Recep Tayyip Erdogan University

Faculty of Engineering and Architecture

Computer Engineering

CE103 Algorithms and Programming-I

Syllabus

Fall Semester, 2023-2024

Download DOC, SLIDE, PPTX

Instructor	Asst. Prof. Dr. Uğur CORUH
Contact Information	ugur.coruh@erdogan.edu.tr
Office No	F-301
Google Classroom Code	TBD
Microsoft Teams Code	etj1k7b
Lecture Hours and Days	Wednesday 09:00 - 12:00 (Theory) / Thursday 13:00 - 14:30 (Lab)

Lecture Classroom	İİBF-414 (Level-4)
Office Hours	Scheduled through your university account, meetings will take place via Google Meet and are coordinated by request through email. To expedite responses, please initiate your email subject line with the [CE103] tag and maintain a formal, concise, and clear email body.

Lecture and Communication Language	**English**
Theory/Laboratory Course Hour Per Week	3/2 Hours
Credit	4
Prerequisite	TBD
Corequisite	TBD
Requirement	TBD

*TBD: To Be Defined.

A. Course Description

The objective of this course is to systematically build foundational skills in algorithms and programming, aimed at enhancing students' career prospects. The instructional approach focuses on the transfer of expert knowledge while guiding students in identifying effective learning techniques and practical applications within the realm of algorithms and programming. Emphasis is placed on hands-on projects and applications, fortifying the learning experience through practice over pure theory. The course covers functional programming in C, C++, C#, and Java, utilizing the latest development environments.

B. Course Learning Outcomes

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

- Understand a software developer's road map and qualifications.

- Use different types of development environments to build applications.

- Understand the relation between real-life problems and their programming practices.

- Use language features in C, C++, C#, and Java for functional programming and evaluate their relative benefits.

- Understand application generation flows and outputs in detail, such as binaries and executables.

- Use the source code, version management systems, and portals based on GIT

- Work on the remote systems with remote connection tools.

- Use common developer tools that help application developers

- Create application libraries such as static, shared libraries for code reusability and functional packaging.

- Create unit tests for their applications to automate tests for their algorithms.

- Create console and GUI-based applications for their solutions.

- Create documentation for their applications.

C. Course Topics

- Developer Road Map

- Algorithm Design and Basics

- Basic Operating System Information for Development Requirement

- Basic Remote Connection and Working Know-How

- Source Code Version Management Systems (GIT)

- Integrated Development Environments

- Application Test Automation

- Application Debugging and Bugfixing

- Functional Programming (C,C++, C#, Java)

- Continues Integration and Continues Development Processes

- Software Development Principles

- Application Documentation Automation

- Shared and Static Library Development and Test in Cross-Environment

D. Textbooks and Required Hardware or Equipment

This course does not require a coursebook.

If necessary, you can use the following books and open-source online resources.

- *Paul Deitel and Harvey Deitel. 2012. C How to Program (7th. ed.). Prentice Hall Press, USA.*

- *Intro to Java Programming, Comprehensive Version (10th Edition) 10th Edition by Y. Daniel Liang*

- *Introduction to Algorithms, Third Edition By Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein*

- *Problem Solving and Program Design in C, J.R. Hanly, and E.B. Koffman, 6th Edition.*

- *Robert Sedgewick and Kevin Wayne. 2011. Algorithms (4th. ed.). Addison-Wesley Professional.*

- *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.
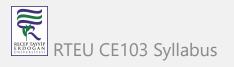
E. Grading System

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

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

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

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

$$Grade = (a_1 HW_1 + a_2 HW_2 + ... + a_n HW_n)/n$$

Homework	Weight
Midterm	%40
Final	%60

$$Passing\ Grade = (40 * Midterm_{Grade} + 60 * Final_{Grade})/100$$

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. Your grade will be reduced by 10% of the full points for each calendar day for overdue assignments.

Overdue assignments will not be accepted after three (3) days.

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

H. Course Platform and Communication

Google Classroom 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 as you needed 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, references, and solutions to technical difficulties, but not for direct answers to the assignment

- Discuss 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?

- Ask 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.

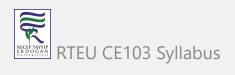## K. Lecture Content and Syllabus Updates

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	04.10.2023 05.10.2023	Course Plan and Communication,Grading System, Assignments, and Exams,Computer Engineering Job Qualifications and Road Map,Google Search Basics,Programming Introduction (Operating System Basics, Computer Network Basics, Numerical System Basics, Character Sets)	TBD
Week 2	11.10.2023 12.10.2023	Algorithm Basics, Flowgorithm, Pseudocode, Programming Environment Setup and Configuration for C, C++, Java, and C#, Common Developer Tools	TBD

Week 3	18.10.2023 19.10.2023	**Source Code Sharing and Version Management.**	**TBD**
Week 4	25.10.2023 26.10.2023	Shared Library Development and Application Test Automation for C, C++, C# and Java, TDD (Test Driven Development)	TBD
Week 5	01.11.2023 02.11.2023	C Functional Console Programming	

Week-6	08.11.2023 09.11.2023	C++ Functional Console Programming	TBD
Week-7	15.11.2023 16.11.2023	C# Functional Console Programming	TBD
Week-8	TBD	Midterm	TBD

Week-9	29.11.2023 30.11.2023	Java Functional Console Programming-I	TBD
Week-10	06.11.2023 07.11.2023	Java Functional Console Programming-II	TBD
Week-11	13.11.2023 14.11.2023	Java Functional Console Programming-III	TBD

Week-12	20.11.2023 21.11.2023	**C / C++ Graphical User Interface (GUI) Programming**	**TBD**
Week-13	27.11.2023 28.11.2023	C# Graphical User Interface (GUI) Programming-I	TBD
Week-14	03.12.2023 04.12.2023	C# Graphical User Interface (GUI) Programming-II	TBD

Week-15	10.12.2023 11.12.2023	Java Graphical User Interface Programming	TBD
Week-16	TBD	Final	TBD

Bologna Information

$$End - Of - CE103 - Syllabus$$