



### 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

<b>Vision:</b> To help businesses uncover crucial insights	<b>Mission:</b> To be a good data scientist
--	---

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

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

**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**

Samriddhi Kaswa– 01/09/2025



## 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.*

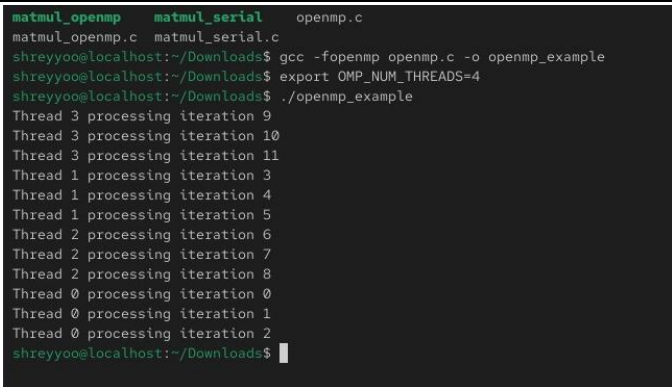
## Mission of the Department

*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	17	Name of Student	Samriddhi Kaswa

Practical Number	3
Course Outcome	<ol style="list-style-type: none"><li>1. Understand and Apply Parallel Programming Concepts</li><li>2. Analyse and Improve Program Performance</li></ol>
Aim	Introduction to OpenMP
Problem Definition	Introduction to OpenMP
Theory (100 words)	<p><b>OpenMP</b> 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><b>Key Features:</b></p> <ol style="list-style-type: none"><li>1. Supports shared memory multiprocessing</li><li>2. Uses fork-join model: master thread forks a specified number of slave threads</li><li>3. Simple and readable syntax using pragma directives</li><li>4. Controlled by environment variables (e.g., OMP_NUM_THREADS)</li><li>5. Scales well for multi-core CPUs</li></ol> <p><b>Execution Steps on CentOS/Linux</b></p> <p><b>Step 1:</b> 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><b>Step 2:</b> Write the OpenMP Program Create a file named openmp_example.c. nano 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>Paste your OpenMP C code (example below).</p> <p><b>Step 3:</b> Compile the Program Use -fopenmp flag to enable OpenMP: gcc -fopenmp -o openmp_example openmp_example.c</p> <p><b>Step 4:</b> Set Number of Threads (Optional) You can set how many threads OpenMP should use: export OMP_NUM_THREADS=4</p> <p><b>Step 5:</b> Run the Program ./openmp_example</p>
Code:	<pre>#include &lt;stdio.h&gt; #include &lt;omp.h&gt;  int main() {     int i;     int n = 12;      #pragma omp parallel for schedule(static, 3)     for (i = 0; i &lt; n; i++) {         printf("Thread %d processing iteration %d\n", omp_get_thread_num(), i);     }      return 0; }</pre>
Output	
Output Analysis	Our program successfully executes using OpenMP and four threads.



### 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.*

Link of student Github profile where lab assignment has been uploaded	<a href="https://github.com/samriddhikaswa/HPC">https://github.com/samriddhikaswa/HPC</a>
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"><li>1. Supports shared memory multiprocessing</li><li>2. Uses fork-join model: master thread forks a specified number of slave threads</li><li>3. Simple and readable syntax using pragma directives</li><li>4. Controlled by environment variables (e.g., OMP_NUM_THREADS)</li><li>5. Scales well for multi-core CPUs</li></ol><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 openmp_ example.c.</p></div></div><div><div>10%Plagiarism<div>Exact Match7%Partial Match3%Unique90%</div><div>Remove PlagiarismDownload Report</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. <a href="https://libraryfiveable.me/parallel-and-distributed-computing/unit-4">https://libraryfiveable.me/parallel-and-distributed-computing/unit-4</a> 3%<div>ExcludeCite Source</div></div><div>2. <a href="https://en.wikipedia.org/wiki/Open...">https://en.wikipedia.org/wiki/Open...</a> 7%</div></div></div></div></div>
Date	01/09/2025