



Recep Tayyip Erdogan University

Faculty of Engineering and Architecture

Computer Engineering

CE204- Object-Oriented Programming

Syllabus

Spring Semester, 2021-2022

Instructor	Asst. Prof. Dr. Uğur CORUH
Contact Information	ugur.coruh@erdogan.edu.tr
Office Number	F-301
Google Classroom Code	n24xrzo
Lecture Hours and Days	Theory - Monday (13:00-16:00) Lab – Thursday (10:00-12:00)
Lecture Classroom	İİBF-402
Office hours	Meetings will be scheduled over Google Meet with your university account and email and performed via demand emails. Please send emails with the subject starts with [CE204] tag for the fast response and write formal, clear, and short emails.
Lecture and Communication Language	English
Theory/Laboratory Course Hour Per Week	3/2 hours
Credit	4
Prerequisite	CE103- Algorithms and Programming I CE100- Algorithms and Programming II
Corequisite	TBD
Requirement	TBD

*TBD: To Be Defined.

A. Course Description

This course introduces advanced programming skills with a focus on object-oriented programming and design fundamentals using a high-level language such as Java. Object-oriented programming is the process of integrating software components into a large-scale software architecture. After learning the fundamentals of coding, this approach to software development is the next logical step, allowing for large-scale programs. The course focuses on understanding and applying object-oriented concepts such as classes, objects, data abstraction, methods, method overloading, inheritance, and polymorphism.

The class will be built around sharing expertise and guiding students to find learning methods and practice for object-oriented programming topics. Making programming applications and projects in the courses will strengthen the learning process by putting theory into practice.

B. Course Learning Outcomes

After completing this course satisfactorily, a student will be able to:

- Design applications using OO techniques
- Use the unified software development process to manage software development
- Use UML for the notation diagrams of applications
- Apply useful design and architecture patterns for software development.

C. Course Topics

- Object-oriented concepts
- Unified object-oriented analysis and design process
- Unified Model Language
- Use case analysis
- Object structure and behavior analysis
- System design
- Application architecture and design patterns
- Java implementation of object-oriented design

D. Textbooks and Required Hardware

This course does not require a coursebook. If necessary, you can use the following books and open-source online resources.

- *Timothy C. Lethbridge and Robert Laganière, Object-Oriented Software Engineering: Practical Software Development using UML and Java, McGraw Hill*
- *Walter Savitch, Absolute C++, Addison-Wesley Longman*
- *Intro to Java Programming, Comprehensive Version (10th Edition) 10th Edition by Y. Daniel Liang*
- *Harvey M. Deitel and Paul J. Deitel. 2001. Java How to Program (4th. ed.). Prentice Hall PTR, USA.*
- *Paul Deitel and Harvey Deitel. 2016. Visual C# How to Program (6th. ed.). Pearson.*
- *Additional Books TBD*

During this course, you should have a laptop for programming practices. You will have your development environment, and you will use this for examination and assignments also classroom practices.

E. Grading System

Midterm and Final grades will be calculated with the weighted average of the project or homework-based examinations. Midterm grades will be calculated between term beginning to the midterm week, and Final grades will be calculated between Midterm and Final week homeworks or projects as follow.

$$a_n = \text{Homework or Project Weight}$$

$$HW_n = \text{Homework or Project Points}$$

$$n = \text{Number of Homework or Project}$$

$$\text{Grade} = \frac{(a_1HW_1 + a_2HW_2 + \dots + a_nHW_n)}{n}$$

Homework/Exam	Weight
<i>Midterm</i>	%40
<i>Final</i>	%60

$$\text{Passing Grade} = \frac{40 * \text{Midterm}_{\text{Grade}} + 60 * \text{Final}_{\text{Grade}}}{100}$$

F. Instructional Strategies and Methods

The basic teaching method of this course will be planned to be face-to-face in the classroom, and support resources, homeworks, and announcements will be shared over google classroom. Students are expected to be in the university. This responsibility is very important to complete this course with success. If pandemic situation changes and distance education is required during this course, this course will be done using synchronous and asynchronous distance education methods. In this scenario, students are expected to be in the online platform, zoom, or meet at the time specified in the course schedule. Attendance will be taken.

G. Late Homework

Throughout the semester, assignments must be submitted as specified by the announced deadline. Overdue assignments will not be accepted.

Unexpected situations must be reported to the instructor for late homeworks by students.

H. Course Platform and Communication

Google Classroom and Github will be used as a course learning management system. All electronic resources and announcements about the course will be shared on this platform. It is very important to check the course page daily, access the necessary resources and announcements, and communicate with the instructor to complete the course with success.

I. Academic Integrity, Plagiarism & Cheating

Academic Integrity is one of the most important principles of RTEÜ University. Anyone who breaches the principles of academic honesty is severely punished.

It is natural to interact with classmates and others to "study together". It may also be the case where a student asks to help from someone else, paid or unpaid, better understand a difficult topic or a whole course. However, what is the borderline between "studying together" or "taking private lessons" and "academic dishonesty"? When is it plagiarism, when is it cheating?

It is obvious that looking at another student's paper or any source other than what is allowed during the exam is cheating and will be punished. However, it is known that many students come to university with very little experience concerning what is acceptable and what counts as "copying", especially for assignments.

The following are attempted as guidelines for the Faculty of Engineering and Architecture students to highlight the philosophy of academic honesty for assignments for which the student will be graded. Should a situation arise which is not described below, the student is advised to ask the instructor or assistant of the course whether what they intend to do would remain within the framework of academic honesty or not.

a. What is acceptable when preparing an assignment?

- Communicating with classmates about the assignment to understand it better
- Putting ideas, quotes, paragraphs, small pieces of code (snippets) that you find online or elsewhere into your assignment, provided that
 - these are not themselves the whole solution to the assignment,
 - you cite the origins of these
- Asking sources for help in guiding you for the English language content of your assignment.
- Sharing small pieces of your assignment in the classroom to create a class discussion on some controversial topics.
- Turning to the web or elsewhere for instructions, for references, and solutions to technical difficulties, but not for direct answers to the assignment
- Discussing solutions to assignments with others using diagrams or summarized statements but not actual text or code.
- Working with (and even paying) a tutor to help you with the course, provided the tutor does not do your assignment for you.

b. What is not acceptable?

- Asking a classmate to see their solution to a problem before submitting your own.
- Failing to cite the origins of any text (or code for programming courses) that you discover outside of the course's lessons and integrate into your work

- Giving or showing a classmate your solution to a problem when the classmate is struggling to solve it.

J. Expectations

You are expected to attend classes on time by completing weekly course requirements (readings and assignments) during the semester. The main communication channel between the instructor and the students will be emailed. Please send your questions to the instructor's email address about the course via the email address provided to you by the university. **Ensure that you include the course name in the subject field of your message and your name in the text field.** In addition, the instructor will contact you via email if necessary. For this reason, it is very important to check your email address every day for healthy communication.

K. Lecture Content and Syllabus Updates

If deemed necessary, changes in the lecture content or course schedule can be made. If any changes are made in the scope of this document, the instructor will inform you about this.

Course Schedule Overview

Weeks	Dates	Subjects	Other Tasks
Week 1	21.02.2022 24.02.2022	Course Plan and Communication Grading System, Assignments, and Exams. Software and Software Engineering Object Orientation and OOP with Java Part-I (Classes, Objects, Methods, Inheritance, Access Modifiers, This and InstanceOf Keywords)	TBD
Week 2	28.02.2022 03.03.2022	OOP with Java Part-II (super keyword, final keyword, Polymorphism / Encapsulation, Method Overriding, Nested Inner Class, Static Class, Anonymous Class, Enums / Enum-Constructor / Enum-String, Abstract Class, Object Class, Forms of Inheritance, Benefits, and Costs of Inheritance, Packages, Access Protection in Packages)	TBD
Week 3	07.03.2022 10.03.2022	OOP with Java Part-III (Defining and Interface and Interface Implementation, Nested Interfaces, Variables in Interfaces, Extending Interfaces, Reflection, Wrapper Classes, Lambda Notation)	TBD
Week 4	14.03.2022 17.03.2022	UML (UML tools, UML building blocks, architecture, diagrams, relationship, association, aggregation, composition comparison, association, dependency, generalization, realization, class diagram, object diagram, component diagram, deployment diagram, interaction diagram, use-case diagram, sequence)	TBD

		diagram, collaboration diagram, state-machine diagram, activity diagram, timing diagram)	
Week 5	21.03.2022 24.03.2022	Plantuml and Examples	TBD
Week 6	28.03.2022 31.03.2022	UMPLE and Examples	TBD
Week 7	04.04.2022 07.04.2022	UMPLE and Examples	TBD
Week 8	09.04.2022 17.04.2022	Midterm	TBD
Week 9	18.04.2022 21.04.2022	Using Design Patterns	TBD
Week 10	25.04.2022 28.04.2022	Using Design Patterns	TBD
Week 11	02.05.2022 05.05.2022	Holiday	TBD
Week 12	09.05.2022 12.05.2022	UML + UMPLE + Java Implementations	TBD
Week 13	16.05.2022 19.05.2022	SMC Remote Service Case Study Part-I	TBD
Week 14	23.05.2022 26.05.2022	SMC Remote Service Case Study Part-II	TBD
Week 15	30.05.2022 02.06.2022	OOP ATM Case Study Part-I	TBD
Week 16	06.06.2022 09.06.2022	OOP ATM Case Study Part-II	TBD
Week 17	13.06.2022 26.06.2022	Final	TBD

<https://www.webagesolutions.com/courses/WA1057-object-oriented-analysis-and-design-with-uml-and-java>

<https://www.unf.edu/~broggio/cen6016/>

Course Schedule Details

Week-1

1. Course Plan and Communication
2. Grading System, Home works, and Exams.
3. TBD
4. LlosengCh01E2v5.ppt Chapter 1: Software and Software Engineering
5. LlosengCh02E2v5.ppt Chapter 2: Review of Object Orientation
6. LlosengCh03E2v5.ppt Chapter 3: Basing Software Development on Reusable Technology
7. <http://www.btechsmartclass.com/java/java-oop-concepts.html>
 - a. [https://medium.com/javascript-scene/the-forgotten-history-of-oop-88d71b9b2d9f#:~:text=%E2%80%9CObject%2DOriented%20Programming%E2%80%9D%20\(his%20Sketchpad%20Thesis%20in%201963.](https://medium.com/javascript-scene/the-forgotten-history-of-oop-88d71b9b2d9f#:~:text=%E2%80%9CObject%2DOriented%20Programming%E2%80%9D%20(his%20Sketchpad%20Thesis%20in%201963.)
8. <http://www.btechsmartclass.com/java/java-buzz-words.html>
 - a. <https://www.javatpoint.com/cpp-vs-java>
9. <http://www.btechsmartclass.com/java/java-classes.html>
 - a. <https://www.programiz.com/java-programming/class-objects>
10. <http://www.btechsmartclass.com/java/java-methods-and-classes.html>
 - a. <https://www.programiz.com/java-programming/methods>
 - b. <https://www.programiz.com/java-programming/method-overloading>
 - c. <https://www.programiz.com/java-programming/constructors>
11. <http://www.btechsmartclass.com/java/java-inheritance-basics.html>
12. <http://www.btechsmartclass.com/java/java-access-specifiers.html>
 - a. <https://www.programiz.com/java-programming/access-modifiers>
13. <http://www.btechsmartclass.com/java/java-constructors-in-inheritance.html>
 - a. <https://www.programiz.com/java-programming/inheritance>
14. <https://www.programiz.com/java-programming/this-keyword>
15. <https://www.programiz.com/java-programming/instanceof>

Week-2

1. <http://www.btechsmartclass.com/java/java-super-keyword.html>
 - a. <https://www.programiz.com/java-programming/super-keyword>
2. <http://www.btechsmartclass.com/java/java-final-keyword.html>
 - a. <https://www.programiz.com/java-programming/final-keyword>
3. <http://www.btechsmartclass.com/java/java-polymorphism.html>
 - a. <https://www.programiz.com/java-programming/polymorphism>
 - b. <https://www.programiz.com/java-programming/encapsulation>
4. <http://www.btechsmartclass.com/java/java-method-overriding.html>
 - a. <https://www.programiz.com/java-programming/method-overriding>
5. <https://www.programiz.com/java-programming/nested-inner-class>
6. <https://www.programiz.com/java-programming/static-class>
7. <https://www.programiz.com/java-programming/anonymous-class>
8. <https://www.programiz.com/java-programming/enums>
9. <https://www.programiz.com/java-programming/enum-constructor>
10. <https://www.programiz.com/java-programming/enum-string>

11. <http://www.btechsmartclass.com/java/java-abstract-class.html>
 - a. <https://www.programiz.com/java-programming/abstract-classes-methods>
12. <http://www.btechsmartclass.com/java/java-Object-class.html>
13. <http://www.btechsmartclass.com/java/java-forms-of-inheritance.html>
 - a. <https://www.programiz.com/java-programming/interfaces>
14. <http://www.btechsmartclass.com/java/java-benefits-and-costs-of-inheritance.html>
15. <http://www.btechsmartclass.com/java/java-defining-packages.html>
16. <http://www.btechsmartclass.com/java/java-access-protection-in-packages.html>
17. <http://www.btechsmartclass.com/java/java-importing-packages.html>

Week-3

1. <http://www.btechsmartclass.com/java/java-defining-an-interface.html>
2. <http://www.btechsmartclass.com/java/java-implementing-an-interface.html>
3. <http://www.btechsmartclass.com/java/java-nested-interfaces.html>
4. <http://www.btechsmartclass.com/java/java-variables-in-interfaces.html>
5. <http://www.btechsmartclass.com/java/java-extending-an-interface.html>
6. <https://www.programiz.com/java-programming/reflection>
7. https://www.w3schools.com/java/java_wrapper_classes.asp
8. https://www.w3schools.com/java/java_lambda.asp

Week-4

- UML
 - o <https://www.guru99.com/best-uml-tools.html>
 - o <https://www.altova.com/umodel>
 - o <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>
 - o <https://www.javatpoint.com/uml>
 - o <https://www.javatpoint.com/uml-building-blocks>
 - o <https://www.javatpoint.com/uml-architecture>
 - o <https://www.javatpoint.com/uml-diagrams>
 - o <https://www.javatpoint.com/uml-relationship>
 - o [https://www.javatpoint.com/uml-Association-vs-aggregation-vs-composition](https://www.javatpoint.com/uml-association-vs-aggregation-vs-composition)
 - o <https://www.javatpoint.com/uml-association>
 - o <https://www.javatpoint.com/uml-dependency>
 - o <https://www.javatpoint.com/uml-generalization>
 - o <https://www.javatpoint.com/uml-realization>
 - o <https://www.javatpoint.com/uml-class-diagram>
 - o <https://www.javatpoint.com/uml-object-diagram>
 - o <https://www.javatpoint.com/uml-component-diagram>
 - o <https://www.javatpoint.com/uml-deployment-diagram>
 - o <https://www.javatpoint.com/uml-interaction-diagram>
 - o <https://www.javatpoint.com/uml-use-case-diagram>
 - o <https://www.javatpoint.com/uml-sequence-diagram>
 - o <https://www.javatpoint.com/uml-collaboration-diagram>
 - o <https://www.javatpoint.com/uml-state-machine-diagram>
 - o <https://www.javatpoint.com/uml-activity-diagram>
 - o <https://www.javatpoint.com/uml-timing-diagram>

- <https://www.javatpoint.com/uml-tools>

Week-5

Week-6

- Plantuml

- <https://plantuml.com/class-diagram>

- Uml

- [~ https://marketplace.visualstudio.com/items?itemName=digized.uml](https://marketplace.visualstudio.com/items?itemName=digized.uml)
- [~ https://cruise.uml.org/uml/](https://cruise.uml.org/uml/)
- [~ https://github.com/uml/Uml/wiki/examples](https://github.com/uml/Uml/wiki/examples)
- [~ https://github.com/uml/uml/tree/master/examples](https://github.com/uml/uml/tree/master/examples)
- [~ https://github.com/uml/uml/tree/master/externalexamples](https://github.com/uml/uml/tree/master/externalexamples)
- [~ https://github.com/TimLethbridge/Lloseng](https://github.com/TimLethbridge/Lloseng)
- [~ https://cruise.uml.org/umleonline/download_uml.shtml](https://cruise.uml.org/umleonline/download_uml.shtml)

Week-7

- Uml Examples

Week-8 (Midterm)

Week-9

1. LlosengCh05E2v5.ppt Chapter 5: Modelling with Classes
2. LlosengCh06E2v5.ppt Chapter 6: Using Design Patterns
3. LlosengCh08E2v5.ppt Chapter 8: Modelling Interactions and Behaviour
4. LlosengCh09E2v5.ppt Chapter 9: Architecting and Designing Software

1. Principles of OOD Chapter-6

2. The Practices: dX Chapter-7

Week-10

3. Packages Chapter- Chapter 8

4. Object Diagrams- Chapter 9

5. State Diagrams- Chapter 10

Week-11

1. Heuristics and Coffee Chapter 11

Week-12

2. SMC Remote Service: Case Study Chapter 12

Week-13

3. SMC Remote Service: Case Study Chapter 12

Week-14

08_KBU_OOP_AtCaseStudy-I.pdf

Week-15

08_KBU_OOP_AtCaseStudy-II.pdf

Week-16 (Final)