



# Mastering the Machine Coding Round: A Candidate's Guide

Welcome to the exciting world of machine coding interviews! This round assesses your ability to design, develop, and implement a working software solution within a limited timeframe. It's your chance to showcase your problem-solving skills, coding proficiency, and understanding of software design principles.

This document equips you with valuable insights on how to approach a machine coding round effectively. We'll delve into the thought process, essential skills, and expectations using a sample scenario, but without getting into the specific question details.

## Understanding the Problem Statement:

The round typically begins with a problem statement outlining a real-world scenario. Our sample presents an application for a vehicle rental system. Here's what to focus on:

- **Functional Requirements:** Carefully analyse the features listed. Identify the key functionalities the system should possess, like user registration, car listing, slot management, searching, and shortlisting etc.
- **Additional Features:** Pay close attention to these details. They can showcase your ability to handle complexity and think beyond basic requirements. Having said that these are still extensions and finishing the basic functional requirements goes a long way.

## Developing a Solution Approach:

1. **Planning:** This initial phase is crucial. Take time to understand the problem thoroughly and finalise your entities, relationships, algorithm logic etc
2. **Data Structures:** Identify the data structures best suited to store and manage the information involved. Will you use classes, dictionaries, or lists?
3. **Modularity:** Break down the problem into smaller, manageable modules. This promotes code readability, maintainability, and easier testing.
4. **Error Handling:** Consider potential edge cases and exceptions. How will your code handle invalid inputs, unavailable data, or unexpected scenarios?

## Coding and Implementation:

- **Language Choice:** Select **Java** or **Golang** as a programming language.
- **Object-Oriented Principles:** Utilise object-oriented programming (OOP) concepts like classes, inheritance, and encapsulation to structure your code effectively.

- **Readability and Maintainability:** Write clean code that is easy to understand and modify in the future. Proper indentation and naming conventions enhance clarity.
- **Testability:** Consider how your code could be tested to ensure its functionality. Writing test cases will count towards brownie points.

### Key Skills and Expectations:

- **Problem-Solving:** Demonstrate your ability to dissect the problem, identify core functionalities, and develop a logical solution.
- **Coding Proficiency:** Write clean, efficient code that adheres to best practices and effectively implements the chosen solution.
- **Data Structures and Algorithms:** Select and utilise appropriate data structures and algorithms to manipulate and manage data efficiently.
- **Design Principles:** Apply OOP concepts to create a modular, maintainable, and scalable codebase.
- **Communication:** Clearly articulate your thought process, design choices, and any limitations in your implementation.

### Round Timeline:

- **60 mins:** We will share the problem statement doc at scheduled time & you will have to write the code independently. *Once completed, please email the code via zipped file or Google drive with access to us.*
- **15 mins** - Evaluation by the interviewer
- **45 mins:** Discussion / Extension on the code written & follow up questions with the interviewer

### Conclusion:

The machine coding round provides an opportunity to showcase your problem-solving prowess and coding capabilities. By following these guidelines, effectively communicating your thought process, and delivering a well-structured solution, you'll be well-positioned to impress in this crucial interview stage.

**Remember:** Don't hesitate to ask clarifying questions during the interview. It demonstrates your engagement and desire to understand the problem completely.