**Code and Name:**

**MAT6020 High-Performance Mathematical Computing in Technology**

**Unit:**

Institute of Science, Department of Mathematics

**Details:**

* **Term:** 2023-2024 Spring
* **Status:** Elective
* **Class Level:** 1
* **Credit Hours:** 3-0-0-3
* **ECTS:** 6
* **Language:** Turkish

**Course Instructors:**

* **Course Coordinator:** ...
* **Assistant Instructor:** ...
	+ **Phone:** ...
	+ **Email:** ...@firat.edu.tr
	+ **Social Accounts:** ...

**Weekly Schedule**

| **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

**Teaching Method:**
Each weekly hour will include at least 45 minutes of face-to-face teaching.

**Location:**

* **In-person (YY):** Classroom (To be announced)
* **Remote (UE):** -

**Objective:**

1. To impart fundamental knowledge and skills in parallel computing.
2. To teach the parallelization design and analysis of traditional and modern numerical methods used for solving medium- and large-scale scientific and engineering problems.
3. To introduce tools and methods required for running parallel algorithms on various computational platforms.
4. To develop applications of algorithms designed for large-scale distributed, shared-memory, and distributed-shared memory server systems.

**Materials:**

1. Wilkinson, B., & Allen, M. (1999). *Parallel Programming*, Prentice Hall
2. Pacheco, P.S. (1997). *Parallel Programming with MPI*, Morgan Kaufmann
3. Arbenz, P., & Petersen, W. (2004). *Introduction to Parallel Computing*, Oxford University Press

**Student Responsibilities:**

Students are required to attend at least 70% of the classes.

**Weekly Lesson Plan:**

| **Week** | **Topic** | **Methodology** |
| --- | --- | --- |
| 1 | Introduction to the course and key concepts | Face-to-Face |
| 2 | **Introduction to Parallel Computing**: Fundamental definitions | Face-to-Face |
| 3 | **Parallel Computing Architectures**: Overview and types | Face-to-Face |
| 4 | Memory hierarchy and operating systems | Face-to-Face |
| 5 | **Caching and Performance**: Definitions and characteristics | Face-to-Face |
| 6 | **Message-Passing Computations**: Point-to-point and collective communications | Face-to-Face |
| 7 | Perfect parallel algorithms | Face-to-Face |
| 8 | **Parallel Techniques**: Partitioning, divide-and-conquer strategies | Face-to-Face |
| 9 | **Midterm Exam** | Face-to-Face |
| 10 | **Parallel Techniques**: Synchronous and data-parallel computations | Face-to-Face |
| 11 | Load balancing | Face-to-Face |
| 12 | **Shared-Memory Programming**: Architectures and threads | Face-to-Face |
| 13 | Accessing shared data, cache-coherent systems | Face-to-Face |
| 14 | **Programming with OpenMP**: Current developments in nanotechnology and new trends in HPC | Face-to-Face |

**Assessment and Evaluation:**

| **Method** | **Quantity** | **Weight** |
| --- | --- | --- |
| **Midterm Exam** | 1 | 50% |
| **Quizzes** | None | - |
| **Assignments** | Pre- and post-midterm activities | - |
| **Projects** | None | - |
| **Final Exam** | 1 | 50% |

**Learning Outcomes:**

1. Understand, use, and identify the limiting factors of parallel algorithms.
2. Select appropriate parallel algorithms for a given problem.
3. Learn point-to-point, collective communications, and perfect parallel algorithms.
4. Understand synchronous and data-parallel computations, load balancing, and shared-memory programming.
5. Access shared data, understand cache-coherent systems, and program with OpenMP.

**Special Notes:**

* **UE:** Remote Education
* **YY:** Face-to-Face Education