



Department of Computer Technology

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

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-IL 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

Soham pimpalgaonkar – 28/10/2025



Department of Computer Technology

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

Session	2025-26 (ODD)	Course Name	HPC Lab
Semester	7	Course Code	22ADS706
Roll No	62	Name of Student	Soham pimpalgaonkar

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)	<p>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.</p> <p>It allows developers to write parallel code easily using compiler directives, library routines, and environment variables.</p> <p>Key Features:</p> <ol style="list-style-type: none">1. Supports shared memory multiprocessing2. Uses fork-join model: master thread forks a specified number of slave threads3. Simple and readable syntax using pragma directives4. Controlled by environment variables (e.g., OMP_NUM_THREADS)5. Scales well for multi-core CPUs <p>Execution Steps on CentOS/Linux</p> <p>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</p> <p>Step 2: Write the OpenMP Program Create a file named openmp_example.c.</p>



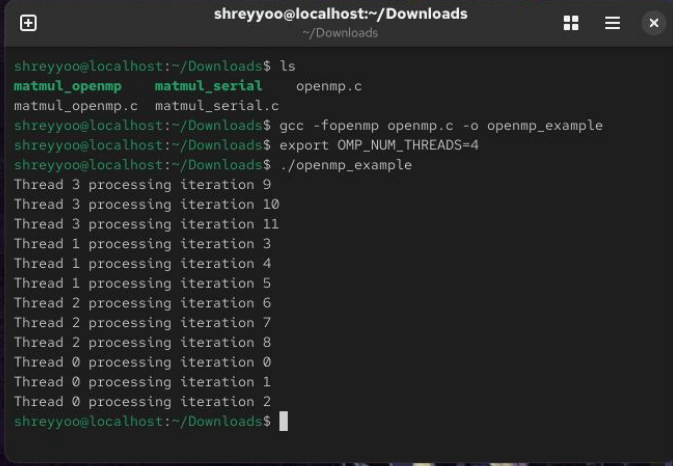
Department of Computer Technology

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

	<p>nano openmp_example.c</p> <p>Paste your OpenMP C code (example below).</p> <p>Step 3: Compile the Program</p> <p>Use -fopenmp flag to enable OpenMP:</p> <pre>gcc -fopenmp -o openmp_example openmp_example.c</pre> <p>Step 4: Set Number of Threads (Optional)</p> <p>You can set how many threads OpenMP should use:</p> <pre>export OMP_NUM_THREADS=4</pre> <p>Step 5: Run the Program</p> <pre>./openmp_example</pre>
Code:	<pre>#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; }</pre>
Output	



Department of Computer Technology

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

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/Sohampimpalgaonkar/HPC
Conclusion	OpenMP supports shared memory multiprocessing, has simple and readable syntax and scales well for multi-core CPUs.
Plag Report (Similarity index < 12%)	<div><div><div>ResultCitationWord Statistics</div><div><p>It allows developers to write parallel code easily using compiler directives, library routines, and environment variables.</p><p>Key Features:</p><ol style="list-style-type: none">1. Supports shared memory multiprocessing2. Uses fork-join model: master thread forks a specified number of slave threads3. Simple and readable syntax using pragma directives4. Controlled by environment variables (e.g., OMP_NUM_THREADS)5. Scales well for multi-core CPUs<p>Execution Steps on CentOS/Linux</p><p>Step 1: Install GCC with OpenMP support</p><p>Most CentOS systems have GCC preinstalled. If not:</p><pre>sudo yum install gcc</pre><p>To verify OpenMP support:</p><pre>gcc -fopenmp --version</pre><p>Step 2: Write the OpenMP Program</p><p>Create a file named oopenmp example.c.</p></div></div><div><div><div>10%Plagiarism</div><div><div>Exact Match7%</div><div>Partial Match3%</div><div>Unique90%</div></div><div><div>Remove Plagiarism</div><div>Download Report</div></div><div>Source(s) 2 matches from 2 Source(s) < 1 / 2 ></div><div><div>1. It allows developers to write parallel code easily using compiler directives, library routines, and environment variables. https://library.fiveable.me/parallel-and-distributed-computing/unit-4 3%</div><div><div>Exclude</div><div>Cite Source</div></div><div><div>2. https://en.wikipedia.org/wiki/Open_ 7%</div></div></div></div></div></div>
Date	28/10/2025