Object-Oriented Programming

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Level of education : first-cycle studies

Form of studies : full-time studies

Study profile : general academic

Mandatory status : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.120.03337.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.POB.OL

Course coordinator

Marcin Kozik

Course instructor

Marcin Kozik

Period Semester 2	Form of verification of learning outcomes graded credit Form of teaching and hours lecture: 45 laboratory classes: 30	Number of ECTS credits 7.0

Learning outcomes for the course

Code	**Effects in the area of**	**Major learning outcomes**	**Verification methods**

Code	Effects in the area of	Major learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	object-oriented programming paradigms and at least two object-oriented programming languages.	IAN_K1_W05, IAN_K1_W06, IAN_K1_W15	graded credit
Skills – The student can:			
U1	use several object-oriented programming languages and tools for versioning, building, and testing programs in these languages.	IAN_K1_U03, IAN_K1_U04, IAN_K1_U05, IAN_K1_U08, IAN_K1_U15, IAN_K1_U20, IAN_K1_U24, IAN_K1_U26	graded credit
Social competences – The student is ready to:			
K1	work independently and in a team on software development using object-oriented programming languages.	IAN_K1_K02	graded credit

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types
lecture	45
laboratory classes	30
independent solving of computer tasks	60
project preparation	60
test preparation	15

Total student workload	Number of hours 210	ECTS credits 7.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course
1.	Object-oriented programming techniques with examples in Java and C++ 1. encapsulation and information hiding, classes and subclasses, inheritance, interfaces, polymorphism, class hierarchies (Java, C++) 2. generic types (Java), templates (C++) 3. containers and iterators (Java, C++) 4. exceptions (Java, C++) 5. reflection (Java) and RTTI (C++) 6. threads (Java) 7. input/output (Java, C++) 8. garbage collection (Java) 9. GUI (Java)	W1, U1, K1

Extended information

Teaching methods:

project method, multimedia lecture, problem solving, e-learning methods

Type of classes	Forms of credit	Course credit requirements
lecture	graded credit	The lecture grade is identical to the laboratory grade
laboratory classes	graded credit	The grade consists of results from tests, evaluation of programming task solutions submitted online by students, and evaluation of a mini-project created as part of the course.

Prerequisites and additional requirements

Completed "Programming Basics" lecture.

Literature

Required

1. Effective modern c++ Scott Meyers
2. Effective Java Joshua Block