

**Recep Tayyip Erdogan University**

**Faculty of Engineering and Architecture**

**Computer Engineering**

CEN310- Parallel Programming

**Syllabus**

**Spring Semester, 2024-2025**

|  |  |
| --- | --- |
| **Instructor** | **Asst. Prof. Dr. Uğur CORUH** |
| **Contact Information** | [ugur.coruh@erdogan.edu.tr](mailto:ugur.coruh@erdogan.edu.tr) |
| **Office Number** | **F-301** |
| **Microsoft Teams Code** | **ilpgjzn** |
| **Lecture Hours and Days** | **Theory-Friday (09:00-12:00)** |
| **Lecture Classroom** | **D-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 [CEN310] tag for the fast response and write formal, clear, and short emails. |
| **Lecture and Communication Language** | English |
| **Theory/Laboratory Course Hour Per Week** | 3/0 hours |
| **Credit** | 4 |
| **Prerequisite** |  |
| **Corequisite** |  |
| **Requirement** |  |

*\*TBD: To Be Defined.*

1. **Course Description**

This course introduces fundamental concepts and practices of parallel programming, focusing on designing and implementing efficient parallel algorithms using modern programming frameworks and architectures. Students will learn to analyze sequential algorithms and transform them into parallel solutions, understanding key concepts such as parallelization strategies, load balancing, synchronization, and performance optimization.

1. **Course Learning Outcomes**

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

* Design and implement parallel algorithms by applying appropriate parallelization strategies and patterns using modern frameworks like OpenMP and MPI
* Analyze and optimize parallel program performance through proper evaluation of efficiency, scalability, and bottleneck identification
* Develop parallel solutions using various programming models (shared memory, distributed memory) while effectively managing synchronization and data structures
* Apply parallel computing concepts to solve real-world computational problems using appropriate architectures and tools
* Evaluate and select appropriate parallel computing approaches based on problem requirements, considering factors such as scalability, efficiency, and hardware constraints

1. **Course Topics**

* Introduction to parallel computing concepts and architecture
* Parallel algorithm design and performance analysis principles
* Shared memory programming using OpenMP framework
* Distributed memory programming with Message Passing Interface (MPI)
* Performance optimization and profiling tools in parallel systems
* GPU computing and heterogeneous parallel architecture
* Advanced parallel programming patterns and synchronization techniques
* Real-world parallel computing applications and case studies

1. **Textbooks and Required Hardware**

This course does not require a specific coursebook. You can use the following books and online resources for reference:

* *Peter S. Pacheco, An Introduction to Parallel Programming, Morgan Kaufmann*
* *Michael J. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill*
* *Barbara Chapman, Using OpenMP: Portable Shared Memory Parallel Programming, MIT Press*
* *Additional resources will be provided during the course*

**Required Hardware and Software**

During this course, you should have:

* A laptop/desktop with Windows 10 or 11 with the following minimum specifications:
  + Multi-core processor
  + 16GB RAM (recommended)
  + 100GB of free disk space
  + Windows 10 version 2004 and higher (Build 19041 and higher) or Windows 11
* Required software (all free):
  + Visual Studio Community 2022
  + Windows Subsystem for Linux (WSL2)
  + Ubuntu distribution on WSL
  + Git for Windows
* Development environment setup:
  + Visual Studio Community 2022 with:
    - "Desktop development with C++" workload
    - "Linux development with C++" workload
    - WSL development tools
* WSL requirements:
  + Ubuntu on WSL
  + GCC/G++ compiler (installed via apt)
  + OpenMP support
  + MPI implementation (will be installed during class)

Installation instructions and support for setting up the development environment will be provided during the first week of the course. All programming assignments, classroom exercises, and examinations will be conducted using this setup.

1. **Grading System**

You will complete **one project** and **two written quizzes** throughout the semester. You are expected to submit your Midterm Parallel Implementation Report at the midterm, demonstrating parallel algorithms and performance analysis aligned with your project plan. In the 15th week, you will present and submit your Final Project Implementation Report.

You will take a written quiz in the 7th week and another in the 14th week.

|  |  |  |  |
| --- | --- | --- | --- |
| **Assessment** | **Code** | **Weight** | **Scope** |
| Midterm Project Report | MPR1 | 60% | Midterm |
| Quiz-1 | QUIZ1 | 40% | Midterm |
| Final Project Report | MPR2 | 70% | Final |
| Quiz-2 | QUIZ2 | 30% | Final |

Your final passing grade can be improved through the following achievements: (Bonus Points TBD)

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, homework, and announcements will be shared over Microsoft teams and Github. 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 on the online platform, zoom, Microsoft teams or google meets, or meet at the time specified in the course schedule. Attendance will be taken.

1. **Late Homework**

Throughout the semester, assignments and reports 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 homework by students.

1. **Course Platform and Communication**

Microsoft Teams 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.

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 for 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**

**Course Schedule Overview**

**Regular Course Time:** Every Thursday [Time TBD] (3 hours) **Project Review Sessions:** Full day (09:00-17:00)

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Date** | **Subjects** | **Other Tasks** |
| Week 1 | 14.02.2025 | Course Introduction and Overview  • Course plan and requirements  • Introduction to parallel computing  • Setting up development environment (VS Code, WSL) | Environment Setup (3 hours) |
| Week 2 | 21.02.2025 | Parallel Computing Fundamentals  • Types of parallelism  • Architecture overview  • Performance metrics  • Analysis of parallel systems | First Code Exercise (3 hours) |
| Week 3 | 28.02.2025 | Introduction to OpenMP  • Shared memory programming  • Basic directives  • Thread management  • Data parallelism concepts | OpenMP Practice (3 hours) |
| Week 4 | 07.03.2025 | Advanced OpenMP  • Parallel loops  • Synchronization  • Data sharing  • Performance optimization strategies | OpenMP Practice (3 hours) |
| Week 5 | 14.03.2025 | Performance Analysis & MPI Introduction  • Profiling tools  • Debugging techniques  • Distributed memory concepts  • Basic MPI concepts | Performance Lab (3 hours) |
| Week 6 | 21.03.2025 | Advanced MPI & Parallel Patterns  • Point-to-point communication  • Collective operations  • Common parallel patterns  • Design strategies | MPI Setup & Implementation (3 hours) |
| Week 7 | 28.03.2025 | **Quiz-1**  • Written examination | **Quiz-1** (3 hours) |
| Week 8 | 04.04.2025 | **Midterm Project Review**  • Project presentations  • Performance analysis discussions | **Project Presentations** (Full Day) 09:00-17:00 |
| Week 9 **MIDTERM** | **5-13.04.2025** | **Midterm Examination Period** | **Midterm Project Report Due**  As scheduled |
| Week 10 | 18.04.2025 | Parallel Algorithm Design & GPU Basics  • Decomposition strategies  • Load balancing  • GPU architecture Fundamentals  • CUDA introduction | Algorithm Design Lab (3 hours) |
| Week 11 | 25.04.2025 | Advanced GPU Programming  • CUDA programming model  • Memory hierarchy  • Optimization techniques  • Performance considerations | CUDA Implementation (3 hours) |
| Week 12 | 02.05.2025 | Real-world Applications I  • Scientific computing  • Data processing applications  • Performance optimization  • Case studies | Application Development (3 hours) |
| Week 13 | 09.05.2025 | Real-world Applications II  • Advanced parallel patterns  • N-body simulations  • Matrix computations  • Big data processing | Case Study Implementation (3 hours) |
| Week 14 | 16.05.2025 | **Quiz-2**  • Written examination | **Quiz-2** (3 hours) |
| Week 15 | 23.05.2025 | **Final Project Review**  • Project presentations  • Performance analysis discussions | **Project Presentations** (Full Day) 09:00-17:00 |
| Week 16 **FINAL** | **24.05-04.06.2025** | **Final Examination Period** | **Final Project Report Due** As scheduled |

**Important Time Notes:**

* Regular classes: 3-hour sessions on Thursdays
* Quiz sessions: Regular 3-hour class period
* Project Review sessions (Week 8 & 14): Full day (09:00-17:00)
* Midterm and Final periods: As scheduled by the university

**Key Dates:**

* Quiz-1: March 28, 2025 (3 hours)
* Midterm Project Review: April 4, 2025 (Full Day)
* Quiz-2: May 16, 2025 (3 hours)
* Final Project Review: May 23, 2025 (Full Day)