Recep Tayyip Erdogan University

Faculty of Engineering and Architecture

Computer Engineering

CE103- Algorithms and Programming I

Syllabus

Fall Semester, 2021-2022

Instructor	**Asst. Prof. Dr. Uğur CORUH**
Contact Information	ugur.coruh@erdogan.edu.tr
Office Number	**F-301**
Google Classroom Code	**3ipdtws**
Lecture Hours and Days	**Monday 13:00-15:30 (Theory)** **Wednesday 13:00-14:30 (Lab)**
Lecture Classroom	**F-119**
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 [CE103] 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	Not Exist
Corequisite	TBD
Requirement	CE100 Algorithms and Programming-II* (1-2) CE205 Data Structures* (2-3) CE204 Object Oriented Programming* (2-4) CE208 Database Management Systems* (2-4) CE303 Web and Mobile Programming* (3-5) ELCE05 CE307 Graph Theory* CE309 Fuzzy Logic* CE311 Robotics*

	CE313 System and Software Analysis and Design* SECCE05 CE315 Matlab Programlama CE317 Kriptografi CE319 Biçimsel Diller ve Otomata Teorisi ELCE06 CE306 Cyber Security* CE308 Detection and Estimation* CE310 Formal Methods for Software Engineering* CE312 Parallel Programming* CE314 Image - Video Processing* SECCE06 CE316 Yapay Zeka CE318 Finansal Teknolojiler ELCE07 CE403 Network Security* CE405 Data Mining* SECCE07 CE407 Güvenli Programlama CE409 Derin Öğrenme ELCE08 CE402 Software Defined Networking* CE404 Fuzzy Systems* CE406 PostQuantum Cryptography* SECCE08 CE408 Dağıtık Sistemler ve Güvenlik

**TBD: To Be Defined.*

A. Course Description

This course goal is to develop algorithm and programming expertise from scratch in a powerful way to provide a high-quality career path for students. The lecture will be based on expertise sharing and guiding students to find learning methods and practice for algorithm and programming topics. By making programming applications and projects in the courses, the learning process will be strengthened by practicing rather than theory. This course provides functional programming for C, C++, C#, and Java with up-to-date 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.

$$a_n = Homework\ or\ Project\ Weight$$

$$HW_n = Homework\ or\ Project\ Points$$

$$n = Number\ of\ Homework\ or\ Project$$

$$Grade = \frac{(a_1 HW_1 + a_2 HW_2 + \cdots + a_n HW_n)}{n}$$

Homework/Exam	Weight
$Midterm$	$\%40$
$Final$	$\%60$

$$Passing\ Grade = \frac{40 * Midterm_{Grade} + 60 * Final_{Grade}}{100}$$

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, 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	Details
Week 1	04.10.2021 06.10.2021	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	11.10.2021 13.10.2021	Algorithm Basics, Flowgorithm, Pseudocode Programming Environment Setup and Configuration for C, C++, Java, and C# Common Developer Tools Online Programming Envoriments	TBD
Week 3	18.10.2021 20.10.2021	Source Code Sharing and Version Management	TBD
Week 4	25.10.2021 27.10.2021	Shared Library Development and Application Test Automation for C, C++, C# and Java TDD (Test Driven Development)	TBD
Week 5	01.11.2021 03.11.2021	C Functional Console Programming	TBD
Week 6	08.11.2021 10.11.2021	C++ Functional Console Programming	TBD
Week 7	15.11.2021 17.11.2021	C# Functional Console Programming	TBD
Week 8	**20.11.2021 28.11.2021**	**Midterm**	
Week 9	29.11.2021 01.12.2021	Java Functional Console Programming-I	TBD

Week 10	06.12.2021 08.12.2021	Java Functional Console Programming-II	TBD
Week 11	13.12.2021 15.12.2021	Java Functional Console Programming-III	TBD
Week 12	20.12.2021 22.12.2021	C / C++ Graphical User Interface (GUI) Programming	TBD
Week 13	27.12.2021 29.12.2021	C# Graphical User Interface (GUI) Programming	TBD
Week 14	03.01.2022 05.01.2022	C# Graphical User Interface (GUI) Programming	TBD
Week 15	10.01.2022 12.01.2022	Java Graphical User Interface Programming	TBD
Week 16	**17.01.2022** **30.01.2022**	**Final**	

Course Schedule Details

Note: Red-marked topics are discarded for this course.

Week-1 (Introduction to Computer Systems)

1. Course Plan and Communication
2. Grading System, Home works, and Exams.
3. Computer Engineering Job Qualifications
4. Developer Road Map
5. Using Google
6. Programming Introduction
    a. Operating System Basics
        i. Types of Operating Systems
        ii. Console commands
        iii. System folders
        iv. System parameters
        v. Storage management
    b. Computer Network Basics
        i. Network connections
        ii. Network related console commands (ipconfig, ipconfig /renew, ipconfig /release, hostname, netstat –a, nslookup)
        iii. IP, Port, DNS settings, NAT etc.
        iv. Remote connections (FTP, SSH, RDP, XRDP)
        v. Putty, Mobaxterm
    c. Numerical System Basics
        i. Binary system
        ii. Hexadecimal system
    d. Character Sets

Week-2 (Introducton to Algorithms and Development Envoriments)

1. Algorithm Basics
2. Introduction to Analysis of Algorithms

a. Algorithm Basics
b. Flowgorithm
c. Pseudocode

3. Programming Topics
    a. Programming Environment Setup and Configuration
        i. C / Cpp
            1. DevCpp
            2. Code Blocks
            3. GCC/G++ Compiler (Mingw) / Clang-cl (LLVM)
            4. vscode
            5. Visual Studio Community Edition
                a. Visual Studio x64 x86 Configurations and Features
                b. Project Types
            6. Notepad++
            7. Vi/Vim
            8. Eclipse
                a. Simple Java Project Generation
                b. Jar Export as Library or Executable
                c. Maven Project Generation
                d. Junit Test Case Generation and Testing
            9. Netbeans
            10. Turbo C
            11. Turbo C++
        ii. Java
            1. JDK, JRE Setup
            2. System Environments and Paths
            3. Netbeans
            4. Eclipse
            5. Intellij Idea (jetbrains)
            6. vscode
            7. Notepad++
        iii. C#
            1. Visual Studio Community Edition
            2. Notepad++
    b. Programming Environment Setup and Configuration
        i. Notepad++ (Notepad for Source Code)
        ii. HxD (Hex Editor)
        iii. Marktext (Markdown Syntax Editor)
        iv. Cygwin (Linux environment for Windows)
        v. Dependency Walker (32-bit or 64-bit Windows module dependency checker)
        vi. Doxygen (Code Documentation)
        vii. Sonarlint (Code Quality and Code Security Extension)
        viii. Codepen.io (online code sharing)
        ix. Codeshare.io (real time code sharing)
        x. Codebeautify.org (online data conversion tools)
        xi. AsciiFlow.com (ASCII drawing tool)
        xii. Freemind (opensource mindmap application)
        xiii. Wireflow (user flow designer)

xiv. PlantUML (software designer)
xv. Drawio (drawing tool)
xvi. Putty (Remote Connection)
xvii. MobaXterm (Remote Connection)
xviii. Teamviewer (Remote Connection)
xix. Paletton.com (Color Chooser)
xx. Understand (Static Code Analysis)
xxi. JD Project (Java Decompiler)
xxii. Cutter (Multi-Platform Reverse Engineering Tool)
xxiii. IDA Pro / Freeware (Native Reverse Engineering Tool)
xxiv. Travis-CI
    1. Travis.yml
xxv. Jenkins
xxvi. Valgrind
xxvii. Docker
    1. Dockerfile
    2. DockerHub
    3. Docker Compose Yaml
    4. Dockerrun.aws.json (AWS)
xxviii. Nuget Packages
xxix. Extras
    1. vim/vim-wim32-installer (windows vim installer)
xxx. SCV Cryptomanager
xxxi. Addario CryptoBench
xxxii. Raymond's MD5 & SHA Checksum Utility
xxxiii. SlavaSoft HashCalc
xxxiv. Portable PGP

c. Online Programming Envoriments
    i. Hackerrank
    ii. CS50 Sandbox
    iii. Programiz C Online Complier

Week-3 (Introduction to Source Code Version Management)

1. Programming Source Code Sharing and Version Management
    a. Introduction to Source Code Management Systems
    b. Features of Source Code Management
    c. Why Do We Need Source Code Management?
    d. Types of Version Control Systems
        i. Centralized
        ii. Distributed
    e. Git Usage
        i. Installation of Git
            1. Git
            2. Git-Extension
            3. Gig (git ignore creator)
        ii. Configuration of Git
        iii. Github Platform Usage
        iv. Create a New Repository
        v. Checkout a Repository

vi. Add & Commit (Write Good Commits)
vii. Pushing Changes
    1. Update Local Repo Before Sending
    2. Send Changes to Remote Repo
viii. Branching
ix. Update & Merge
    1. Fast-forward (-ff) Merging
    2. No-fast-forward (--no-ff) Merging
    3. Merge Conflicts
x. Rebasing
xi. Replace Local Changes / Resetting
    1. Soft reset
    2. Hard reset
    3. Reverting
xii. Cherry-picking
xiii. Fetching
xiv. Reflog
xv. Tagging
xvi. Log
xvii. Gource

f. Maven Usage
g. TFS Usage
h. SVN Usage

Week-4 (Introduction to Code Reusability and Automated Testing)

1. Shared Library Development
    a. C Programming
    b. Cpp Programming
    c. Csharp Programming
    d. Java Programming
2. Program Testing
3. Unit Test Development
    a. C
    b. Cpp
    c. Csharp
    d. Java
4. TDD (Test Driven Development)
5. Test and Deployment Automation Management

Week-5 (C Functional Console Programming)

1. Programming Development
    i. Debugging
    ii. Console Application Development
        1. C Programming
            a. C Introduction
                i. Keywords & Identifiers
                ii. Variables & Constants
                iii. C Data Types
                iv. C Input/Output

v. C Operators
vi. C Introduction Examples (homework)

b. C Flow Control
   i. C if..else
   ii. C for loop
   iii. C while loop
   iv. C break and continue
   v. C switch...case
   vi. C Programming goto
   vii. Control Flow Examples (homework)

c. C Functions
   i. C Programming Functions
   ii. C User-defined Functions
   iii. C Function Types
   iv. C Recursion
   v. C Storage Class
   vi. C Function Examples

d. C Programming Arrays
   i. C Programming Arrays
   ii. C Multi-dimensional Arrays
   iii. C Arrays & Functions

e. C Programming Pointers
   i. C Programming Pointers
   ii. C Pointers & Arrays
   iii. C Pointers and Functions
   iv. C Memory Allocation
   v. Array & Pointer Examples

f. C Programming Strings
   i. C Programming Strings
   ii. C String Functions
   iii. C String Examples

g. C Structure and Union
   i. C Structure
   ii. C Struct & Pointers
   iii. C Struct & Functions
   iv. C Unions
   v. C Struct Examples

h. C Programming Files
   i. C Files Input/Output
   ii. C Files Examples

i. Additional Topics
   i. C Enumeration
   ii. C Preprocessors
   iii. C Standard Library
   iv. C Programming Examples

Week-6 (Cpp Functional Console Programming)

1. Cpp Programming
   a. C++ Introduction

i. C++ Variables and Literals
ii. C++ Data Types
iii. C++ Basic I/O
iv. C++ Type Conversion
    1. C++ String to Int and Vice-Versa
    2. C++ String to Float, Double and Vice-Versa
v. C++ Operators
vi. C++ Comments

b. C++ Flow Control
i. C++ if..else
ii. C++ for loop
iii. C++ do..while loop
iv. C++ break statement
v. C++ continue statement
vi. C++ switch statement
vii. C++ goto statement

c. C++ Functions
i. C++ Functions
ii. C++ Function Types
iii. C++ Function Overloading
iv. C++ Default Argument
v. C++ Storage Class
vi. C++ Recursion
vii. C++ Return Reference

d. C++ Arrays & String
i. C++ Arrays
ii. Multidimensional Arrays
iii. C++ Function and Array
iv. C++ String

e. C++ Structures
i. C++ Structures
ii. Structure and Function
iii. C++ Pointers to Structure
iv. C++ Enumeration

f. C++ Object & Class
i. C++ Objects and Class
ii. C++ Constructors
iii. C++ Objects & Function
iv. C++ Operator Overloading

g. C++ Pointers
i. C++ Pointer
ii. C++ Pointers and Arrays
iii. C++ Pointers and Functions
iv. C++ Memory Management

h. C++ Inheritance
i. C++ Inheritance
ii. Inheritance Access Control
iii. C++ Function Overriding
iv. Multiple & Multilevel Inheritance

v. C++ Friend Function
vi. C++ Virtual Function
vii. C++ Templates

Week-7 (Csharp Functional Console Programming)

1. Csharp Programming
    a. Introduction
        i. C# Hello World
        ii. C# Keywords & Identifiers
        iii. C# Variables
        iv. C# Operators
        v. C# Basic I/O
        vi. C# Expressions & Statements
        vii. C# Comments
    b. Flow Control
        i. C# if..else
        ii. C# for loop
        iii. C# while loop
        iv. C# foreach loop
        v. C# switch statement
        vi. C# ternary operator
    c. Exception Handling
    d. Other Topics
        i. C# Bitwise Operators
        ii. C# Preprocessor Directives
        iii. C# Namespaces
        iv. C# Partial Class & Method

Week-8 (Midterm)

Week-9 (Java Functional Console Programming)

1. Java Programming
    a. Java Introduction
        i. Java Hello World
        ii. Java JVM, JRE, and JDK
        iii. Java Variables
        iv. Java Data Types
        v. Java Operators
        vi. Java Input and Output
        vii. Java Expressions & Blocks
        viii. Java Comment
    b. Java Flow Control
        i. Java if..else
        ii. Java switch statement
        iii. Java for loop
        iv. Java for-each loop
        v. Java while loop
        vi. Java break statement
        vii. Java continue statement

c. Java Arrays
    i. Java Arrays
    ii. Multidimensional Array
    iii. Java Copy Array

Week-10 (Java Functional Console Programming)

a. Java OOP-I
    i. Java Class and Objects
    ii. Java Methods
    iii. Java Constructor
    iv. Java Strings
    v. Java Access Modifiers
    vi. Java this keyword
    vii. Java final keyword
    viii. Java recursion
    ix. Java instanceof operator
b. Java OOP-II
    i. Java Inheritance
    ii. Java Method Overriding
    iii. Java super keyword
    iv. Abstract Class & Method
    v. Java Interfaces
    vi. Java Polymorphism
    vii. Java Encapsulation
c. Java OOP-III
    i. Nested & Inner Class
    ii. Java Static Class
    iii. Java Anonymous Class
    iv. Java Singleton
    v. Java enum class
    vi. Java enum constructor
    vii. Java enum string
    viii. Java reflection
d. Java Exception Handling
    i. Java Exceptions
    ii. Java Exception Handling
    iii. Java try..catch
    iv. Java throw and throws
    v. Java catch Multiple Exceptions
    vi. Java try-with-resources
    vii. Java Annotations
    viii. Java Annotation Types
    ix. Java Logging
    x. Java Assertions
e. Java List
    i. Java Collection Framework
    ii. Java Collection Interface
    iii. Java List Interface
    iv. Java ArrayList

v. Java Vector
vi. Java Stack

f. Java Queue
i. Java Queue Interface
ii. Java PriorityQueue Interface
iii. Java Deque Interface
iv. Java LinkedList
v. Java ArrayDeque
vi. Java BlockingQueue Interface
vii. Java ArrayBlockingQueue
viii. Java LinkedBlocking Queue

g. Java Map
i. Java Map Interface
ii. Java HashMap
iii. Java LinkedHashMap
iv. Java WeakHashMap
v. Java EnumMap
vi. Java SortedMap Interface
vii. Java NavigableMap Interface
viii. Java TreeMap
ix. Java ConcurrentMap Interface
x. Java ConcurrentHashMap

h. Java Set
i. Java Set Interface
ii. Java HashSet
iii. Java EnumSet
iv. Java LinkedHashSet
v. Java SortedSet Interface
vi. Java NavigableSet Interface
vii. Java TreeSet
viii. Java Algorithms
ix. Java Iterator
x. Java ListIterator

Week-11 (Java Functional Console Programming)

i. Java I/O Streams
i. Java I/O Streams
ii. Java InputStream
iii. Java OutputStream
iv. Java FileInputStream
v. Java FileOutputStream
vi. Java ByteArrayInputStream
vii. Java ByteArrayOutputStream
viii. Java ObjectInputStream
ix. Java ObjectOutputStream
x. Java BufferedInputStream
xi. Java BufferedOutputStream
xii. Java PrintStream

j. Java Reader/Writer

i. Java Reader
ii. Java Writer
iii. Java InputStreamReader
iv. Java OutputStreamWriter
v. Java FileReader
vi. Java FileWriter
vii. Java BufferedReader
viii. Java BufferedWriter
ix. Java StringReader
x. Java StringWriter
xi. Java PrintWriter

k. Additional Topics
i. Java Scanner Class
ii. Java Type Casting
iii. Java autoboxing and unboxing
iv. Java Lambda Expression
v. Java Generics
vi. Java File Class
vii. Java Wrapper Class
viii. Java Command Line Arguments
ix. JNLP (Java Network Launch Protocol)

Week-12 (C/Cpp GUI Programming)

6. GUI Application Development (Windows)
    a. C Programming
    b. Cpp Programming

Week-13 (Csharp GUI Programming)

7. GUI Application Development (Windows)
    a. Csharp Programming

Week-14 (Csharp GUI Programming)

8. GUI Application Development (Windows)
    a. Csharp Programming

Week-15 (Java GUI Programming)

9. GUI Application Development (Windows)
    a. Java Programming

Week-16 (Final)