



COURSE DESCRIPTION CARD - SYLLABUS

Course name

PO 6.2.2 Programowanie terminali mobilnych (WM i iOS) - EC 6.2.2 Programming of Mobile Terminals in IOS

Course

Field of study

Teleinformatics

Year/Semester

3/6

Area of study (specialization)

Profile of study

general academic

Level of study

first-cycle studies

Course offered in

Polish

Form of study

full-time

Requirements

elective

Number of hours

Lecture

15

Laboratory classes

30

Other (e.g. online)

Tutorials

0

Projects/seminars

0/0

Number of credit points

3

Lecturers

Responsible for the course/lecturer:

Responsible for the course/lecturer:

dr inż. Paweł Sroka
Institute of Radiocommunications
pawel.sroka@put.poznan.pl
phone: +48 61 665 3813

dr inż. Marcin Rodziewicz
Institute of Radiocommunications
e-mail: marcin.rodziewicz@put.poznan.pl
phone: +48 61 665 3915

Prerequisites

Student starting this course should have knowledge in computer science and object-oriented programming, particularly knowing syntax of various programming languages such as: C++, C# or Java.



Moreover, students should know the fundamentals of the functioning of the operating systems, particularly Windows and Unix, and databases. Furthermore, he/she should have understanding of the architecture and operation of a programmable chips and digital communication systems. Furthermore, students should be able to implement advanced algorithms using selected programming languages and be able to find the required solutions for identified problems in different sources. Students should be also aware of their limitations and skills and the need to pursue with their education. Finally, they should understand the need for professional treatment of the problems and their responsibility for developed solutions.

Course objective

The main goal of the course is to provide students with knowledge on rules of programming of mobile devices, and to develop student's skills in programming of mobile terminals focusing on the development of mobile application devices using available frameworks, libraries, data structures and application programming interfaces (APIs). After completing the course students will be able to implement their own application using rich resources and tools, test it and, finally, they will know how to publish it in the Internet market. Student will be also aware of the dynamic development in the area of programming of mobile devices and will understand the need to continuously study the newest achievements.

Course-related learning outcomes

Knowledge

1. Has ground knowledge in the area of programming of mobile terminals.
2. Has knowledge about the design and object-oriented programming of mobile applications, their architecture and lifecycle.
possibilities of usage of.
3. Knows how to use the additional various modules and resources available in nowadays mobile devices.
4. Knows how to use the available tools and libraries to extend the functionality of the implemented application.

Skills

1. Possesses the skills of using various resources available in Internet.
2. Is able to prepare the complete application, accounting for the specific features of used mobile device and its operating system, together with the required documentation.
3. Is able to verify the correctness of the developed application using testing frameworks.

Social competences

1. Is aware of his/her knowledge and skills limitations; understands the need of further study.
2. Is aware of the need for professional treatment of the problems to be solved.
3. Is aware of his/her responsibilities for the developed systems and applications.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:



The knowledge acquired in the lectures is verified in form of a written or oral exam. The written exam comprises 6-10 questions that are graded (with points) differently. The written exam is passed if at least 45% of the total score is obtained. The oral exam relies on student's answer to at least three questions about topics indicated to students during the lectures, with the evaluation taking into account the overall understanding of the problem and the completeness of the answer. The oral exam is passed if more than 50% of the answers are evaluated as sufficient.

The abilities acquired during the laboratories are verified with 3-7 exercises covering the topics introduced during lectures. Each exercise is based on implementation of application and its certain features and might be graded differently (with points) depending on the difficulty and the required work effort. The evaluation is performed based on the written report describing the implemented application and the observed involvement of the student into work. The final grade depends on the number of collected points, with a positive result achieved if at least 50% of the maximum number of points is obtained.

Programme content

Lectures comprise the following topics:

- Introduction to programming for mobile devices and the related programming languages.
- Programming architectures (models) of mobile applications and the related structure of a mobile application.
- Building of user interface for a mobile application. Handling gestures.
- Persistent data storage with a mobile application.
- Notifications and their role in application.
- Testing of mobile applications.
- Selected features of applications: obtaining location information, using maps, networking.
- Publishing an application.

In the laboratories the following topics are taught:

- Basics of mobile application. Creating a basic user interface.
- Implementation of a multi-page application.
- Working with adaptive layout (adaptive user interface)
- Persistent data storage.
- Testing of mobile applications.
- Implementation of selected features to create advanced functionalities.

Lectures comprise the following topics:

- Introduction to programming for iOS devices and to Swift programming language.
- View Controllers in iOS application - their role, types and lifecycle.
- Building of user interface for iOS application. Creating adaptive layout. Handling gestures.
- Persistent data storage on an iOS device.
- Notifications and their role in iOS application.
- Testing of iOS applications.
- Additional features of iOS applications: obtaining location information, using maps, networking.
- Publishing of an iOS application.

In the laboratories the following topics are taught:

- Creating of a simple application with basic user interface.



- Introduction to graphics.
- Implementation of a multi-page application.
- Persistent data storage.
- Implementation of adaptive layout and gestures.
- Testing of iOS applications: unit and UI tests.
- Usage of maps and location information in iOS applications.
- Creating and handling notifications in iOS.

Teaching methods

Lecture: multimedia presentation supported with practical demonstration/videos of implementation of selected applications.

Laboratories: laboratory exercises - students develop applications with specific features, according to provided instructions, on computers equipped with development environment and simulators of selected mobile devices.

Bibliography

Basic

Mark A. Lassooff, Tom Stachowitz (tł. Robert Górczyński), "Podstawy języka Swift : programowanie aplikacji dla platformy iOS", Helion, 2016.

Matt Neuburg (tł. Robert Górczyński), "iOS 12 : wprowadzenie do programowania w Swifcie", Helion, 2019.

Additional

N.Smyth, "iOS 10 App Development Essentials", CreateSpace Independent Publishing Platform, 2016
<https://developer.apple.com>
<https://www.appcoda.com/>
<https://www.raywenderlich.com/>

Breakdown of average student's workload

| | Hours | ECTS |
|--|-------|------|
| Total workload | 86 | 3.0 |
| Classes requiring direct contact with the teacher | 45 | 2.0 |
| Student's own work (preparation for tests, preparation for laboratory classes, literature studies) | 41 | 1.0 |