Faculty of Science and Mathematics / Computing and Information Technology (2017) /

Prerequisites	
Aims	Programming languages: The aim is becoming familiarized with the concepts of advanced software modelling tools, object-oriented modelling and UML language.
Lecturer / Teaching assistant	Igor Jovančević
Metdod	Lectures and practical work in a computer lab. Studying and independent work on practical projects. Consultations.
Week 1, lectures	Basic concepts of object-oriented modelling
Week 1, exercises	Lab work
Week 2, lectures	Overview of UML, basic concepts, building blocks
Week 2, exercises	Lab work
Week 3, lectures	Use case diagrams
Week 3, exercises	Lab work
Week 4, lectures	Class diagrams. Classes, interfaces. Object diagrams.
Week 4, exercises	Lab work
Week 5, lectures	Interaction diagrams
Week 5, exercises	Lab work
Week 6, lectures	Statechart diagrams and activity diagrams. Activities and actions
Week 6, exercises	Lab work
Week 7, lectures	Vacation week
Week 7, exercises	
Week 8, lectures	States: Transitions. Events. State diagrams.
Week 8, exercises	Lab work
Week 9, lectures	Component diagrams
Week 9, exercises	Preparation for midterm exam
Week 10, lectures	Midterm exam
Week 10, exercises	
Week 11, lectures	Implementation strategies: Associations, State diagrams
Week 11, exercises	Lab work
Week 12, lectures	Constraints: Object Constraint Language (OCL)
Week 12, exercises	Lab work
Week 13, lectures	Reverse engineering. Collaboration diagrams
Week 13, exercises	Lab work
Week 14, lectures	Object-oriented principles
Week 14, exercises	Lab work
Week 15, lectures	Design patterns
Week 15, exercises	Preparation for final exam
Student obligations	Midterm and final exam. Homework. Lab work
Consultations	
Workload	15 weeks x 6h (lectures+lab) = 90 hours 15 weeks x 2h (homework, studying) = 30 hours 30h (additional work) Total: 150 hours
Literature	
Examination metdods	Midterm (40 points), final exam(40 points), homework (20 points)

Special remarks	
Comment	
Learning outcomes	Capability to model software products from higher level of abstraction. Capability to produce, understand UML diagrams and translate them into code. Capability to programme in Java and C++ languages.