STUDY MODULE DESCRIPTION FORM

Name of the module/subject		Code
Programming platforms		**1010334561010334966**
Field of study	Profile of study (general academic, practical)	Year /Semester
Information Engineering	**(brak)**	**3 / 6**
Elective path/specialty	Subject offered in:	Course (compulsory, elective)
-	**Polish**	**obligatory**
Cycle of study:	Form of study (full-time,part-time)	
First-cycle studies	**part-time**	
No. of hours		No. of credits
Lecture: **16** Classes: **-** Laboratory: **16** Project/seminars: **-**		**4**
Status of the course in the study program (Basic, major, other)	(university-wide, from another field)	
(brak)	**(brak)**	
Education areas and fields of science and art		ECTS distribution (number and %)
technical sciences		**4 100%**
Technical sciences		**4 100%**

Responsible for subject / lecturer:

dr inż. Michał Ciesielczyk
email: Michal.Ciesielczyk@put.poznan.pl
tel. 61 647 5988
Faculty of Electrical Engineering
ul. Piotrowo 3A 60-965 Poznań

Prerequisites in terms of knowledge, skills and social competencies:

1	**Knowledge**	K_W04: possesses ordered and theoretically founded knowledge on the basic algorithms and analytic techniques for designing algorithms, abstract data structures and their implementation, computationally difficult problems; K_W08: has structured and theoretically founded knowledge on databases and data warehouses; K_W012: has ordered and methodological knowledge of software engineering
2	**Skills**	K_U02:is able to work independently and in a team, is able to estimate the time needed for the commissioned tasks, able to develop and implement a schedule of work to ensure deadlines, K_U03: is able to develop documentation of engineering tasks and prepare a text containing a discussion of the results of this task
3	**Social competencies**	K_K04:is aware of responsibility for his/her own work and a willingness to comply with the principles of teamwork and shared responsibility for the implementation of tasks

Assumptions and objectives of the course:

To acquaint the students with basic programming platforms in Java and Python.

Study outcomes and reference to the educational results for a field of study

Knowledge:

1. Student has organized knowledge with theoretical foundations of basic program constructions, algorithm implementations, paradigms and programming styles, software verification methods, formal languages, compilers, platforms. - [K_W05]
2. Student is familiarized with state of the art and current trends in computer science. - [K_W19]
3. Student knows common IT engineering technology. - [K_W18]

Skills:

1. Student is able to use software platforms and environments for simple programs encoding, running and testing in imperative, object-oriented and declarative programming languages. - [K_U10]
2. Student is able to prepare requirements, to create object model and to evaluate uncomplicated IT system, including system functions and relations between system elements. - [K_U16]
3. Student is able to evaluate tools and methods usefulness for simple engineering tasks related to computer science. Student is able to choose and to implement proper technologies. - [K_U22]

Social competencies:

1. understands the need and knows the opportunity of continuous training (second-and third-degree, postgraduate courses) ? improvement of language, professional, personal and social skills - [K_K01]

Assessment methods of study outcomes

Lecture: written test that checks the basic knowledge of programming platforms and paradigms.

Laboratory: regular assessment during the course, project.

Course description

Lectures and laboratories cover the following topics:

- Introduction to object-oriented programming in Java.
- Collections and generic types in Java.
- Software build automation tools.
- Database access via JDBC/JPA interface.
- Introduction to the JavaFX graphical library.
- Fulltext search using Apache Lucene.
- Test-driven Development (TDD) methodology.
- Test automation on JUnit example.
- Introduction to programming in Python.
- Database access using SQLAlchemy.

Teaching methods:

lectures - lectures including multimedia presentation supported by the examples, the theory presented in close connection with practice;

laboratories - laboratories implemented in accordance with provided instructions, the use of open-access tools, demonstrations and reports.

Update 2017: Java SE 8, JavaFX 8, JUnit 4, Python 3.5

Basic bibliography:

1. Oracle (2017). The Java Tutorials. http://docs.oracle.com/javase/tutorial/
2. Oracle (2017). Java Platform, Standard Edition 8 API Specification. https://docs.oracle.com/javase/8/docs/api/
3. Oracle (2017). Outline of the Collections Framework. http://docs.oracle.com/javase/8/docs/technotes/guides/collections/reference.html
4. Oracle (2017). Java SE Technologies - Database. http://www.oracle.com/technetwork/java/javase/jdbc/index.html
5. Oracle (2017). JDBC(TM) Database Access. http://docs.oracle.com/javase/tutorial/jdbc/index.html
6. McCandless M., Hatcher E., Gospodnetić O. (2010). Lucene in Action, Second Edition. Chapter 1. http://www.manning.com/hatcher3/
7. JUnit (2017). JUnit. http://www.junit.org/
8. Python Software Foundation (2017). Welcome to Python. https://www.python.org/
9. Python Software Foundation (2017). Python 3.x documentation. https://docs.python.org/3/

Additional bibliography:

1. Risberg T. (2017). Spring Data JDBC Extensions Reference Documentation. http://docs.spring.io/spring-data/data-jdbc/docs/current/reference/pdf/spring-data-jdbc-ext-reference.pdf
2. Srinivasan K. (2007). Introduction to Java Persistence API(JPA). http://javabeat.net/jpa/
3. The Apache Software Foundation (2017). Apache Lucene. http://lucene.apache.org/
4. Chin S. (2017). JavaFX: Making it Easier to Build Better RIAs. https://dzone.com/refcardz/getting-started-javafx
5. Oracle (2017). Writing JUnit Tests in NetBeans IDE. https://netbeans.org/kb/docs/java/junit-intro.html
6. LearnPython.org (2017). Free Interactive Python Tutorial. http://www.learnpython.org/pl/
7. SQLAlchemy (2017). Object Relational Tutorial. http://docs.sqlalchemy.org/en/rel_0_9/orm/tutorial.html

Result of average student's workload

Activity	Time (working hours)
1. Lectures	16
2. Laboratories	16
3. Preparation to laboratories	32
4. Independent work on topics discussed in lectures	16
5. Consultations	5

Student's workload		
Source of workload	**hours**	**ECTS**
Total workload	82	4
Contact hours	34	2
Practical activities	48	2