

Course Syllabus

Course from study programme for the cycle: 2022/2023

I. General Information

| | |
|--|-----------------------------|
| Course name | Object-oriented programming |
| 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 (<i>use only the types mentioned below</i>) | Number of teaching hours | Semester | ECTS Points |
|---|--------------------------|-------------------------|-------------|
| lecture | 30 | INF: III MATH: III or V | 5 |
| tutorial | | | |
| classes | | | |
| laboratory classes | 30 | INF: III MATH: III or V | |
| 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 |
|-----------------------|---|

II. Course Objectives

| |
|--|
| Familiarizing students with the methodology and technique of object-oriented programming. |
| Familiarizing students with the Java programming language Presentation of features of the modern programming language |

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 is able to present the basic concepts of object oriented programming | INF: K_W01, K_W06, MAT: K_W01, K_W04 |
| W_02 | The student is able to analyze the source files of object oriented applications | INF: K_W03, K_W06; MAT: K_W01, K_W04 |
| SKILLS | | |
| U_01 | The student is able to apply the rules for defining classes, creating objects and modeling selected issues in an object-oriented way | INF: K_U04, K_U06, K_U08, K_U11, K_U12, K_U17 MAT: K_U38 |
| U_02 | The student is able to write an application in an object-oriented programming language | INF: K_U07, K_U08, K_U10, K_U11, K_U12, K_U17 MAT: K_U38 |
| U_03 | The student is able to use inheritance and polymorphism, abstract classes and interfaces | INF: K_U08, K_U10, K_U11, K_U12, K_U17 MAT: K_U38 |
| SOCIAL COMPETENCIES | | |
| K_01 | The student is able to formulate a solution to the given problem, is open to the new solutions | INF: K_K01 MAT: K_K02, K_K05 |
| K_02 | The student solves the given problems individually and while working in a group. | INF: K_K01 MAT: K_K02, K_K05 |

IV. Course Content

| |
|--|
| <ol style="list-style-type: none"> 1. Paradigm of object-oriented programming 2. The concepts of class and object, fields, methods, constructors, accessibility 3. Static fields and methods in classes 4. String, Math and Scanner classes- examples of usage. 4. Inheritance 5. The Object class and its methods. 6. Polymorphism. 7. Abstract classes, 8. Interfaces |
|--|

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 |
| 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 |