

School of Engineering and Natural Sciences / Computer Engineering (English)

2022 - 2023 Academic Year

OBJECT- ORIENTED PROGRAMMING

Syllabus

Course Description						
Name	Code	Semester	T+A Hour	Credit	ECTS	
OBJECT- ORIENTED PROGRAMMING		COE2113250	Fall Semester	3+2	4	6
Prerequisites Courses	PROGRAMLAMAYA GİRİŞ					
Recommended Elective Courses						
Language of Instruction	English					
Course Level	First Cycle (Bachelor's Degree)					
Course Type	Required					
Course Coordinator	Prof.Dr. Selim AKYOKUŞ					
Name of Lecturer(s)	Prof.Dr. Tansel ÖZYER					
Assistant(s)						
Aim	The objective of this course is to teach object-oriented programming (OOP) in a problem-driven way that focuses on problem solving rather than syntax. Students learn object thinking and design, and improve their algorithmic thinking skills. They learn how and when to apply OOP techniques to a given problem effectively. The course starts with a review of fundamental techniques in Java, and then teaches objects and classes, object-oriented thinking, inheritance and polymorphism, exception handling and text I/O, abstract classes and interfaces, Java GUI basics, event-driven programming and animations, Java UI controls and multimedia, binary I/O, recursion, generics, lists, stacks, queues, and priority queues, sets and maps, implementing lists, stacks, queues, and priority queues.					
Course Content	This course contains; 1.Week: Introduction to object-oriented programming, Computer programs, Becoming familiar with Java programming environment, Algorith Design,2.Week: Fundamental Data Types, Variables, Arithmetic operators, Input and Output, Strings, Decisions, Boolean Variables and Operators, Comparing Numbers and Strings, Nested Branches,3.Week: Loops (for, do, while), Nested Loops, Random numbers, methods, Parameter passing, Return values, Methods without Return values, Variable scope, Recursive methods,4.Week: Arrays and Array Lists, Common Array Algorithms, Two-Dimensional Arrays,5.Week: Input/Output and Exception Handling, Reading and Writing Text Files, Command Line Arguments, Catching Exceptions,6.Week: Object and Classes, instance methods, Constructors, Object References, Static Variables and Methods,7. Week: Inheritance, Implementing Sub-classes, Overriding Methods, Polymorphism, Interface Types,8. Week: Graphical User Interfaces and Advanced GUI Topics,9. Week: Event-Driven Programming and Animations,10. Week: Recursion, Sorting and Searching (selection/merge/binary search),11. Week: Java Collections Framework: Lists, Stacks, Queues, and Priority Queues,12. Week: Implementing Lists, Stacks, Queues, and Priority Queues,13. Week: Introduction to Web Development with Java and Java Enterprise (J2EE),14. Week: Review.					
Course Learning Outcomes			Teaching Methods	Assessment Methods		
1. He/she will be able to describe and discuss fundamentals of object-oriented programming and languages.			1, 13	A, C		
2. He/she can use and apply fundamental concepts and constructs of object-oriented programming like class, object, polymorphism, inheritance, overriding.			1, 13	A, C		
3. He/she can use basic data types, basic control structures, loop types, functions and libraries in Java.			1, 13	A, C		
4. He/she can use and apply classes, object-oriented thinking, inheritance and polymorphism, exception handling, abstract classes and interfaces in Java.			1, 13	A, C		
5. He/she can design and develop object-oriented programs that use classes, inheritance, Java GUIs, event-driven programming and animations, Java UI controls and multimedia, binary I/O, recursion, generics, lists, stacks, queues, and priority queues, sets and maps, and priority queues.			1, 13	A, C		
Teaching Methods	1: Lecture, 13: Experiment / Laboratory					
Assessment Methods	A: Written Exam, C: Homework					
Lecture Schedule						
Sequence	Topics	Preliminary Preparation				
1	1.Week: Introduction to object-oriented programming, Computer programs, Becoming familiar with Java programming environment, Algorith Design					
2	2.Week: Fundamental Data Types, Variables, Arithmetic operators, Input and Output, Strings, Decisions, Boolean Variables and Operators, Comparing Numbers and Strings, Nested Branches					
3	3.Week: Loops (for, do, while), Nested Loops, Random numbers, methods, Parameter passing, Return values, Methods without Return values, Variable scope, Recursive methods					
4	4.Week: Arrays and Array Lists, Common Array Algorithms, Two-Dimensional Arrays					
5	5.Week: Input/Output and Exception Handling, Reading and Writing Text Files, Command Line Arguments, Catching Exceptions					
6	6.Week: Object and Classes, instance methods, Constructors, Object References, Static Variables and Methods					
7	7. Week: Inheritance, Implementing Sub-classes, Overriding Methods, Polymorphism, Interface Types					
8	8. Week: Graphical User Interfaces and Advanced GUI Topics					
9	9. Week: Event-Driven Programming and Animations					
10	10. Week: Recursion, Sorting and Searching (selection/merge/binary search)					
11	11. Week: Java Collections Framework: Lists, Stacks, Queues, and Priority Queues, Sets and Maps					
12	12. Week: Implementing Lists, Stacks, Queues, and Priority Queues					
13	13. Week: Introduction to Web Development with Java and Java Enterprise (J2EE)					
14	14. Week: Review					
Evaluation Methods		Weight(%)				
Midterm Exam		30				
General Exam		70				

Resources

- Introduction to Java Programming and Data Structures, Comp. Version (11th Edition), Y. Daniel Liang, Pearson, 2018.
- Computer Science An Interdisciplinary Approach, Robert Sedgewick, Kevin Wayne, Pearson 2018.
- Java Software Solutions, Global Edition, 9/e, Lewis & Loftus, Pearson, 2019.
- Java: An Introduction to Problem Solving and Programming, Global Edition, 8/e, Savitch, Pearson, 2019.
- Core Java Volume I and II --Fundamentals, 11/e, Horstmann, Pearson, 2019.
- Java How to Program, Early Objects, Global Edition, 11/e, Deitel & Deitel, Pearson, 2018.
- Introduction to Programming Using Java, Eighth Edition Version 8.0, 2018 David J. Eck.