**Session 2025-2026**

|  |  |
| --- | --- |
| **Vision:** To help businesses uncover crucial  insights | **Mission:** To be a good data scientist |

**Program Educational Objectives of the program (PEO):** (broad statements that describe the professional and career accomplishments)

|  |  |  |  |
| --- | --- | --- | --- |
| PEO1 | **Preparation** | **P: Preparation** | **Pep-CL abbreviation**  **pronounce as Pep-si-lL easy to recall** |
| PEO2 | **Core Competence** | **E: Environment (Learning Environment)** |
| PEO3 | **Breadth** | **P: Professionalism** |
| PEO4 | **Professionalism** | **C: Core Competence** |
| PEO5 | **Learning Environment** | **L: Breadth (Learning in diverse areas)** |

**Program Outcomes (PO):** 1. Understand and Apply Parallel Programming Concepts

2. Analyse and Improve Program Performance.

3. Demonstrate Practical Skills in HPC Tools and Environments.

**Keywords of POs:**

Engineering knowledge, Problem analysis, Design/development of solutions, Conduct Investigations of Complex Problems, Engineering Tool Usage, The Engineer and The World, Ethics, Individual and Collaborative Team work, Communication, Project Management and Finance, Life-Long Learning

**PSO Keywords:** Cutting edge technologies, Research

“I am an engineer, and I know how to apply engineering knowledge to investigate, analyse and design solutions to complex problems using tools for entire world following all ethics in a collaborative way with proper management skills throughout my life.” *to contribute to the development of cutting-edge technologies and Research*.

**Integrity:** I will adhere to the Laboratory Code of Conduct and ethics in its entirety.

**Name and Signature of Student and Date**

Sujal Kumeriya– 01/09/2025

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Session** | **2025-26 (ODD)** | | **Course Name** | **HPC Lab** | |
| **Semester** | **7** | | **Course Code** | 22ADS706 | |
| **Roll No** | 64 | | **Name of Student** | Sujal Kumeriya | |
|  |  | |  |  |  |
| Practical Number | | 3 | | | |
| Course Outcome | | 1. Understand and Apply Parallel Programming Concepts 2. Analyse and Improve Program Performance | | | |
| Aim | | Introduction to OpenMP | | | |
| Problem Definition | | Introduction to OpenMP | | | |
| Theory  (100 words) | | **OpenMP** stands for Open Multi-Processing. It is an API (Application Programming Interface) that supports multi-platform shared-memory multiprocessing programming in C, C++, and  Fortran.  It allows developers to write parallel code easily using compiler directives, library routines, and environment variables.  **Key Features:**   1. Supports shared memory multiprocessing 2. Uses fork-join model: master thread forks a specified number of slave threads 3. Simple and readable syntax using pragma directives 4. Controlled by environment variables (e.g., OMP\_NUM\_THREADS) 5. Scales well for multi-core CPUs   **Execution Steps on CentOS/Linux**  **Step 1**: Install GCC with OpenMP support  Most CentOS systems have GCC preinstalled. If not:  sudo yum install gcc  To verify OpenMP support:  gcc -fopenmp --version  **Step 2**: Write the OpenMP Program  Create a file named openmp\_example.c.  nano openmp\_example.c  Paste your OpenMP C code (example below).  **Step 3**: Compile the Program  Use -fopenmp flag to enable OpenMP:  gcc -fopenmp -o openmp\_example openmp\_example.c  **Step 4**: Set Number of Threads (Optional)  You can set how many threads OpenMP should use:  export OMP\_NUM\_THREADS=4  **Step 5**: Run the Program  ./openmp\_example | | | |
| Code: | | #include <stdio.h>  #include <omp.h>  int main() {  int i;  int n = 12;  #pragma omp parallel for schedule(static, 3)  for (i = 0; i < n; i++) {  printf("Thread %d processing iteration %d\n", omp\_get\_thread\_num(), i);  }  return 0;  } | | | |
| Output | | A screenshot of a computer  AI-generated content may be incorrect. | | | |
| Output Analysis | | Our program successfully executes using OpenMP and four threads. | | | |
| Link of student Github profile where lab assignment has been uploaded | | **https://github.com/sujalkumeriya59/HPC\_LAB** | | | |
| Conclusion | | OpenMP supports shared memory multiprocessing, has simple and readable syntax and scales well for multi-core CPUs. | | | |
| Plag Report (Similarity index < 12%) | | **A screenshot of a computer  AI-generated content may be incorrect.** | | | |
| Date | | 01/09/2025 | | | |