

Programming in Graphical Environment — Rules and Regulations

mgr inż. Paweł Aszklar
P.Aszklar@mini.pw.edu.pl

Warsaw, February 22, 2021

1 Prerequisite subjects

Programming 1 - Fundamentals, Programming 2 - object oriented, Programming 3 - advanced, Algorithms and Data Structures

2 Class Types

The subject consists of following class types:

- Lectures — 15 classes
- Laboratories:
 - Graded Laboratories — 10 classes
 - Tutorial Laboratories — 3 classes

Presence is mandatory only during graded laboratories.

3 Assessment Rules

- There will be 9 graded laboratories in the semester and one retake
- Graded laboratory 5 is scored for up to 4 points, all others for 12 points each.
- During each graded laboratory students will be given an "in class" task for which they will have 90 minutes to solve. Some tasks will be followed by an "at home" part.
- "In class" tasks will be graded during the laboratory, "at home" part must be submitted for grading during the following class.

- Before solving the "at home" part of a task, the "in class" portion must be completed, even if it was not finished during the laboratory.
- Each task will be graded during class based on student's understanding of the problem, progress towards writing a solution.
- Reasonable code quality regarding naming conventions, consistent code formatting, function length, code repetition avoidance etc. is expected. Not meeting those requirements will result in a lowered score.
- All solutions must be delivered to the laboratory teacher. Failure to do so will result in 0 points for a given task.
- Solutions will be compared using anti-plagiarism software. If significant similarities are detected between solutions, their authors will receive 0 points for the task. Repeating offenders will fail the subject.
- During the class students are allowed to use lecture slides, own notes and previous tasks solutions, documentation and references for various technologies available on the internet.
- Laboratory tasks worth 12 points each will be split into two parts:
 - Part A — Graded Labs 1-4
 - Part B — Graded Labs 6-9
- The result of the retake will replace a single score from one of the labs in Part A and B, chosen by the student.
- Points lost during one absence can be made up during the retake.
- In case of an excused absence (e.g. illness, provided a sick note from a doctor is presented) laboratory teacher can:
 - for tasks containing "at home" part, allow the student to send the solution of the whole task via e-mail on the due date;
 - for extended absence, establish individual assessment rules.
- To pass the subject students must score at least 24 points from each of Part A and B and at least 50 points from all laboratories.
- Final grade will depend on the sum total of points:

Points	Final Grade
0 – 50	2.0 (D - Fail)
51 – 60	3.0 (C - Satisfactory)
61 – 70	3.5 (C+ - Fairly Good)
71 – 80	4.0 (B - Good)
81 – 90	4.5 (B+ - Above Good)
91 – 100	5.0 (A - Very Good)

4 Course Content

Students will get familiar with basic concepts designing and implementing User Interfaces in Graphical Environments. Course focuses on programming Graphical User Interfaces on Windows Operating System and will cover creating GUI applications in native Windows API, and in .NET Framework using Windows Forms and WPF along with discussion about similarities, advantages and disadvantages between them.

In particular the course covers the following topics:

- GUI design guidelines
- Windows API:
 - Windows, messages, data structures, error handling
 - Input handling (Mouse, Keyboard, etc.)
 - GDI graphics
 - Resources, controls and dialog boxes.
- Windows Forms:
 - Application structure and its settings
 - Forms, properties, events
 - Built-in and custom controls
 - Resources and localization
- WPF:
 - XAML language, applications, windows, pages
 - Dependency properties, routed events, logical and visual trees
 - Layout, panels, content and items controls, commands
 - 2D and 3D visuals, media elements
 - Styles, templates, data binding
 - Resources, MVVM pattern
- Introduction to HTML and CSS
- Internationalisation (globalization, localization), Accessibility
- Unicode
- UI testing

5 Learning Outcomes

Students passing the subject will:

- have general knowledge of typical approaches to creating windows-based graphical user interfaces for human-machine communication,
- know basic methods, approaches and tools for solving simple programming tasks concerned with building software for Microsoft Windows operating system,
- have systematized knowledge of operating systems and software architecture,
- know basic methods, approaches and tools employed for solving simple programming tasks making use of Microsoft Windows operating system functionality,
- be able to make use of the Microsoft Windows operating system APIs,
- be able to solve simple human-machine communication problems and to design simple software systems,
- be able to choose appropriate tools, design and implement simple software system given fixed specification.