1. **Course Number and Course Title:**
   COE 694-07 – Parallel Computing for Data-Intensive and AI Applications

2. **Credit Hours:**
   3 – 0 – 3

3. **Prerequisites and/or Co-Requisites:**
   Prerequisite: Approval of the CSE Head of Department
   Co-requisites: None
   Competencies: Undergraduate-level knowledge of C++ programming, data structures, and computer architecture

4. **Name and Contact Information of Instructor:**
   Name: Dr. Mohammad Daoud
   Email: mdaoud@aus.edu
   Office: ESB 2078
   Phone: 06 515 4995
   Office Hours: Posted on office door and iLearn; also by appointment

5. **Course Description (Catalog Description):**
   Provides an in-depth exploration of parallel computing systems for modern data processing and Artificial Intelligence (AI) applications. Covers a range of parallel architectures, including multi-core CPUs, many-core graphics processing units (GPUs), and distributed-memory clusters. Includes parallel programming paradigms such as multi-threading, heterogeneous computing with GPUs, and message passing. Emphasizes parallel computation concepts and strategies relevant to Deep Learning, including tensor-based operations, convolutional workloads, and techniques for overlapping communication and computation in high-performance AI systems.

6. **Textbook and other Supplemental Material:**
   Textbook:
   - An Introduction to Parallel Programming, 2nd Edition, Peter Pacheco and Matthew Malensek, Morgan Kaufmann (Elsevier), 2021.

   Other supplemental material:
   - Programming Massively Parallel Processors, 4th Edition, Wen-mei W. Hwu, David B. Kirk, and Izzat El Hajj, Morgan Kaufmann (Elsevier), 2022.
   - Parallel & High Performance Computing, Robert Robey & Yuliana Zamora, Manning Publications, 2021.

7. **Course Learning Outcomes:**
   Upon completion of the course, students will be able to:

   1. Analyze the foundational motivations and performance limits of serial execution in the context of large-scale data and Artificial Intelligence (AI) workloads.

   2. Differentiate between parallel shared address-space, many-core GPU, and distributed-memory parallel architectures.

3. Apply parallel design methodologies, including problem decomposition and mapping, to computationally intensive workloads.
4. Evaluate the performance and scalability of parallel algorithms using standard metrics and analytical performance models.
5. Develop shared-memory parallel programs using appropriate parallel programming paradigms and synchronization mechanisms.
6. Design GPU-accelerated computational primitives for compute-intensive AI workloads.
7. Build end-to-end distributed-memory parallel applications that exploit communication–computation overlap to improve performance and scalability.

8. Teaching and Learning Methodologies:

Methods include lectures, problem and project-based learning methods (assignments, exams, project, presentation), and class discussions.

9. Course Topics and Schedule:

Topic/Activity	**Weeks**
Motivation for parallel computing and performance limits of serial execution in data-intensive and AI workloads.	Week #1
Overview of shared address-space, many-core GPU, and distributed-memory parallel system architectures.	Week #2
Principles of parallel algorithm design, including problem decomposition and workload mapping.	Week #3
Analytical modeling and performance evaluation of parallel programs, including speedup, efficiency, and scalability.	Week #4
Concepts of shared address-space parallelism, including threading models and synchronization mechanisms.	Week #5
Practical development of shared-memory parallel programs using OpenMP.	Week #6
Architecture of many-core systems including the Single Instruction, Multiple Threads (SIMT) model and GPU memory hierarchy.	Week #7
Development of GPU-accelerated applications using the CUDA execution model and thread indexing.	Week #8
GPU-accelerated implementation of machine learning primitives for AI workloads.	Week #9
Application of parallel computation in artificial intelligence, with emphasis on convolutional and training workloads.	Week #10
Evaluation of parallel software performance, including identification of performance bottlenecks using profiling tools.	Week #11
Build and analyze end-to-end parallel applications with emphasis on performance bottlenecks.	Week #12
Programming distributed-memory parallel architectures using the MPI programming model.	Week #13
Point-to-point and collective communication operations in distributed-memory systems.	Week #14
Optimization of parallel performance, including strategies for overlapping communication with computation.	Week #15
Final Exam	Week #16

10. Schedule of Laboratory and other Non-Lecture Sessions:

This course has no laboratory component. The course project is completed in groups of two students and focuses on the design, implementation, and evaluation of a parallel AI application using GPU platforms covered in class.

The project includes the following deliverables:

Project deliverables	Due Date (tentative)
Project topic proposal	Week #8
Working demonstration	Week #13
Written technical report and presentation	Week #14

11. Out-of-Class Assignments with Due Dates:

Homework	Due Date (tentative)
Homework 1: Parallel algorithm design and performance modeling	Week #5
Homework 2: Shared address-space and GPU-accelerated parallel computation	Week #10
Homework 3: Distributed-memory parallelism and performance optimization	Week #14

12. Student Evaluation:

Assessment	Weight	Due Date (tentative)
Homework	10%	Cf. Section 11
Course Project and Presentation	25%	Week #14
Midterm exam	30%	Week #10
Final Exam	35%	Week #16

13. Assessment Instruments:

Assessment	Course Learning Outcomes
Homework	O1–O7
Course Project and Presentation	O3, O4, O6
Quizzes	O1–O6
Midterm exam	O1–O5
Final Exam	O1–O7

14. Contribution of Course to Program Outcomes:

MSCOE Program Outcomes	Emphasis in this course	Course Learning Outcomes
(1) Perform research emphasizing creativity, independent learning and scientific methods in a chosen area of computer engineering.	◑	O2 – O7

(2) Apply advanced mathematics and engineering knowledge in identifying, formulating and solving engineering problems.	●	O3 – O7
(3) Select and use techniques, skills and modern tools necessary for research or professional practice.	◑	O2 – O7
(4) Communicate effectively	○	O6 – O7
(5) Recognize the need for, and engage in, lifelong learning in professional areas.	○	O1 – O7
(6) Attend to professional and ethical responsibilities.		

Emphasis: ● High; ◑ Medium; ○ Low; Blank – Nothing Specific Expected

MSMLR Program Outcomes	**Emphasis in this course**	**Course Learning Outcomes**
1. Perform research emphasizing creativity, independent learning, and scientific methods in the field of Machine Learning.	○	O2 – O7
2. Apply advanced mathematics, computer science knowledge, and software tools in identifying, formulating, and solving real world problems.	●	O3 – O7
3. Demonstrate an in-depth understanding of modern Machine Learning approaches, algorithms, and tools.	◑	O1, O6
4. Select and use techniques, skills, and modern tools necessary for research or professional practice.	◑	O2 – O7
5. Communicate effectively through technical presentations and reports.	○	O6 – O7
6. Recognize the need for, and engage in, lifelong learning in professional areas.	○	O1 – O7
7. Attend to professional and ethical responsibilities within global and societal contexts.		

Emphasis: ● High; ◑ Medium; ○ Low; Blank – Nothing Specific Expected

15. Letter Grade Policy:

Total (T)	Letter Grade
$90 \leq T$	A
$85 \leq T < 90$	A-
$80 \leq T < 85$	B+
$75 \leq T < 80$	B
$70 \leq T < 75$	B-
$65 \leq T < 70$	C+
$60 \leq T < 65$	C
$T < 60$	F