Course Title	Parallel and Distributed Computing				
Course Code	ECE430				
Course Type	Elective				
Level	Bachelor (1st Cycle)				
Year / Semester	4th Year / 8th Semester				
Teacher's Name	TBA				
ECTS	6	Lectures / week	3 hours / 14 weeks	Laboratories / week	N/A
Course Purpose and Objectives	The course aims to introduce students to the technologically important area of parallel and distributed computing, and how local computation and information exchange can result to a global computation and problem solving. Important architectures and technological tools and advances of parallel computer machines are discussed. The course introduce the students to important characteristics and algorithms for important problems of distributed and parallel systems and computer communication networks, such as multi-core, distributed shared memory, message passing, fault-tolerance, consensus, and leader election.				
Learning Outcomes	Upon successful completion of the course, students will be able to: • Describe and discuss parallel computing and performance measures • Explain parallel and distributed computing models and architectures • Describe and apply parallel and distributed programming languages, environments and algorithms in parallel and distributed settings • Describe, utilize and compare important distributed and parallel algorithms for each model, for various important problems such as routing, sorting, numerical and control problems • Understand, discuss and apply fundamental concepts in the area of message passing and shared memory concurrency • Analyze and compare parallel and distributed algorithms for correctness, reliability, and performance.				
Prerequisites	CSE320	Co-requisites	None		
Course Content	Distributed systems Protocols: The model, Communication protocols, Routing algorithms Deadlock-free packet switching				

	Fundamental Algorithms: Wave and traversal algorithms, Election algorithms, Termination detection, Synchrony in networks Fault Tolerance: Fault tolerance in distributed systems Parallel systems Introduction, The Power and Potential of Parallelism, Examining Sequential and Parallel Programs, Parallelism Using Multiple Instruction Streams, Scalable Performance and Portability Parallel Computers And Their Model: Balancing Machine Specifics with Portability, A Look at Five Parallel Computers, The RAM: An Abstraction of a Sequential Computer, The PRAM: A Parallel Computer Model Reasoning about Performance: Basic Concepts, Performance Trade-Offs, Measuring Performance First Steps Towards Parallel Programming: Task and Data Parallelism Scalable Algorithmic Techniques: The Inevitability of Trees, Blocks of Independent Computation, Schwartz' Algorithm, Assigning Work To Processes Statically and Dynamically, The Reduce & Scan Abstractions, Trees Programming with Threads: POSIX Threads, Thread Creation and Destruction, Mutual Exclusion Synchronization, Safety Issues, Performance Issues Open MP: Examples, Thread Behavior and Interaction Sections Local View Programming Languages: MPI: The Message Passing Interface, Getting Started, Safety Issues, Performance Issues Introduction to General General-Purpose GPU Programming with CUDA.
Teaching Methodology	Face- to- face Students in this course are expected to work on group activities, such as projects, assignments, literature reviews that may deal with the investigation and solution of a problem or with the design and/or implementation of a system. Group activities aim to motivate students to work within a group, develop critical thinking, improve their communication and decision-making skills, and promote active learning.
Bibliography	Vipin Kumar, George Karypis, Anshul Gupta, Ananth Grama, Introduction to Parallel Computing, Addison-Wesley Introduction to Distributed Algorithms, Gerard Tel, Cambridge University Press. Calvin Lin, Larry Snyder, Principles of Parallel Programming, Pearson

<table>
<tr><td></td><td>Michael J. Quinn, Parallel Computing, Theory and Practice, McGraw-Hill

Ian Foster, Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering, Addison-Wesley Longman

Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, NVIDIA, Addison Wesley.</td></tr>
<tr><td>Assessment</td><td><table>
<tr><td>Examinations</td><td>70%</td></tr>
<tr><td>Assignments/Lab</td><td>20%</td></tr>
<tr><td>Class Participation and Attendance</td><td>10%</td></tr>
<tr><td></td><td>100%</td></tr>
</table></td></tr>
<tr><td>Language</td><td>English</td></tr>
</table>