

Course Syllabus**I. General Information**

Course name	Laboratory of programming: applications in basic programming environments
Programme	Informatics
Level of studies (BA, BSc, MA, MSc, long-cycle MA)	BA
Form of studies (full-time, part-time)	Full-time
Discipline	Informatics
Language of instruction	english

Course coordinator	mgr Krzysztof Buszowski
--------------------	-------------------------

Type of class (<i>use only the types mentioned below</i>)	Number of teaching hours	Semester	ECTS Points
lecture			3+3
tutorial			
classes			
laboratory classes	30+30	V+VI	
workshops			
seminar			
introductory seminar			
foreign language classes			
practical placement			
field work			
diploma laboratory			
translation classes			
study visit			

Course pre-requisites	Basic knowledge in Object-oriented programming and Data Bases. Basic knowledge in mathematics, physics and computer science.
-----------------------	---

II. Course Objectives

Getting used to distributed version control system (GIT), continuous integration (CI) and tools dedicated to project management (technical)
Getting to know the characteristics and features of various frameworks and libraries used in development of desktop application (GUI application).
Getting to know features of various frameworks used in development of server application (backend).
Examples of good practises related to software development.

III. Course learning outcomes with reference to programme learning outcomes

Symbol	Description of course learning outcome	Reference to programme learning outcome
KNOWLEDGE		
W_01	The student can use distributed version control system and other tools which allows to take control over code development and release process	K_W08
W_02	The student can use various frameworks and libraries in order to implement application	K_W06, K_W08
SKILLS		
U_01	The student can implement software that consist of two separate applications that integrate one to each other (desktop and server application)	K_U08, K_U17
SOCIAL COMPETENCIES		
K_01	The student is aware of constant need to improve knowledge about used tools	K_K06

IV. Course Content

<ol style="list-style-type: none"> 1. Introduction. Overview of the technologies that will be used during lectures. Working with GIT. 2. Introduction to Java ecosystem. 3. Application dependency & build management using Gradle. Build lifecycle including tests (unit testing and integration testing). 4. Graphical user interface using JavaFX. Introduction to design patterns used in development of such applications. 5. Maintaining code reliability by extensive unit testing. 4. Persistence of application state within database. 6. Introduction into HTTP/2 protocol. 7. Introduction into REST API. Good practises for API design. Security and potential vulnerabilities. 8. Introduction to Spring framework and Spring MVC. Using tools to quickly implement HTTP server. Extensive integration testing.

V. Didactic methods used and forms of assessment of learning outcomes

Symbol	Didactic methods <i>(choose from the list)</i>	Forms of assessment <i>(choose from the list)</i>	Documentation type <i>(choose from the list)</i>
KNOWLEDGE			
W_01	Practical classes, Guided practice	Preparation / implementation of the project	Project rating card
W_02	Practical classes, Guided practice	Preparation / implementation of the project	Project rating card
SKILLS			
U_01	Practical classes design thinking	Preparation / implementation	Project rating card

		of the project	
SOCIAL COMPETENCIES			
K_01	Discussion, PBL (Problem-Based Learning) design thinking	Preparation / implementation of the project	Project rating card

VI. Grading criteria, weighting factors.....

Final project using specific technology.

VII. Student workload

Form of activity	Number of hours
Number of contact hours (with the teacher)	100
Number of hours of individual student work	60

VIII. Literature

Basic literature
Benjamin J. Evans, David Flanagan, "Java in a Nutshell", O'Reilly 2014 Documentation of used tools Documents prepared by lecturer
Additional literature
Benjamin J. Evans, Martijn Verburg, "The Well-Grounded Java Developer", Manning 2012