



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Java programming

Course

Field of study

Electronics and Telecommunications

Area of study (specialization)

Level of study

First-cycle studies

Form of study

full-time

Year/Semester

3/5

Profile of study

general academic

Course offered in

English

Requirements

elective

Number of hours

Lecture

30

Laboratory classes

30

Other (e.g. online)

Tutorials

Projects/seminars

Number of credit points

5

Lecturers

Responsible for the course/lecturer:

dr hab. inż. Mariusz Żal,
mariusz.zal@put.poznan.pl

Responsible for the course/lecturer:

Prerequisites

Students have a basic knowledge of computer networks; Has a basic knowledge of C++ programming. Students are able to find information in literature, as well as other reference sources; is able to integrate and interpret obtained information, draws conclusions and justifies. Student understands a necessity to acquire a new knowledge and skills stemming from a chosen field of studies.

Course objective

This course provides an introduction to object oriented programming (OOP) using the Java programming language. Allows students to understand the fundamentals of Java programming such as variables, conditional and iterative execution, methods as well as GUI programming.

Course-related learning outcomes

Knowledge

1. Knows the principles of construction of computer programs; has knowledge from the area of computing science; knows the syntax of Java
2. Has a systematic knowledge of computer architectures.



Skills

1. A student is able to develop a console application using Java language.
2. A student is able to develop a windows application using Java language.
3. A student is able to prepare application using event-driven programming model.

Social competences

1. Demonstrates responsibility for designed software. Is aware of the hazards they pose for individuals and communities if they are improperly designed.
2. A student is able to formulate opinions concerning challenges of contemporary networks application programming;
3. A student is aware of the impact of network application on the information society

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Student's knowledge is verified during test (either written or oral). Test in the written form contains 7-10 questions (open questions and multi-choice questions) with different points assigned to each question. There are three or four groups of points. In oral test a student draws one question from each group. Moreover, for each drawn question an extra question (related to drawn question) may be asked. Rating for each question (drawn question and extra question are considered together) depends on range and depth of understanding of a problem. In both, written and oral form, for the test 50-60 questions are prepared. The test is passed if the a student gets at least 50% of the total score.

Verification of student skills is conducted through project that is realized during the last laboratory. Project is divided into 5-6 tasks with different points assigned to each task. All task form a whole problem but can be realized separately. Particular tasks are rating separately. For a pass, student need to get at least 50% of the total score.

Grading scale:

number of points	grade
<=50 %	2,0
51% - 60%	3,0
61% - 70%	3,5
71% - 80%	4,0
81% - 90%	4,5
91% - 100%	5,0

Programme content

Lectures:

1. Fundamentals of Java
2. Variables and basic data types, arrays
3. Controls and loops
4. Methods, method visibility, static methods



5. Class and Object
6. Class Inheritance, abstract classes
7. Interfaces
8. Strings
9. Exception Handling
10. File processing
11. Streams family
12. Events and delegates
13. GUI applications
14. Multithread programming

Laboratory classes:

1. Fundamentals of Java
2. Control and loops
3. Dynamic collections
4. Interfaces - sorting algorithms
5. Exception Handling
6. Streams vs static structures
7. GUI applications

Teaching methods

Lectures:

- a) multimedia presentation with additional examples presented and explained on a board,
- b) case study based on the presentation with usage of runtime environment or Java IDE,

Laboratory classes:

- a) practical programming exercises with computers and Java IDE,
- b) short multimedia presentations

Bibliography

Basic

1. Gosling J., Joy B., Steele G., Bracha G., Buckley A., Smith D. , The Java® Language Specification Java SE 11 Edition, available online: <https://cr.openjdk.java.net/~iris/se/11/latestSpec/java-se-11-jls-draft-diffs.pdf>

Additional

1. Anthony Potts, David H. Friedel Jr, Java programming language handbook, Coriolis Group Inc. 1996
2. Cay S. Horstmann, Gary Cornell, Core Java Volume I--Fundamentals, Prentice Hall; 9 edition (December 7, 2012)



Breakdown of average student's workload

	Hours	ECTS
Total workload	125	5,0
Classes requiring direct contact with the teacher	70	3,0
Student's own work (literature studies, preparation for laboratory classes, preparation for tests) ¹	55	2,0

¹ delete or add other activities as appropriate