



# CSCI 471

## Parallel and Distributed Programming Spring 2025

### Course Project

**Instructor:**

Dr. Muhammad Haris Rais

**Office Hours:**

Mon, Tues, Wed: 10:00 AM – 12:00 PM

By appointment: Thurs: 10:00 AM – 12:00 PM

Email: [mrais@vsu.edu](mailto:mrais@vsu.edu)

Office phone: 804-524-5415

# Course Project

- This is a group project
- No of members: 2 - 3
- Aim: To demonstrate the following capabilities:
  - Multiprocessing
  - Multithreading
  - Interprocess communication
  - Distributed computing over networked machines
  - Internode communication
- Your project must constitute and demonstrate each of the above features

# Demonstration

- We will earmark 4 machines in the lab. Each group will install their application on those machines. I will create a client pool to test the performance and correctness of their application
- Each group will explain their project using a small PowerPoint presentation and the IDE
- There is a sample idea in the next slides that covers the aspects required for this project. Take a look at them.
- Feel free to ask questions

# Deadlines

- 2/27/2025 – Project title and 10 mins presentation in the class
- 3/25/2025 - Interim Presentation – Project update
- 4/15/2025 – Final code for the machines (could be 3 to 5 depending upon your architecture) & a log file showing the tasks performed by each member with dates
- 4/22/2025 – Demonstration and presentation in the classroom

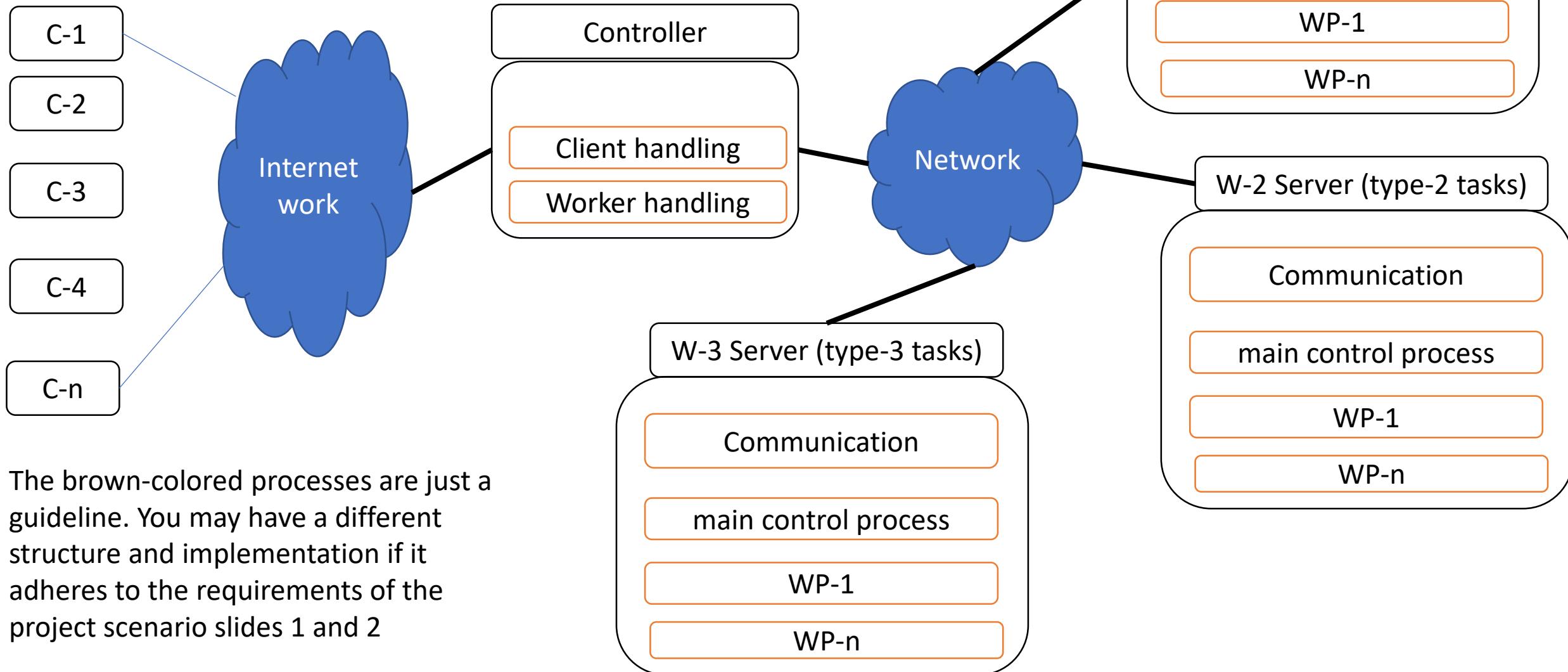
# A Sample Project Scenario (page 1)

- You have to develop a distributed computing application.
- Your clients (could be in thousands) will connect to a central server (say controller) over TCP/IP network and request for service by providing a work order file. [You can use any method for this transfer of workload but have to document it. If it requires any custom software for the clients, you have develop and share it.]
- The client work order file may contain 3 types of works: eg.
  - type-1: check if a number is prime
  - type-2: sum of first n Fibonacci numbers
  - type-3: factors of the number ‘n’
- The controller will keep track of each client’s workload.
- There will be 3 more separate machines (workers) serving the main controller
- Each machine is a specialist of one type of task

# A Sample Project Scenario (page 2)

- As soon as the controller receives tasks from the clients, it keep pushing them in 3 task-queues. It ensures that a client Identifier is attached to avoid any confusion later on.
- Simultaneously (you have to interpret it), the controller is sending the new tasks to the worker machines
- The worker machines are also connected over TCP/IP with the controller
- As the task is received by a worker, it puts it in some queue. The worker spawns as many processes as the number of logical processors on the machine. Each process keeps checking the queue for any work and if found, fetches the entry from the queue and starts performing the work
- Once the work is done, the worker process will put the result in a result queue
- The worker's main process keep looking for the items in the results queue and if found, send the result back to the controller (with the identifier)
- Once the job is completed for a client, the controller sends the result file back to the client and closes the connection with the client
- The complete service stays ON unless manually stopped

# A Sample Project Scenario (3)



# Files format

Input File

```
2,3  
1,234234234235674  
3,32  
1,4,NO  
1,47
```

Output File

```
2,3,6  
1,234234234235674,NO  
3,32,[1,2,4,8,16,32]  
1,4,NO  
1,47,YES
```