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 2025-2026

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 3 November 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

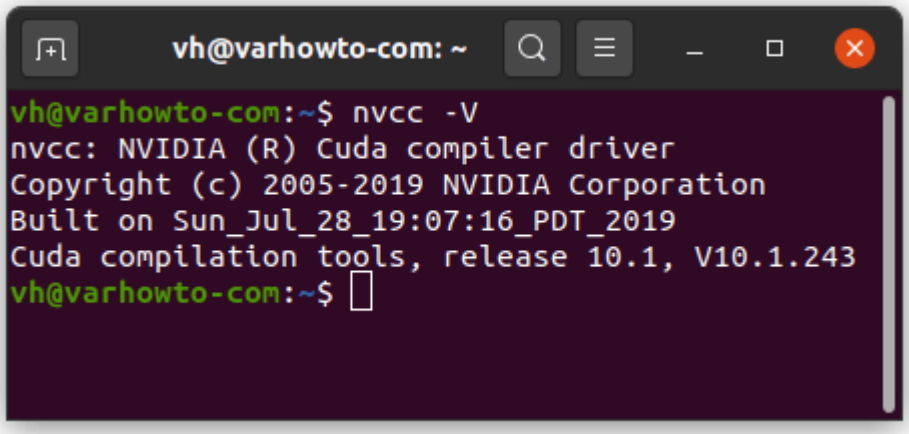
Practical Number	8
Course Outcome	**CO1:** Understand and Apply Parallel Programming Concepts **CO2:** Analyze and Improve Program Performance. **CO3:** Demonstrate Practical Skills in HPC Tools and Environments.
Aim	Introduction to GPU Computing
Theory (100 words)	CUDA (Compute Unified Device Architecture) is a parallel computing platform and programming model developed by NVIDIA. It provides a set of extensions to standard programming languages like C/C++ (called CUDA C/C++) that allow developers to harness the parallel power of NVIDIA GPUs. The Heterogeneous Programming Model: The CUDA model is heterogeneous, meaning an application uses both the CPU and the GPU: 1. Host Code: The sequential parts of the application run on the CPU (Host). 2. Device Code: The parallel, compute-intensive parts are written as special functions called kernels that run on the GPU (Device). Typical CUDA Program Flow: A standard CUDA program involves the following steps: 1. Allocation: Allocate memory on both the Host (CPU RAM) and the Device (GPU VRAM). 2. Transfer: Copy input data from Host memory to Device memory (via the PCIe bus). o *Function:* cudaMemcpy(..., cudaMemcpyHostToDevice) 3. Kernel Launch: The Host launches the kernel function on the Device. The kernel runs in parallel across thousands of threads. o *Syntax:* kernel_name<<<GridDim, BlockDim>>>(arguments); 4. Transfer Back: Copy the results from Device memory back to Host memory. o *Function:* cudaMemcpy(..., cudaMemcpyDeviceToHost)

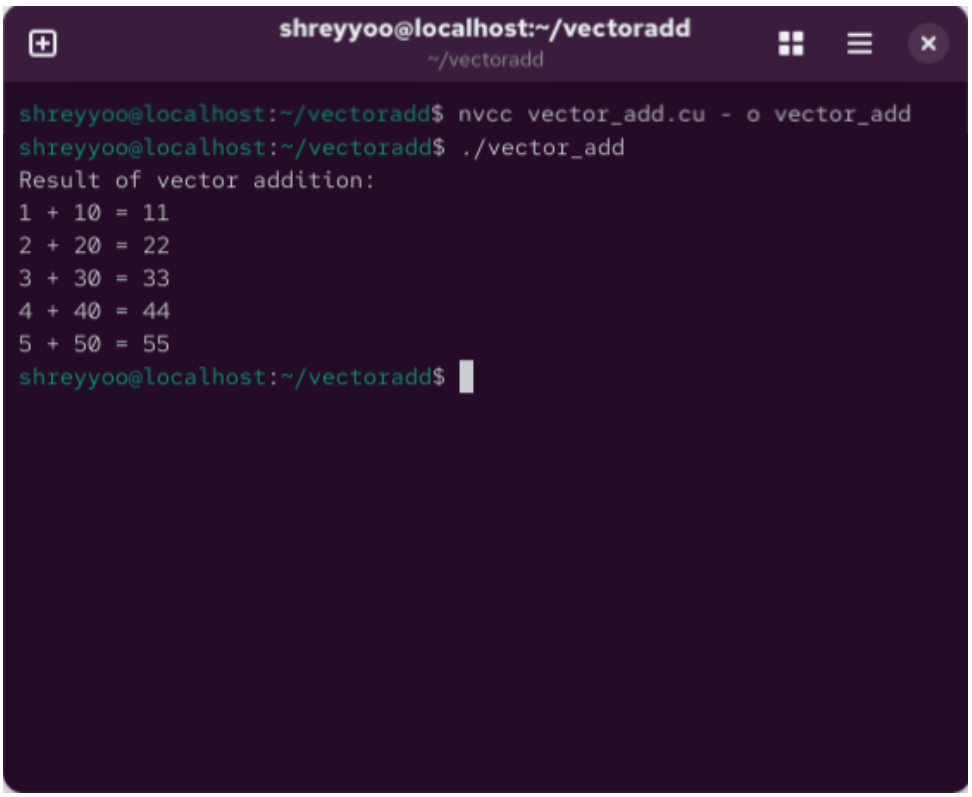
	5. Synchronization: The CPU code waits for the GPU kernel execution to complete (often implicit in the copy-back step). 6. Cleanup: Free memory on both the Host and the Device.
Procedure and Execution (100 Words)	Steps of implementation: • **Step 1: Check Prerequisites** nvcc --version • **Step 2: Create a CUDA File** • **Step 3: Compile the Program** nvcc vector_add.c -0 vector_add • **Step 4: Run the Program** ./vector_add • **Step 5: Verify GPU Availability** Nvidia-smi

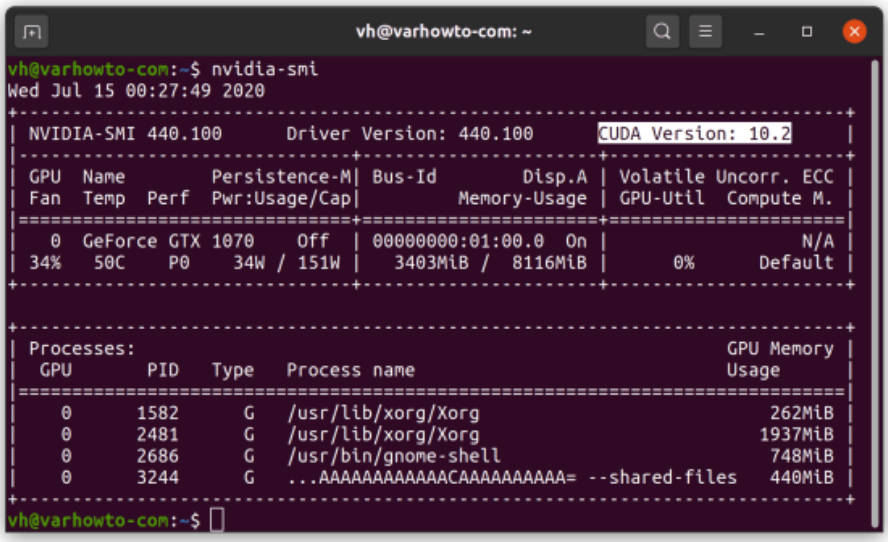
	Code:
	#include <stdio.h>
	__global__ void add(int *a, int *b, int *c, int n) {
	int index = threadIdx.x;
	if (index < n)
	c[index] = a[index] + b[index];
	}
	int main(void) {
	int n = 5;
	int a[5] = {1, 2, 3, 4, 5};
	int b[5] = {10, 20, 30, 40, 50};
	int c[5] = {0};
	int *d_a, *d_b, *d_c;
	int size = n * sizeof(int);
	cudaMalloc((void **)&d_a, size);
	cudaMalloc((void **)&d_b, size);
	cudaMalloc((void **)&d_c, size);
	cudaMemcpy(d_a, a, size, cudaMemcpyHostToDevice);
	cudaMemcpy(d_b, b, size, cudaMemcpyHostToDevice);
	add<<<1, n>>>(d_a, d_b, d_c, n);
	cudaMemcpy(c, d_c, size, cudaMemcpyDeviceToHost);
	printf("Result of vector addition:\n");
	for (int i = 0; i < n; i++)
	printf("%d + %d = %d\n", a[i], b[i], c[i]);
	cudaFree(d_a);
	cudaFree(d_b);
	cudaFree(d_c);
	return 0;

}

Output:


vh@varhowto-com: ~
vh@varhowto-com:~$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Sun_Jul_28_19:07:16_PDT_2019
Cuda compilation tools, release 10.1, V10.1.243
vh@varhowto-com:~$


shreyyoo@localhost:~/vectoradd
~/vectoradd
shreyyoo@localhost:~/vectoradd$ nvcc vector_add.cu - o vector_add
shreyyoo@localhost:~/vectoradd$ ./vector_add
Result of vector addition:
1 + 10 = 11
2 + 20 = 22
3 + 30 = 33
4 + 40 = 44
5 + 50 = 55
shreyyoo@localhost:~/vectoradd$

	
Output Analysis	The program executes successfully using CUDA and gives us the output.
Github Link	https://github.com/Prerana-Bijekar/HPC
Conclusion	The experiment successfully validated the principles of heterogeneous parallel computing by implementing a performance-critical task using NVIDIA CUDA on a CentOS platform.
Plag Report (Similarity index < 12%)	
Date	3 November 2025