Course Syllabus

Course from study programme for the cycle: 2023/2024

I. General Information

Course name	Object-oriented programming II
Programme	Informatics
Level of studies (BA, BSc, MA, MSc, long-cycle MA)	BA
Form of studies (full-time, part-time)	Full-time
Discipline	Informatics
Language of instruction	English

Course coordinator	Dorota Pylak, PhD

Type of class *(use only the types mentioned below)*	Number of teaching hours	Semester	ECTS Points
lecture	30	IV	5
tutorial			
classes			
laboratory classes	30	IV	
workshops			
seminar			
introductory seminar			
foreign language classes			
practical placement			
field work			
diploma laboratory			
translation classes			
study visit			

Course pre-requisites	Introduction to computer science. Fundamentals of algorithms and programming Object-oriented programming

II. Course Objectives

Familiarizing students with the methodology and technique of object-oriented programming - continuation.
Deepening the knowledge of Java

III. Course learning outcomes with reference to programme learning outcomes

Symbol	Description of course learning outcome	Reference to programme learning outcome
KNOWLEDGE		
W_01	The student recognizes selected elements of the Java language, presents the concepts of object-oriented programming	K_W01 K_W03
W_02	The student is familiar with the basic algorithms and examples of their practical implementation using concepts of the object-oriented programming	K_W01, K_W03, K_W06
W_03	The student recognizes the elements of functional programming	K_W01, K_W03 K_W06
W_04	The student knows the possibilities of sample Java classes and interfaces	K_W01, K_W03
SKILLS		
U_01	The student is able to recognize and apply classes, interfaces, selected collections, and program algorithms. Uses parameterized types, exceptions, selected streams and simple regular expressions.	K_U04, K_U07, K_U08, K_U10, K_U11, K_U12
U_02	The student is able to create console applications and use the IDE programming environment	K_U04, K_U07, K_U08, K_U10, K_U11, K_U12, K_U17
U_03	The student is able to use the elements of functional programming. Creates a simple lambda expression	K_U04, K_U07, K_U08, K_U10, K_U11, K_U12, K_U17
U_04	The student is able to create applications using selected Java classes and interfaces	K_U04, K_U07, K_U08, K_U10, K_U11, K_U12, K_U17
SOCIAL COMPETENCIES		
K_01	The student can communicate and cooperate in professional environment	K_K01

IV. Course Content

1. Exceptions (Java).
2. Enum
3. Generic types
4. Lists, collections, maps
5. Streams
6. Inner and anonymous classes
7. Built-in Java interfaces like Comparator and Comparable
8. Lambda expressions and stream programming. Optional

9. String and StringBuilder.
10. Regular expressions -introduction

V. Didactic methods used and forms of assessment of learning outcomes

Symbol	Didactic methods *(choose from the list)*	Forms of assessment *(choose from the list)*	Documentation type *(choose from the list)*
KNOWLEDGE			
W_01	Conventional lecture / Guided practice	Exam/Written test	Examination card / written test/report file
W_02	Conventional lecture / Guided practice	Exam/Written test	Examination card / written test/report file
SKILLS			
U_01	-practical classes -design thinking	Exam/Written test	Examination card / written test/report file
U_02	-practical classes -design thinking	Exam/Written test	Examination card / written test/report file
U_03	-practical classes -design thinking	Exam/Written test	Examination card / written test/report file
SOCIAL COMPETENCIES			
K_01	Discussion, PBL (Problem-Based Learning) design thinking	Exam/Written test	Examination card / written test/report file
K_02	Discussion, PBL (Problem-Based Learning) design thinking	Exam/Written test	Examination card / written test/report file

VI. Grading criteria, weighting factors.....

To pass a course, the student has to attend a classes and has to pass the tests and the final exam.

- passing classes - colloquia - 90% of the final grade, student's activity and work during classes - 10% of the final grade.

- written exam - for people who have passed the classes. Detailed conditions of exemption are given to students with each course edition.

Detailed assessment rules are given to the students with each edition of the course.

VII. Student workload

Form of activity	Number of hours
Number of contact hours (with the teacher)	**90**
Number of hours of individual student work	**60**

VIII. Literature

Basic literature
Herbert Schildt,Java: The Complete Reference, Eleventh Edition,McGraw-Hill Education, 2018 Herbert Schildt, Java: A Beginner's Guide, Eighth Edition, McGraw-Hill Education, 2018 http://docs.oracle.com/javase/8/docs/ http://docs.oracle.com/javase/11/docs/ C. S. Horstmann, G. Cornell, Core Java Volume I – Fundamentals (10th Edition), Pearson Education, 2018 C. S. Horstmann, Java, Core Java, Volume II--Advanced Features, 11th Edition, Pearson Education, 2019 http://download.oracle.com/javase/tutorial/
Additional literature
R. Sedgewick, K. Wayne, Algorithms, 4th ed., Addison-Wesley, Upper Saddle River, NJ, 2011. N. Wirth, Algorithms + Data Structures = Programs, Prentice-Hall 1976