Course Name : Advanced Object Oriented Programming

Course Code	Course Type	Regular Semester	Lecture (hours/week)	Seminar (hours/week)	Lab. (hours/week)	Credits	ECTS
CMP 407	B	Fall	3.00	1.00	0.00	3.50	6.00

Lecturer	Sadije Bushati, Prof. Dr
Assistant	Sokol Shurdhi, Msc
Course language	Albanian
Course level	Master
Description	This course provides an in-depth exploration of the object-oriented programming paradigm, covering advanced concepts such as multiple inheritance, abstraction, sophisticated design patterns, and implementation of large-scale object-based applications.
Objectives	To strengthen students' understanding of core OOP concepts. To apply advanced design and programming techniques. To analyze code structures and make them more maintainable and scalable. To implement large-scale projects using OOP.
Core Concepts	Inheritance, polymorphism, and encapsulation Design patterns Abstraction and interfaces Collections and generics Unit testing and refactoring

Course Outline

Week	Topic
1	Review of Basic OOP Concepts
2	Abstract Classes and Interfaces
3	Inheritance and Polymorphism
4	Collections and Generics
5	Design Patterns: Singleton, Factory
6	Design Patterns: Observer, Strategy
7	Code Refactoring and Maintainability
8	Midterm Exam
9	Exceptions and Error Handling
10	Unit Testing and Automation Tools
11	Serialization and Persistent Objects
12	Interaction Between Software Modules
13	Structuring Large-Scale OOP Applications
14	Full Application Development Project
15	Project Presentation / Final Review
16	Final Exam

Prerequisites	The student must attend the course at a minimum rate of 75%.
Literature	• Timothy Budd – Object-Oriented Programming, 3rd Edition, Addison-Wesley, 2001.
References	• Robert C. Martin – Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008. • Erich Gamma et al. – Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994. • Herbert Schildt – Java: The Complete Reference, McGraw-Hill, latest edition.

Course Outcome

1	Students will understand advanced OOP concepts and their implementation.
2	They will be able to use and create design patterns for common software problems.
3	They will develop well-structured, maintainable large-scale applications.
4	They will apply testing and refactoring techniques to improve code quality.

Course Evaluation

In-term Studies	Quantity	Percentage
Midterms	0	0
Quizzes	0	0
Projects	1	15
Term Projects	1	15
Laboratory	0	0
Class Participation	1	10
Total in-term evaluation percent		**40**
Final exam percent		**60**
Total		**100**

ECTS Workload (Based on Student Workload)

Activities	Quantity	Duration (hours)	Total (hours)
Course duration (Including the exam week: 16x Total hours of the course)	16	4	64
Study hours outside the classroom (Preparation, Practice, etc.)	14	6	84
Duties	2	2	4
Midterms	0	0	0
Final Exam	1	2	2
Other	0	0	0
Total Work Load			**154**
Total Work Load / 25 (hours)			**6.16**
ECTS			**6.00**