



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Advanced programming of mobile terminals [S2EiT1-TMiB>ZPTM]

Course

Field of study

Electronics and Telecommunications

Year/Semester

1/2

Area of study (specialization)

Mobile and Wireless Technologies

Profile of study

general academic

Level of study

second-cycle

Course offered in

Polish

Form of study

full-time

Requirements

elective

Number of hours

Lecture

15

Laboratory classes

30

Other

0

Tutorials

0

Projects/seminars

15

Number of credit points

4,00

Coordinators

dr inż. Paweł Sroka

pawel.sroka@put.poznan.pl

Lecturers

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++ or C# or Matlab. Moreover, students should know the fundamentals of the functioning of the operating systems and databases, have understanding of the architecture and operation of a programmable chip and have deepened knowledge of current available communication systems. Furthermore, students should be able to implement advanced algorithms using selected programming languages, be able to find the required solutions for identified problems in different sources and be prepared to work in group. 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 develop student's skills in programming of mobile terminals focusing on the devices working with iOS operating system and online tools provided by Apple or other parties. After completing the course students will be able to implement their own application with different functionality using rich resources and tools, test it and, finally, they will know how to prepare it for publication in the Internet market.

Course-related learning outcomes

Knowledge:

1. Has ground knowledge in the area of programming of mobile terminals.
2. Has knowledge about the possibilities of usage of various modules and resources available in nowadays mobile devices.
3. Knows how to use the additional 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 together with the required documentation.
3. Is able to verify the application implementation correctness using testing mechanisms.

Social competences:

1. Is aware of his/her knowledge and skills limitations; understands the need of further study caused by information aging in the area of iOS programming.
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:

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, graded between 2 and 5, 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 4-7 exercises covering the topics introduced during lectures and described in the laboratory instructions. 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.

Skills and competences acquired during the project classes are evaluated based on the realization of a large task - a project - relying on the implementation of an advanced application for iOS, with the topic and scope of work agreed with the student. Projects might be realized individually or in groups comprising at most 3 persons. The final evaluation for the project task, graded between 2 and 5, depends on the level of difficulty of the task, completed modules/functionalities and the involvement of student into work.

Programme content

The course covers the following topics:

- Introduction to programming for iOS devices.
- Creating basic applications and building of user interface.
- Creating adaptive and responsive user interface. Handling gestures.
- Persistent data storage on an iOS device.
- Networking with iOS devices.
- Other selected features of iOS applications.

Course topics

Lectures comprise the following topics:

- Introduction to programming for iOS devices and to Swift programming language.
- Structure and lifecycle of an iOS app. Scene and its lifecycle.
- Building of user interface for iOS application using UIKit and SwiftUI. Creating adaptive layout and handling gestures.

- Persistent data storage on an iOS device, including use of CoreData and SwiftData.
- Networking in iOS apps - HTTP and MultipeerConnectivity
- Notifications and their role in application.
- Multithreading and concurrency in iOS.
- Additional features of iOS applications: testing, obtaining location information, using maps, publishing a application.

In the laboratories the following topics are taught:

- Creating of a simple application with basic user interface.
- Implementation of a multi-page application.
- Persistent data storage.
- Networking in iOS application - downloading data and P2P connectivity.
- Implementation of adaptive layout and gestures.
- Testing of iOS applications: unit and UI tests.
- Use of maps and location tracking in application.
- Scheduling and handling notifications.

Project topics cover the design and implementation of a multi-page application with adaptive user interface, making use of advanced features available in discussed iOS libraries and frameworks (e.g., executing other applications, persistent data storage, usage of electronic modules available in the device). The functionality of the implemented application is then verified using the iOS simulator or a physical device.

Teaching methods

Lecture: multimedia presentation.

Laboratories: laboratory exercises - students develop iOS applications with specific features, according to the provided laboratory instructions, on Mac computers equipped with iOS device simulator.

Project: Realization of the project task - design and implementation of an application with specific functionality using the Mac computers equipped with the needed software and iOS simulator and, eventually, additional tools available in the Internet. The project tasks may be realized in groups.

Bibliography

Basic

Mark A. Lassoﬀ, 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	100	4,00
Classes requiring direct contact with the teacher	70	3,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	30	1,00