

**Recep Tayyip Erdogan University**

**Faculty of Engineering and Architecture**

**Computer Engineering**

CE103- Algorithms and Programming I

**Syllabus**

**Fall Semester, 2022-2023**

|  |  |
| --- | --- |
| Instructor | Asst. Prof. Dr. Uğur CORUH |
| Contact Information | [ugur.coruh@erdogan.edu.tr](mailto: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.*

1. **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.

1. **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.

1. **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

1. **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.

1. **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.

|  |  |
| --- | --- |
| **Homework/Exam** | **Weight** |
|  |  |
|  |  |

1. **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.

1. **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.

1. **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.

1. **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.

* 1. **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.
  2. **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.

1. **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.

1. **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** | **20.11.202128.11.2021** | **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
   1. Operating System Basics
      1. Types of Operating Systems
      2. Console commands
      3. System folders
      4. System parameters
      5. Storage management
   2. Computer Network Basics
      1. Network connections
      2. Network related console commands (ipconfig, ipconfig /renew, ipconfig /release, hostname, netstat –a, nslookup)
      3. IP, Port, DNS settings, NAT etc.
      4. Remote connections (FTP, SSH, RDP, XRDP)
      5. Putty, Mobaxterm
   3. Numerical System Basics
      1. Binary system
      2. Hexadecimal system
   4. Character Sets

**Week-2 (Introducton to Algorithms and Development Envoriments)**

1. Algorithm Basics
2. Introduction to Analysis of Algorithms
   1. Algorithm Basics
   2. Flowgorithm
   3. Pseudocode
3. Programming Topics
   1. Programming Environment Setup and Configuration
      1. C / Cpp
         1. DevCpp
         2. Code Blocks
         3. GCC/G++ Compiler (Mingw) / Clang-cl (LLVM)
         4. vscode
         5. Visual Studio Community Edition
            1. Visual Studio x64 x86 Configurations and Features
            2. Project Types
         6. Notepad++
         7. Vi/Vim
         8. Eclipse
            1. Simple Java Project Generation
            2. Jar Export as Library or Executable
            3. Maven Project Generation
            4. Junit Test Case Generation and Testing
         9. Netbeans
         10. Turbo C
         11. Turbo C++
      2. Java
         1. JDK, JRE Setup
         2. System Environments and Paths
         3. Netbeans
         4. Eclipse
         5. Intellij Idea (jetbrains)
         6. vscode
         7. Notepad++
      3. C#
         1. Visual Studio Community Edition
         2. Notepad++
   2. Programming Environment Setup and Configuration
      1. Notepad++ (Notepad for Source Code)
      2. HxD (Hex Editor)
      3. Marktext (Markdown Syntax Editor)
      4. Cygwin (Linux environment for Windows)
      5. Dependency Walker (32-bit or 64-bit Windows module dependency checker)
      6. Doxygen (Code Documentation)
      7. Sonarlint (Code Quality and Code Security Extension)
      8. Codepen.io (online code sharing)
      9. Codeshare.io (real time code sharing)
      10. Codebeautify.org (online data conversion tools)
      11. AsciiFlow.com (ASCII drawing tool)
      12. Freemind (opensource mindmap application)
      13. Wireflow (user flow designer)
      14. PlantUML (software designer)
      15. Drawio (drawing tool)
      16. Putty (Remote Connection)
      17. MobaXterm (Remote Connection)
      18. Teamviewer (Remote Connection)
      19. Paletton.com (Color Chooser)
      20. Understand (Static Code Analysis)
      21. JD Project (Java Decompiler)
      22. Cutter (Multi-Platform Reverse Engineering Tool)
      23. IDA Pro / Freeware (Native Reverse Engineering Tool)
      24. Travis-CI
          1. Travis.yml
      25. Jenkins
      26. Valgrind
      27. Docker
          1. Dockerfile
          2. DockerHub
          3. Docker Compose Yaml
          4. Dockerrun.aws.json (AWS)
      28. Nuget Packages
      29. Extras
          1. vim/vim-wim32-installer (windows vim installer)
      30. SCV Cryptomanager
      31. Addario CryptoBench
      32. Raymond's MD5 & SHA Checksum Utility
      33. SlavaSoft HashCalc
      34. Portable PGP
   3. Online Programming Envoriments
      1. Hackerrank
      2. CS50 Sandbox
      3. Programiz C Online Complier

**Week-3 (Introduction to Source Code Version Management)**

1. Programming Source Code Sharing and Version Management
   1. Introduction to Source Code Management Systems
   2. Features of Source Code Management
   3. Why Do We Need Source Code Management?
   4. Types of Version Control Systems
      1. Centralized
      2. Distributed
   5. Git Usage
      1. Installation of Git
         1. Git
         2. Git-Extension
         3. Gig (git ignore creator)
      2. Configuration of Git
      3. Github Platform Usage
      4. Create a New Repository
      5. Checkout a Repository
      6. Add & Commit (Write Good Commits)
      7. Pushing Changes
         1. Update Local Repo Before Sending
         2. Send Changes to Remote Repo
      8. Branching
      9. Update & Merge
         1. Fast-forward (-ff) Merging
         2. No-fast-forward (--no-ff) Merging
         3. Merge Conflicts
      10. Rebasing
      11. Replace Local Changes / Resetting
          1. Soft reset
          2. Hard reset
          3. Reverting
      12. Cherry-picking
      13. Fetching
      14. Reflog
      15. Tagging
      16. Log
      17. Gource
   6. Maven Usage
   7. TFS Usage
   8. SVN Usage

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

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

**Week-5 ( C Functional Console Programming)**

1. Programming Development
   * 1. Debugging
     2. Console Application Development
        1. C Programming
           1. C Introduction

Keywords & Identifiers

Variables & Constants

C Data Types

C Input/Output

C Operators

C Introduction Examples (homework)

* + - * 1. C Flow Control

C if..else

C for loop

C while loop

C break and continue

C switch...case

C Programming goto

Control Flow Examples (homework)

* + - * 1. C Functions

C Programming Functions

C User-defined Functions

C Function Types

C Recursion

C Storage Class

C Function Examples

* + - * 1. C Programming Arrays

C Programming Arrays

C Multi-dimensional Arrays

C Arrays & Functions

* + - * 1. C Programming Pointers

C Programming Pointers

C Pointers & Arrays

C Pointers and Functions

C Memory Allocation

Array & Pointer Examples

* + - * 1. C Programming Strings

C Programming Strings

C String Functions

C String Examples

* + - * 1. C Structure and Union

C Structure

C Struct & Pointers

C Struct & Functions

C Unions

C Struct Examples

* + - * 1. C Programming Files

C Files Input/Output

C Files Examples

* + - * 1. Additional Topics

C Enumeration

C Preprocessors

C Standard Library

C Programming Examples

**Week-6 (Cpp Functional Console Programming)**

* + - 1. Cpp Programming
         1. C++ Introduction

C++ Variables and Literals

C++ Data Types

C++ Basic I/O

C++ Type Conversion

C++ String to Int and Vice-Versa

C++ String to Float, Double and Vice-Versa

C++ Operators

C++ Comments

* + - * 1. C++ Flow Control

C++ if..else

C++ for loop

C++ do..while loop

C++ break statement

C++ continue statement

C++ switch statement

C++ goto statement

* + - * 1. C++ Functions

C++ Functions

C++ Function Types

C++ Function Overloading

C++ Default Argument

C++ Storage Class

C++ Recursion

C++ Return Reference

* + - * 1. C++ Arrays & String

C++ Arrays

Multidimensional Arrays

C++ Function and Array

C++ String

* + - * 1. C++ Structures

C++ Structures

Structure and Function

C++ Pointers to Structure

C++ Enumeration

* + - * 1. C++ Object & Class

C++ Objects and Class

C++ Constructors

C++ Objects & Function

C++ Operator Overloading

* + - * 1. C++ Pointers

C++ Pointer

C++ Pointers and Arrays

C++ Pointers and Functions

C++ Memory Management

* + - * 1. C++ Inheritance

C++ Inheritance

Inheritance Access Control

C++ Function Overriding

Multiple & Multilevel Inheritance

C++ Friend Function

C++ Virtual Function

C++ Templates

**Week-7 (Csharp Functional Console Programming)**

1. Csharp Programming
   1. Introduction
      1. C# Hello World
      2. C# Keywords & Identifiers
      3. C# Variables
      4. C# Operators
      5. C# Basic I/O
      6. C# Expressions & Statements
      7. C# Comments
   2. Flow Control
      1. C# if..else
      2. C# for loop
      3. C# while loop
      4. C# foreach loop
      5. C# switch statement
      6. C# ternary operator
   3. Exception Handling
   4. Other Topics
      1. C# Bitwise Operators
      2. C# Preprocessor Directives
      3. C# Namespaces
      4. C# Partial Class & Method

**Week-8 (Midterm)**

**Week-9 (Java Functional Console Programming)**

* + - 1. Java Programming
         1. Java Introduction

Java Hello World

Java JVM, JRE, and JDK

Java Variables

Java Data Types

Java Operators

Java Input and Output

Java Expressions & Blocks

Java Comment

* + - * 1. Java Flow Control

Java if..else

Java switch statement

Java for loop

Java for-each loop

Java while loop

Java break statement

Java continue statement

* + - * 1. Java Arrays

Java Arrays

Multidimensional Array

Java Copy Array

**Week-10 (Java Functional Console Programming)**

1. Java OOP-I
2. Java Class and Objects
3. Java Methods
4. Java Constructor
5. Java Strings
6. Java Access Modifiers
7. Java this keyword
8. Java final keyword
9. Java recursion
10. Java instanceof operator
11. Java OOP-II
12. Java Inheritance
13. Java Method Overriding
14. Java super keyword
15. Abstract Class & Method
16. Java Interfaces
17. Java Polymorphism
18. Java Encapsulation
19. Java OOP-III
20. Nested & Inner Class
21. Java Static Class
22. Java Anonymous Class
23. Java Singleton
24. Java enum class
25. Java enum constructor
26. Java enum string
27. Java reflection
28. Java Exception Handling
29. Java Exceptions
30. Java Exception Handling
31. Java try..catch
32. Java throw and throws
33. Java catch Multiple Exceptions
34. Java try-with-resources
35. Java Annotations
36. Java Annotation Types
37. Java Logging
38. Java Assertions
39. Java List
40. Java Collection Framework
41. Java Collection Interface
42. Java List Interface
43. Java ArrayList
44. Java Vector
45. Java Stack
46. Java Queue
47. Java Queue Interface
48. Java PriorityQueue Interface
49. Java Deque Interface
50. Java LinkedList
51. Java ArrayDeque
52. Java BlockingQueue Interface
53. Java ArrayBlockingQueue
54. Java LinkedBlocking Queue
55. Java Map
56. Java Map Interface
57. Java HashMap
58. Java LinkedHashMap
59. Java WeakHashMap
60. Java EnumMap
61. Java SortedMap Interface
62. Java NavigableMap Interface
63. Java TreeMap
64. Java ConcurrentMap Interface
65. Java ConcurrentHashMap
66. Java Set
67. Java Set Interface
68. Java HashSet
69. Java EnumSet
70. Java LinkedHashSet
71. Java SortedSet Interface
72. Java NavigableSet Interface
73. Java TreeSet
74. Java Algorithms
75. Java Iterator
76. Java ListIterator

**Week-11 (Java Functional Console Programming)**

1. Java I/O Streams
2. Java I/O Streams
3. Java InputStream
4. Java OutputStream
5. Java FileInputStream
6. Java FileOutputStream
7. Java ByteArrayInputStream
8. Java ByteArrayOutputStream
9. Java ObjectInputStream
10. Java ObjectOutputStream
11. Java BufferedInputStream
12. Java BufferedOutputStream
13. Java PrintStream
14. Java Reader/Writer
15. Java Reader
16. Java Writer
17. Java InputStreamReader
18. Java OutputStreamWriter
19. Java FileReader
20. Java FileWriter
21. Java BufferedReader
22. Java BufferedWriter
23. Java StringReader
24. Java StringWriter
25. Java PrintWriter
26. Additional Topics
27. Java Scanner Class
28. Java Type Casting
29. Java autoboxing and unboxing
30. Java Lambda Expression
31. Java Generics
32. Java File Class
33. Java Wrapper Class
34. Java Command Line Arguments
35. JNLP (Java Network Launch Protocol)

**Week-12 (C/Cpp GUI Programming)**

1. GUI Application Development (Windows)
   1. C Programming
   2. Cpp Programming

**Week-13** **(Csharp GUI Programming)**

1. GUI Application Development (Windows)
   1. Csharp Programming

**Week-14 (Csharp GUI Programming)**

1. GUI Application Development (Windows)
   1. Csharp Programming

**Week-15 (Java GUI Programming)**

1. GUI Application Development (Windows)
   1. Java Programming

**Week-16 (Final)**