



## COURSE DESCRIPTION CARD - SYLLABUS

Course name

Technologie programistyczne

### Course

Field of study

computer science

Area of study (specialization)

Intelligent information technologies

Level of study

Second-cycle studies

Form of study

full-time

Year/Semester

1/1

Profile of study

general academic

Course offered in

Polish

Requirements

compulsory

### Number of hours

Lecture

30

Laboratory classes

Other (e.g. online)

Tutorials

Projects/seminars

30

### Number of credit points

4

### Lecturers

Responsible for the course/lecturer:

Szymon Wilk, PhD. hab.

Responsible for the course/lecturer:

Bartłomiej Prędko, PhD.

### Prerequisites

Student attending this course should have basic knowledge of programming and popular programming technologies and languages (e.g. Java) that is being taught in earlier courses.

Student should be able to design and implement a simple software system, acquire knowledge from different sources. Student should be aware of need to broaden his horizons and be able to work in team.

Besides, student should have basic social competence like honesty, responsibility, persistence, curiosity and creativity, respect for others.

### Course objective

1. Students should obtain basic computer science knowledge concerning new programming technologies used to build advanced computer systems (e.g. mobile systems, systems using machine learning) and design and implementation of such systems.

2. Students should develop their ability of solving problems and choosing suitable technologies for designed problems, documenting their choices and present their ideas.



3. Students should enhance their team-work skills associated with implementation of complex computer systems.

### Course-related learning outcomes

#### Knowledge

1. Student has enhanced knowledge of programming technologies, tools and development environments used to implement software systems (especially mobile and intelligent).
2. Student is aware of current trends in computing and of newest advances in programming technologies.

#### Skills

1. Student is able to acquire information considering programming technologies from many sources (publications, Internet), integrate and interpret this information.
2. Student is able to choose suitable technology for the task of software system implementation. He/She is aware of its limitations.
3. Student is able, according to provided specification, to design and implement a software system, using suitable technologies and tools, including the adaptation of existing tools.
4. Student can prepare and present a presentation in Polish or English language, showing his/her ideas and approaches to design and implementation process.

#### Social competences

1. Student knows, that skills and knowledge can quickly become obsolete.
2. Student understands the importance of acquiring and adapting new technologies to practical problems.

### Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Presented outcomes are verified as follows:

- 45-minute test on the last lecture, consisting of 10-15 questions (open and choice questions with different point value). Student has to score at least 50% points to pass the test. Students will be provided with test subjects using computer mail system.
- presentation during lecture. Presentations should be about novel programming technologies and tools. Presentation can be done by a group of 2 students or by a single person. Presentation should last about 30 minutes (1 person) to 60 minutes (2 persons). Presentation will be scored in scale from 2 to 5. Positive score entitles student(s) to pass on the aforementioned test. Presentations are voluntary and student should agree to them in the first 3 weeks of the semester.

Skills obtained during project classes are verified as following:



- scores for the presentations of advancement of implemented project including the final presentation. Each project group is obliged to make 3 presentations in the semester,

- score for the final paper documenting designed project,

The final score is based on above partial scores.

### Programme content

Lecture agenda consist of the following items (because we try to always present the most novel technologies, below list should be considered as a subject to change):

1. History of mobile platforms,
2. Tools and technologies for mobile platform development, including iOS and Android,
3. Tools and libraries used to data analysis and decision model construction using machine learning, e.g. TensorFlow, Microsoft Cognitive Services,
4. Tools and libraries used to solve optimization and constraint satisfaction problems, e.g. MiniZinc, ECLiPSe,
5. Tools and libraries for text document processing, e.g. Apache Lucene, Elasticsearch, Spacy,
6. Tools and libraries used to construction of advanced data processing systems, including unstructured data, e.g. Apache UIMA.

Project classes are of seminar type. During those students in teams design and implement chosen (by themselves) projects. The goal of these projects is to design and implement an advanced software systems using tools presented in lectures. Subject of those projects should broadly concern decision making and data processing.

Project classes agenda consists of:

1. selection of project and analysis of problem domain,
2. collecting project requirements, choosing the suitable technologies and planning a work schedule,
3. designing the system, implementing it and making the tests and experiments,
4. doing the three presentations of the project and preparing the final paper,
5. presentation and demonstration of the finished system.

### Teaching methods

1. Lecture: multimedia presentation, discussion, demonstration.
2. Project classes: implementation of the chosen sample projects in 2 person teams, multimedia presentation, demonstration.



## Bibliography

### Basic

1. D. Conway, J. Myles White: *Uczenie maszynowe dla programistów*, Helion, 2015.
2. M.A. Lasoff: *Podstawy języka Swift : programowanie aplikacji dla platformy iOS*, Helion, 2016.
3. M. Galloway: *Effective Objective-C 2.0 : 52 specific ways to improve your iOS and OS X programs*, Addison-Wesley, 2013.

### Additional

1. A. Niederliński: *Programowanie w logice z ograniczeniami. Łagodne wprowadzenie dla platformy ECLiPSe*. <http://www.pwlzo.pl/>
2. Documentation for the chosen tools and libraries.

## Breakdown of average student's workload

	Hours	ECTS
Total workload	110	4,0
Classes requiring direct contact with the teacher	60	2,0
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) <sup>1</sup>	50	2,0

<sup>1</sup> delete or add other activities as appropriate