Task Structure & Guidelines:-

1. General Overview of Tasks and Their Design

The induction tasks are designed to simulate the real workflow followed in modern software development and open-source ecosystems. Each task focuses on building practical technical skills while encouraging self-learning, collaboration, and problem-solving.

Task 1.1 – Linux Migration

This task introduces members to controlling their development environment by installing and configuring a Linux distribution such as Ubuntu, Fedora, or Pop OS. Members either perform a dual boot setup or configure WSL (Windows Subsystem for Linux).

Purpose of Design:

- Helps members understand operating system fundamentals.
- Encourages familiarity with developer-preferred environments.
- Builds confidence in system-level setup and configuration.

Task 1.2 – Terminal Velocity

Members complete typing training and learn essential terminal commands such as navigation commands, file operations, and command piping.

Purpose of Design:

- Improves productivity and efficiency in development workflows.
- Builds comfort with command-line tools widely used in software engineering.
- Encourages speed, accuracy, and system interaction skills.

Task 2.1 – Containerization ("It Works on My Machine")

Members containerize a basic web application using Docker and configure services using docker-compose along with a small database or Redis instance.

Purpose of Design:

- Teaches environment consistency across systems.
- Introduces DevOps practices and container technology.
- Demonstrates real-world deployment workflows.

Task 2.1 – Algorithm Tour

Members implement Binary Search in multiple programming languages such as C/C++, Python, and JavaScript.

Purpose of Design:

- Strengthens algorithmic thinking.
- Helps understand programming paradigms across different languages.
- Develops ability to analyze performance and language-specific differences.

Task 2.2 – The Broken Web App

Members debug and fix an issue where a frontend button fails to communicate with the backend server.

Purpose of Design:

- Introduces full-stack development fundamentals.
- Builds debugging and troubleshooting skills.
- Helps understand API communication and browser developer tools.

Task 4.1 – Dotfiles Showcase

Members personalize their shell environment and development editors, then upload configuration files to GitHub.

Purpose of Design:

- Encourages workflow optimization.
- Promotes productivity through environment customization.
- Introduces configuration management.

Task 4.2 – Good First Issue

Members contribute to a real open-source project by solving documentation or beginner-level issues.

Purpose of Design:

- Introduces open-source collaboration culture.
- Builds confidence in contributing to external repositories.
- Enhances documentation and communication skills.

Task 5 – Career Track Selection

Members choose one specialization track:

Track A – Systems Engineer

Develop scripts to monitor system resources and automate execution using cron.

Track B – AI Researcher

Build a sentiment analysis tool using pretrained machine learning libraries.

Track C – Software Developer

Develop a command-line To-Do list application with persistent storage.

Purpose of Design:

- Helps members explore career interests.
- Provides domain-specific project exposure.
- Encourages independent project execution.

---

2. Evaluation Methodology and Criteria

Evaluation focuses on practical implementation and professional workflow practices.

Primary Evaluation Platform

- GitHub submissions are mandatory for evaluation.
- Optional blogs allow members to showcase learning experiences.

Evaluation Criteria

Task Completion

- Proper execution of deliverables such as screenshots, scripts, or applications.

Technical Implementation

- Correct configuration of Linux, Docker, debugging fixes, or algorithms.
- Functional and tested outputs.

Version Control Practices

- Proper branching strategy.
- Clear commit messages.
- Structured repositories.

Documentation

- Clear README explaining approach and results.
- Screenshots or output proof.

Independent Learning

- Demonstration of research and troubleshooting effort.

---

3. General Instructions for Raising a Pull Request (PR)

Pull Requests serve as the official method of submitting contributions for review.

PR Workflow

1. Fork the club repository.
2. Clone the fork to the local system.
3. Create a new branch for the assigned task.
4. Implement the task and test it locally.
5. Commit changes with meaningful messages.
6. Push the branch to GitHub.
7. Submit a Pull Request to the original repository.

PR Guidelines

- PR titles should clearly mention the task name.
- Include a short explanation of changes made.
- Attach screenshots or output results.
- Address review feedback promptly.

---

4. Expectations for Written Blogs (In Storytelling Format)

Blogs help members reflect on their technical journey and demonstrate conceptual clarity.

Blog Content Expectations

Task Introduction

Explain the objective and importance of the task.

Learning Experience

Describe challenges faced and how they were resolved.

Technical Understanding

Explain tools, commands, or algorithms used during implementation.

Reflection

Discuss personal learning outcomes and future improvements.

Blogs should focus on clarity, simplicity, and authentic learning experiences rather than technical complexity.

---

5. Fork Contribution and Pull Request Support

The club encourages guided independence rather than direct spoon-feeding.

Contribution Approach

Understanding Before Contributing

Members should:

- Read repository documentation.
- Explore existing project structures.
- Identify assigned tasks or beginner issues.

Forking Best Practices

- Maintain synchronization with the original repository.
- Work on dedicated branches for each task.

Pull Request Submission

Members should:

- Clearly describe their contribution.
- Provide testing evidence.
- Follow coding and documentation standards.

Team Collaboration Culture

- Encourage peer discussions.
- Share resources and research findings.
- Maintain professional communication.

Conclusion

The induction task structure is designed to introduce members to real-world software engineering workflows through progressive and hands-on learning. By completing tasks related to environment setup, terminal efficiency, containerization, debugging, customization, open-source contribution, and specialization tracks, members gain both technical expertise and professional collaboration skills.

The program aims to develop self-driven learners capable of contributing effectively to both club projects and the global open-source community.