Subject (course) name: **Programming in JAVA**		
Programme: **Computer Science** Specialty:		Subject code: **15**
		Title graduate: **Engineer**
Type of course: **obligatory**	Course level: **First-cycle studies**	Year: **III** Semester: **VI** Semester: **spring**
Form of classes: **Lectures, Classes, Labs, Seminar, Project**	Number of hours per week: **1L, 0, 2Lab, 0, 0**	Credit points: **4 ECTS**

GUIDE TO SUBJECT

SUBJECT OBJECTIVES

C1. Acquaintance with NetBeans integrated development environment.

C2. Learning syntax of JAVA programming language.

C3. Acquaintance with using selected classes from JAVA standard API library.

C4. Learning basic skills in programming graphical user interface in JAVA.

C5. Acquaintance with methods of reading and writing files on file system.

C6. Learning methods of drawing basic two-dimensional shapes on screen and making animations in JAVA.

SUBJECT REQUIREMENTS

1. General knowledge in programming concepts such as conditional statements and loops.
2. General ability with computer handling.

LERNING OUTCOMES

EK 1 – Student can create a simple project in NetBeans and create a form containing graphical user interface components.

EK 2 – Student knows methods of class String in JAVA and can perform elementary operations on text strings.

EK 3 – Student knows conditional statements and loops in JAVA.

EK 4 – Student knows how define constructors in classes and knows how create inheritance in own class set.

EK 5 – Student knows how to use dynamic data structures in JAVA.

EK 6 – Student can draw 2D graphics using JAVA instructions.

SUBJECT CONTENT

Form of classes - lectures

contents	hours
W 1 – NetBeans integrated development environment. Creating basic projects in NetBeans. Adding graphical components. Writing Java code as an answer to events generated by graphical components.	**1**
W 2 – Syntax of the JAVA programming language. Conditional statements. The String class and its methods.	**1**
W 3 – Functions of graphical components JLabel, JButton, JTextField. Displaying dialog windows with static methods of JOptionPane class. The JPanel component.	**1**
W 4 – Loop instruction in JAVA programming language. Tables in JAVA. Integer and Float class and their methods.	**1**
W 5 – Object oriented programming in JAVA. Class variables, methods. Static variables and methods. Creating objects with constructors.	**1**
W 6 – Inheritance in JAVA programming language. Class hierarchy. Abstract classes and interfaces.	**1**
W 7 – Dynamic data structures. The ArrayList class and its methods.	**1**
W 8 – Dynamic data structures. – collections nad maps. Class Set, Collection and HashMap. Using iterator to view all items from collection. The for..each loop in JAVA programming language.	**1**
W 9 – counting the real time in JAVA. The Timer class and the abstract class TimerTask. Implementation of the abstract method TimerTask.	**1**
W 10 – Implementation of event processing in JAVA using interfaces Class DocumentListener and ListSelectionListener. The JList component and class DefaultListModel as a data container.	**1**
W 11 – Creating complex forms in GUI. An example of car database GUI in Java.	**1**
W 12 – Errors processing in JAVA with exceptions. the keywords try, catch, finally.	**1**
W 13 – Writing and reading files in JAVA. Serialization in JAVA.	**1**
W 14 – Drawing 2D shapes. The PAINT event.	**1**
W 15 – 2D animation programming in JAVA languages. Buffering drawing operations in memory bitmap.	**0,5**
Test	**0,5**
SUM	**15**

Form of classes – laboratory

contents	hours
Test on preliminary requirements	**0,5**
L 1 – NetBeans integrated development environment. The „Hello Word" application in NetBeans. Graphical components JTextField, JLabel, JButton. Dialog windows, the JOptionPane class. The JPanel component.	**1,5**
L 2 – The methods of String class. An application simulating the user registration and logging process.	**4**
L 3 – Loop i JAVA programming language. Processing data in tables.	**2**
L 4 – Creating a library of classes and inheritance hierarchy	**2**
L 5 – The ArrayList and its methods. Nesting ArrayList classes.	**2**
L 6 – Set, Collection i HashMap classes. Viewing collections with iterators and for..each loop.	**2**
L 7 – The Timer class. The abstract class TimerTask. A simple animation that switch background on JPanel components.	**2**
L 8 – Using dialog windows JColorChooser and JFileChooser.	**2**
L 9 – Writing simple database with GUI	**4**
L 10 – Writing and reading files in JAVA programming languages.	**2**
L 11 – Exceptions propagation. The debugger in NetBeans.	**2**
L 12 – Drawing 2D shapes in JAVA.	**2**
L 13 – 2D animation in JAVA	**1,5**
Final test	**0,5**
SUMA	**30h**

STUDY METHODS

1. Lectures using multimedia presentations
2. Laboratory – analysis of the operation and development of software

EDUCATIONAL TOOLS

1. Audiovisual equipment, black(white)board, lectures in electronic version
2. Dedicated software for presentation of chosen aspects discussed during lectures

METHODS OF ASSESMENT (F – Forming, P – Summary)

F1. assessment of the correctness and timeliness of presentation software created
P1. lecture – written test of the theory and completion of tasks in computer arithmetic
P2. laboratory – assessment of ability to software analysis and software development

STUDENT WORKLOAD

Form of activity	Averaged workload (hours)		
	[h]	∑ [h]	ECTS
Participation in class activities lecture laboratory consultation	15 30 3	45	3
Preparation for tutorials (reading literature)	20	45	1
Preparation for test	15		
Familiarizing yourself with programming software	10		
Total		**90**	**4**

A. BASIC READING

1. Herbert Schildt *: JAVA: A Beginner's Guide, Sixth Edition*, 2nd edition, 2014.
2. Joshua Bloch: Efective Java (2nd edition), 2008.

B. FURTHER READING

1.. http://www.oracle.com/technetwork/java/javase/downloads/index.html
2. http://www.oracle.com/technetwork/java/index.html

Learning objectives	In relation to the learning outcomes specified for the field of study	Subject objectives	Subject content	Course study methods
EK1	K_W11, K_U17	C1	lecture laboratory	P1, P2, P3
EK2	K_W05, K_W14	C3	lecture laboratory	P1, P2, P3
EK3	K_W05, K_W14	C2	lecture laboratory	P1, P2, P3
EK4	K_W05, K_W14	C2	lecture laboratory	P1, P2, P3
EK5	K_W05, K_W07	C2, C3	lecture laboratory	P1, P2, P3
EK6	K_W11, K_U17	C5, C6	lecture laboratory	P1, P2, P3

II. EVALUATION

Grade	Outcome
EK1	**Student can create a simple project in NetBeans and create a form containing graphical user interface components.**

2 (D)	Student cannot create JAVA project in NetBeans.
3 (C)	Student can create a JAVA project in NetBeans and is able to describe properties of at least two graphical components
4 (B)	Student is able to connect JAVA code with events triggered by components.
5 (A)	Student can create JAVA code that change properties of JLabel, JButton and JPanel properties.
EK2	**Student knows methods of class String in JAVA and can perform elementary operations on text strings.**
2 (D)	Student doesn't know any method of the Java string class.
3 (C)	Student knows at least three method of Java String class and is able to use them in Java code.
4 (B)	Student knows at least six method of Java String class and is able to use them in Java code.
5 (A)	Student knows at least six method of Java String class and at least three static method of this class. Student is able to use them in Java code.
EK3	**Student knows conditional statements and loops in JAVA**
2 (D)	Student doesn't know any conditional statements and loops in Java.
3 (C)	Student knows the if..else and switch statements in java and is able to use them.
4 (B)	Student knows the for, while and do..while loops in java and is able to use them.
5 (A)	Student knows the for each loop in java and is able to use it.
EK4	**Student knows how define constructors in classes and knows how create inheritance in own class set.**
2 (D)	Student cannot define constructors in classes and cannot implement basic inheritance in classes.
3 (C)	Student is able to define a simple non-parametric constructor in class and is able to call it while creating new objects.
4 (B)	Student can write a parametric constructor and is able to use this keyword.
5 (A)	Student can implement inheritance in a set of several classes.
EK5	**Student knows how to use dynamic data structures in JAVA**
2 (D)	Student doesn't know any class that represent a dynamic data structure in JAVA.
3 (C)	Student knows the ArrayList class and is able to store objects inside it.
4 (B)	Student knows the HashMap class and is able to store objects inside it.
5 (A)	Student can write a class in Java that can be used as stack.
EK6	**Student can draw 2D graphics using JAVA instructions**
2 (D)	Student cannot draw any basic 2D shapes in JAVA.
3 (C)	Student can draw basic 2D shapes in Java.
4 (B)	Student knows how to process Paint event.
5 (A)	Student can make a simple 2D animation.

III. OTHER USEFUL INFORMATION

1. All information for students on the schedule are available on the notice board and on the website: https://we.pcz.pl/
2. Information on the consultation shall be provided to students during the first lecture and will be placed on the website https://we.pcz.pl/
3. Terms and conditions of credit courses will be provided to students during the first lecture