



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Mobile Application Development with AI Elements

Course

Field of study

Artificial Intelligence

Area of study (specialization)

Level of study

Second-cycle studies

Form of study

full-time

Year/Semester

1/2

Profile of study

general academic

Course offered in

English

Requirements

compulsory

Number of hours

Lecture

30

Tutorials

Laboratory classes

15

Projects/seminars

15

Other (e.g. online)

Number of credit points

4

Lecturers

Responsible for the course/lecturer:

Bartłomiej Prędko, PhD

Bartlomiej.Predki@cs.put.poznan.pl

faculty: Computing and Telecommunications

address: Piotrowo 2, 60-965 Poznań

Responsible for the course/lecturer:

Prerequisites

Student should have knowledge concerning the way the computer works, imperative programming (obtained in earlier courses), the basics of computer networks and artificial intelligence. Should be able to solve basic problems in computing, especially in user interface design and application of specific algorithms. Student should understand the need to expand his competence and be ready to partake in group activities.

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



Course objective

1. Students should obtain knowledge concerning the history of mobile computer systems.
2. Students should know the difference between mobile and desktop/web applications design.
3. Students should be able to design and programme a mobile application for iOS and Android systems.
4. Students should have knowledge and experience in implementing mobile applications using client/server architecture.
5. Students should enhance their ability to work in teams.

Course-related learning outcomes

Knowledge

1. Student has a structured and well grounded knowledge of mobile systems and mobile application development.
2. Student has knowledge of current developments and trends in mobile systems.
3. Student knows basic techniques, methods and tools used to solve problems associated with mobile application design.
4. Student knows how to include Machine Learning approaches into mobile applications.
5. Student has a structured knowledge of computer architectures and operating systems.

Skills

1. Student can search for information concerning mobile systems and applications in literature, data bases and other sources, integrate it and formulate opinion.
2. Student is able to use information-communication techniques while solving problems in system design, especially in mobile systems.
3. Student is able to choose and apply adequate methods considering mobile application development.
4. Student can design a mobile application, choose an appropriate programming language and environment.
5. Student can formulate algorithms and implement them using one of the dominating mobile application development languages (Swift and Kotlin).
6. Student can correctly use the chosen method of estimating the labor consumption of software development, with the emphasis on the artificial intelligence methods
7. Student can plan his/her own development and can see need for constant discovery of new knowledge.

Social competences

1. Student knows, that skills and knowledge can quickly become obsolete.



2. Student is aware of knowledge importance in solving of engineering problems and knows the dangers of bad design and computer system malfunctions.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Presented outcomes are verified as follows:

Formative assessment:

- a) on lectures - based on the answers concerning material presented on previous lectures;
- b) on laboratories and projects - constant verification in classes - verification of knowledge and skill acquisition,

Summative assessment:

Concerning laboratories - verification of skills used in laboratory excersises and obtained on lectures,

Concerning projects - based on the implementation of assigned projects.

Concerning lectures - written test consisting of 10-15 questions; to pass the test student has to obtain at least 50% of correct answers.

Additional points obtained in classes, especially:

- demonstration of interesting extraculicual competences,
- presentation of additional problem aspects,
- efficacy of obtained knowledge use while solving a problem,
- ability to work in team,
- useful remarks concerning teaching materials.

Programme content

Following subjects are presented on lectures:

- introduction to Swift language,
- programming for iOS and iPadOS using Xcode IDE,
- basics of Kotlin language,
- programming for Android using Android Studio
- programming using different API's,
- using Cloud services,



- using ML models in mobile applications,
- data exchange protocols, e.g. JSON, REST.

In laboratories student are trying to solve in practice tasks presented in lectures as a series of mini projects; as a project students will implement a sample closed project for the chosen platform.

Teaching methods

1. Lecture: multimedia presentation, discussion, demonstration.
2. Laboratories: doing tasks, team work, design and implementation of sample problems.

Bibliography

Basic

1. iOS 5: programowanie: receptury / Vandad Nahavandipoor ; [tł.: Robert Górczyński], Helion 2013.
2. Tworzenie aplikacji na platformę iOS 5 : z wykorzystaniem Xcode, Interface Builder, Instruments, GDB oraz innych kluczowych narzędzi, Brandon Alexander, J. Bradford Dillon, Kevin Y. Kim, Helion, 2012
3. Objective-C : praktyczny podręcznik tworzenia aplikacji na systemy iOS i Mac OS X!, Stephen G. Kochan, Helion 2012
4. Podstawy języka Swift: programowanie aplikacji dla platformy iOS / Mark A. Lasso & Tom Stachowitz, Helion 2016
5. Service design patterns: fundamental design solutions for SOAP/WSDL and RESTful Web services, Robert Daigneau, Addison-Wesley, 2012
6. Android : programowanie aplikacji / Dawn Griffiths, David Griffiths, Helion 2016

Additional

1. The Swift Programming Language 3.1, Apple Inc., 2017
2. Using Swift with Cocoa and Objective-C, Apple Inc., 2014

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	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) ¹	40	2,0

¹ delete or add other activities as appropriate

