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 harness the power of artificial intelligence and data science to solve real-world problems and enhance human potential.	**Mission:** To acquire skills through coursework, projects, and internships, while actively engaging in research and collaboration with peers to innovate and apply AI solutions.

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): (statements that describe what a student should be able to do and know by the end of a program)

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.

Prerana Bijekar 28 August 2025
Name and Signature of Student and Date
(Signature and Date in Handwritten)

Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri, Nagpur - 441 110
NAAC A++
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu
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	11	**Name of Student**	Prerana Bijekar

Practical Number	3
Course Outcome	**CO1:** Understand and Apply Parallel Programming Concepts **CO2:** Analyze and Improve Program Performance. **CO3:** Demonstrate Practical Skills in HPC Tools and Environments.
Aim	Introduction to OpenMP
Theory (100 words)	• Definition: OpenMP (Open Multi-Processing) is an API for parallel programming in shared-memory systems. • Language Support: Works with C, C++, and Fortran. • Execution Model: Follows fork-join model – master thread forks into multiple threads and joins back. • Directives: Uses compiler directives (`#pragma omp`) for parallelism. • Portability: Platform-independent and supported by most modern compilers. • Ease of Use: Incremental parallelization – sequential code can be parallelized step by step. • Parallel Constructs: Supports parallel regions, work-sharing constructs (`for`, `sections`), synchronization, and data sharing. • Scalability: Suitable for multi-core CPUs but limited to shared-memory systems.
Procedure and Execution (100 Words)	Algorithm: • touch filename.extension: create file • vi filename.extension: open file in editor • gcc –o filename filename.extension: compile file • ./filename: run file
	Code:

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.

```c
#include <stdio.h>
#include <omp.h>

int main(){
        int i;
        int n=10;
        int a[10];

#pragma omp parallel for
        for (i=0; i<n; i++){
                a[i] = i*i;
                printf("Thread %d: a[%d] = %d\n", omp_get_thread_num(), i, a[i]);
        }
        return 0;
}
```

```c
#include <stdio.h>
#include <omp.h>
int main() {
        omp_set_nested(1); // Enable nested parallelism
        #pragma omp parallel num_threads(2)
        {
                int id = omp_get_thread_num();
                printf("Outer thread %d starting\n", id);
                #pragma omp parallel num_threads(2)
                {
                        int inner_id = omp_get_thread_num();
                        printf("Inner thread %d of outer thread %d\n", inner_id, id);
                }
        }
return 0;
}
```

```c
#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;
}
```

```c
#include <stdio.h>
#include <omp.h>
int main() {
        #pragma omp parallel num_threads(4)
        {
                int id = omp_get_thread_num();

                printf("Thread %d before barrier\n", id);
                #pragma omp barrier
                if (id == 0) {
                        printf("All threads reached the barrier. Thread %d continuing.\n", id);
                }
        }
        return 0;
}
```

Output:

```
[lab1@localhost ~]$ touch mpcode2.c
[lab1@localhost ~]$ vi mpcode2.c
[lab1@localhost ~]$ gcc -fopenmp mpcode2.c -o mpcode2
[lab1@localhost ~]$ ./mpcode2
Thread 4: a[4] = 16
Thread 0: a[0] = 0
Thread 2: a[2] = 4
Thread 9: a[9] = 81
Thread 7: a[7] = 49
Thread 1: a[1] = 1
Thread 6: a[6] = 36
Thread 8: a[8] = 64
Thread 3: a[3] = 9
Thread 5: a[5] = 25
[lab1@localhost ~]$
```

```
Thread 6: a[6] = 36
Thread 8: a[8] = 64
Thread 3: a[3] = 9
Thread 5: a[5] = 25
[lab1@localhost ~]$ touch mpcode3.c
[lab1@localhost ~]$ vi mpcode3.c
[lab1@localhost ~]$ gcc -fopenmp mpcode3.c -o mpcode3
[lab1@localhost ~]$ ./mpcode3
Thread 0 processing iteration 0
Thread 0 processing iteration 1
Thread 0 processing iteration 2
Thread 2 processing iteration 6
Thread 2 processing iteration 7
Thread 2 processing iteration 8
Thread 1 processing iteration 3
Thread 1 processing iteration 4
Thread 1 processing iteration 5
Thread 3 processing iteration 9
Thread 3 processing iteration 10
Thread 3 processing iteration 11
[lab1@localhost ~]$ touch mpcode4.c
[lab1@localhost ~]$ vi mpcode4.c
[lab1@localhost ~]$ gcc -fopenmp mpcode4.c -o mpcode4
[lab1@localhost ~]$ ./mpcode4
Thread 2 before barrier
Thread 1 before barrier
Thread 3 before barrier
Thread 0 before barrier
All threads reached the barrier. Thread 0 continuing.
[lab1@localhost ~]$ touch mpcode5.c
[lab1@localhost ~]$ vi mpcode5.c
[lab1@localhost ~]$ gcc -fopenmp mpcode5.c -o mpcode5
[lab1@localhost ~]$ ./mpcode5
Outer thread 0 starting
Outer thread 1 starting
Inner thread 0 of outer thread 0
Inner thread 1 of outer thread 0
Inner thread 0 of outer thread 1
Inner thread 1 of outer thread 1
[lab1@localhost ~]$
```

Output Analysis	OpenMP improves program performance by distributing workload across multiple CPU cores. Performance gain depends on the problem size, number of threads, and hardware, with diminishing returns if overhead or synchronization dominates.
Conclusion	OpenMP provides a simple and efficient way to exploit parallelism in shared-memory systems. It helps programmers speed up applications with minimal code changes, making it a widely used tool for scientific and engineering computations.

Nagar Yuwak Shikshan Sanstha's

Yeshwantrao Chavan College of Engineering

(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)

Hingna Road, Wanadongri, Nagpur - 441 110

NAAC A++

Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu

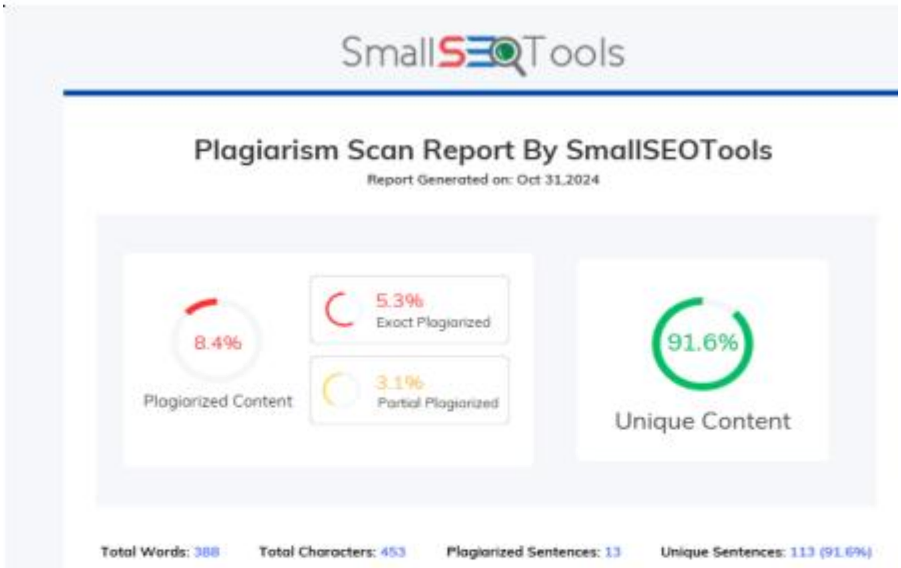
Department of Computer Technology

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.

<table>
<tr><td>Plag Report (Similarity index < 12%)</td><td>
</td></tr>
<tr><td>Date</td><td>09 September 2025</td></tr>
</table>