



## COURSE DESCRIPTION CARD - SYLLABUS

Course name

Programming of mobile terminal

### Course

Field of study

Electronics and Telecommunications

Area of study (specialization)

Level of study

Second-cycle studies

Form of study

full-time

Year/Semester

II/III

Profile of study

general academic

Course offered in

English

Requirements

elective

### Number of hours

Lecture

30

Laboratory classes

15

Other (e.g. online)

Tutorials

15

Projects/seminars

### Number of credit points

4

### Lecturers

Responsible for the course/lecturer:

dr inż. Marcin Rodziewicz

Responsible for the course/lecturer:

marcin.rodziewicz@put.poznan.pl

### Prerequisites

The student should possess a basic knowledge on programming, architecture of computer systems and operating systems. The student should also have knowledge on current wireless communications systems. The student should be able to implement basic algorithms and be able to find necessary information from different resources. The student should also be able to work in a team.

### Course objective

The main goal of the course is to develop student's skills in programming of mobile terminals. After completing the course students will be able to implement their own application, ready for release in Internet markets. Particular attention will be put on the devices using Android system.

### Course-related learning outcomes

Knowledge

1. Possesses the grounded knowledge in the area of programming of mobile terminals.
2. Possesses the knowledge about the possibilities of usage of various modules and resources available in nowadays mobile terminals.



### Skills

1. Possesses the skills of using various resources available in Internet (usually in English).
2. Is able to prepare the complete application together with the required documentation.

### Social competences

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

### Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Theoretic knowledge (based on the lectures) will be checked during a test or oral verification scheduled on the last lecture.

The test consists of 10 questions (open and multiple-choice). Passing grade threshold: 50%.

The following grades scale is assumed:

pkt.	ocena
<= 50%	2.0
51%-60%	3.0
61%-70%	3.5
71%-80%	4.0
81%-90%	4.5
91%-100%	5.0

The oral verification of knowledge is based on answering 3-5 questions. Each question will be graded using the 2-5 scale. The final grade will be the average of grades for individual questions. The passing grade is given if the average is higher than 3.0.

The topics for the test and the oral examination will be provided to the students on the webpage specified by the lecturer.

Laboratories will be credited based on the tasks assigned during laboratories. Each task will be graded using the 2-5 scale. The final grade will be the average of grades for individual tasks.

Tutorial will be evaluated based on a project. The project will be graded using the 2-5 scale based on the its presentation. The projects can be completed individually or in small teams (up to 3 persons)



## Programme content

### Lecture:

1. Description of the programming environment (AndroidStudio).
2. Description of the project structure and resources used in the project.
3. Description of the rules for creating user interface of an application and related layouts and views.
4. Description of the Activity concept and its lifecycle.
5. Description of the Fragment concept and the ways of managing the fragments.
6. Description of the build and debugging process.
7. Description of the Intent concept.
8. Description of other application components: Services, Broadcasts
9. Description of the application manifest
10. Description of background processing and threading
11. Description of menus and notifications in applications
12. Description of input events - usage of event listeners and handlers
13. Description of storage options in Android applications
14. Descriptions of sensors available on Android devices
15. Description of selected external tools useful for application development: Google services, Firebase, Volley

### Laboratories:

1. Introduction to AndroidStudio IDE.
2. Introduction to application layout basics.
3. Introduction to rules and methods of creating application components (Activities, Fragments, Dialogs, Lists).
4. Introduction to methods of utilizing hardware capabilities of devices:
5. Introduction to Google Services.

### Tutorials:



1. AndroidStudio IDE.
2. Application basics.
3. Application components (Activities, Fragments, Dialogs, Lists).
4. External libraries

### Teaching methods

1. Lecture: Multimedia presentation
2. Laboratories: Performing tasks given in instructions provided by the supervisor
3. Tutorials: Presentation with code and application examples

### Bibliography

Basic

<https://developer.android.com>

Additional

### Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,0
Classes requiring direct contact with the teacher	70	3,0
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests, homework,project) <sup>1</sup>	30	1,0

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