visual Studio 2017 Community Edition



- A.Troelsen, P.Japikse. Pro C# 7 with .NET and .NET Core. Eighth edition (or any of the previous 2 editions)
- M.Weisfeld. The Object-Oriented Thought Process. Third or Fourth edition.



Assignments

Type	Modules		Notes
	3	4	
Homework	2	2	Coding problems, focused on a specific topic
Test		1	Theory only
Programming project		1	Working in teams (last month of the course)
Exam		1	One or two coding problems

Checking assignments

- Assignments go through an anonymous checking process among 10 people including course instructors and teaching assistants.
- Each assignment will have to be defended in class. If a student does not show any understanding of his/her own code, the grade is cancelled.

Late policy

Each assignment has a deadline. If submitted after the deadline, the grade is reduced with the following penalties:

Delay	Downgrade
less than 1 week	20%
1-2 weeks	30%
2-3 weeks	40%
more than 3 weeks but less than 1 month	50%

An assignment can be submitted **only once**

Anti-plagiarism

- All submitted solutions (HW assignments, projects) will be compared to each other for plagiarism by an automated service.
- If a group of students is caught cheating, the corresponding assignment will be counted towards their aggregated grade with a value of 0.
- See clarification in the course syllabus



Grading policy

```
1 // Aggregated grade for 3rd and 4th modules
2 static int GetAggregated2018(int[] hw, int test, int project)
3 {
4     int sum = 0;
5     for (int i = 0; i < hw.Length; i++)
6         sum += hw[i];
7     double avg = (double)sum / hw.Length;
8     return (int)Math.Round(0.4 * avg + 0.3 * test + 0.3 *
9         project);
10 }
11 // Final course grade
12 static int GetFinalGrade(int aggr2017, int aggr2018, int exam)
13 {
14     if (aggr2018 >= 9)
15         return aggr2018;
16     return (int)Math.Round(0.15 * aggr2017 + 0.5 * aggr2018 +
17         0.35 * exam);
18 }
```

Before the lecture:

- 1 Read the supplied handout.
- 2 If a handout contains sample code, study it carefully, run on your computer.

During the lecture:

- 1 Answer questions from a quiz when instructed by the lecturer. The quiz goes online on Canvas LMS when the lecture begins.
- 2 Follow live code demos on your laptop.

Additionally:

- 1 Post and discuss questions on the LMS forum