



HOPE Foundation's

Finolex Academy of Management & Technology, Ratnagiri
Department of Information Technology

Association for Computing Machinery (ACM) Student Chapter

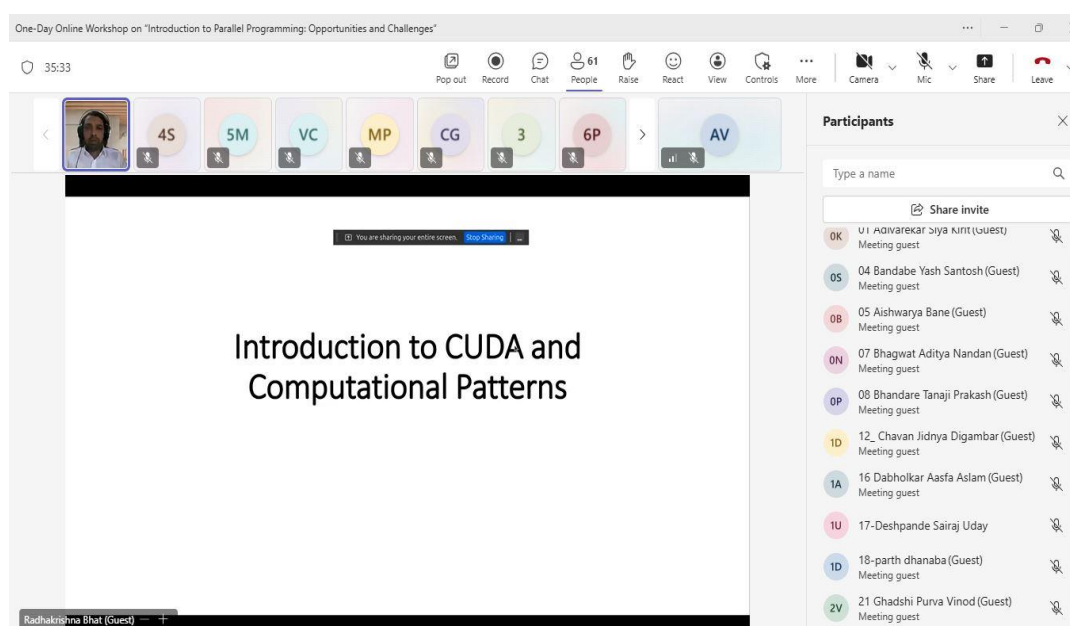
A One-Day Online Workshop on “Introduction to Parallel Programming: Opportunities and Challenges”

The **Department of Information Technology**, in association with its **Association for Computing Machinery (ACM) Student Chapter**, successfully organized a One-Day Online Workshop on “**Introduction to Parallel Programming: Opportunities and Challenges**” on 28 September 2026 for third-year Information Technology students.

The workshop was delivered by **Dr. Radhakrishna Bhat**, Associate Professor, School of Computer Engineering, Manipal Institute of Technology, Manipal. He introduced students to the fundamentals of parallel programming and discussed its relevance in high-performance computing, multicore processors, cloud computing, artificial intelligence, and data-intensive applications.

The session also highlighted the opportunities and practical challenges involved in developing parallel programs. The interactive workshop gave students valuable exposure to contemporary computing concepts beyond the regular syllabus.

A total of **44 students** attended the workshop. The event was coordinated successfully by Prof. A. B. Vartak under the guidance of Dr. V. A. Bharadi, the Head of the Department.



Dr. Radhakrishna Bhat expressing their views

One-Day Online Workshop on "Introduction to Parallel Programming: Opportunities and Challenges"

02:21:05

Pop out Recording Chat People 65 Raise React View Controls More Camera Mic Share Leave

Participants

Type a name

Share invite

In this meeting (65) Mute all

AV Ashish Vartak
Organiser

OK 01 Adivarekar Siya Kirit (Guest)
Meeting guest

OS 04 Bandabe Yash Santosh (Guest)
Meeting guest

OB 05 Aishwarya Bane (Guest)
Meeting guest

ON 07 Bhagwat Aditya Nandan (Guest)
Meeting guest

1D 12_ Chavan Jidnya Digambar (Guest)
Meeting guest

1A 16 Dabholkar Aasfa Aslam (Guest)
Meeting guest

TU 17-Deshpande Sairaj Uday

1D 18-parth dhanaba (Guest)
Meeting guest

Radhakrishna Bhat (Guest)

```

nvcc: NVIDIA (R) CUDA compiler driver
Copyright (c) 2002-2025 NVIDIA Corporation
Built on Fri Feb 22 20:21:59 PST 2025
CUDA compilation tools, release 12.8, V12.8.93
Build cuda_12.8.r12.8/compiler.35563879_0

1 //writefile main.cu
2 #include <iostream>
3 #include <cuda_runtime.h>
4
5 __global__ void helloFromGPU() {
6     printf("Reply from GPU thread %d in block %d\n", threadIdx.x, blockIdx.x);
7 }
8
9 int main() {
10     cudaDeviceProp devProp;
11     cudaGetDeviceProperties(&devProp, 0);
12     // Launch a kernel with 1 block and 5 threads
13     helloFromGPU(<1, 5>);
14
15     // Wait for the GPU to finish before the CPU proceeds
16     cudaDeviceSynchronize();
17     return 0;
18 }
19
Writing main.cu
1 //nvcc main.cu -o main_exec
2 //./main_exec
3
nvcc warning : Support for offline compilation for architectures prior to '-compute/sm/tee_75' will be removed in a future release (Use '-no-deptx'
Hello World from CPU!
Reply from GPU thread 0 in block 0!

```

Dr. Radhakrishna Bhat delivering practical session

One-Day Online Workshop on "Introduction to Parallel Programming: Opportunities and Challenges"

04:33:19

Pop out Recording Chat People 54 Raise React View Controls More Camera Mic Share Leave

Participants

Type a name

Share invite

In this meeting (54) Mute all

AV Ashish Vartak
Organiser

OK 01 Adivarekar Siya Kirit (Guest)
Meeting guest

OB 05 Aishwarya Bane (Guest)
Meeting guest

ON 07 Bhagwat Aditya Nandan (Guest)
Meeting guest

OP 08 Bhandare Tanaji Prakash (Guest)
Meeting guest

1C 15- Vaibhavi Chougule (Guest)
Meeting guest

1A 16 Dabholkar Aasfa Aslam (Guest)
Meeting guest

TU 17-Deshpande Sairaj Uday

2G 20-sumedh Gadhe (Guest)
Meeting guest

Radhakrishna Bhat (Guest)

Constant Memory and Caching

01 You are sharing your entire screen. Stop sharing

$$P[1] = 0 * M[0] + N[0] * M[1] + N[1] * M[2] + N[2] * M[3] + N[3] * M[4]$$

$$= 0 * 3 + 1 * 4 + 2 * 5 + 3 * 4 + 4 * 3$$

$$= 38$$

- We can make three interesting observations about the way the mask array M is used in convolution:
 - First, the size of the M array is typically small.
 - Second, the contents of M are not changed throughout the execution of the kernel.
 - Third, all threads need to access the mask elements. Even better, all threads access the M elements in the same order, starting from $M[0]$ and move by one element a time through the iterations of the for loop in 1D parallel convolution.
- These properties make the mask array an excellent candidate for **constant memory and caching**.

Dr. Radhakrishna Bhat delivering Theory session