

SKILL POINT.

Where Engineering Meets Execution



In collaboration with

c~~o~~irei

Introduction

About Us

Skillpoint is not a traditional training institute. We are an execution-first engineering academy built to close the gap between what colleges teach and what companies actually expect.

Most students **fail** not because they lack intelligence — but because they lack engineering depth.

Skillpoint exists to fix three core problems:

01

Fragmented Learning – Students know tools but not systems

02

Theory-Heavy Teaching – Concepts without execution

03

Placement-Driven Shortcuts – Resume skills without substance

AI FRONTIER ENGINEERING

01

CORE PROGRAMMING FOUNDATIONS

Before systems, before scale — code must be solid.

- Programming fundamentals
- Variables, control flow, functions, recursion
- Memory concepts: stack vs heap
- Object-Oriented Programming (OOP)

Abstraction, Encapsulation, Inheritance, Polymorphism

SOLID principles

02

FOUNDATIONS OF INTELLIGENT SYSTEMS

We begin by building mathematical and conceptual clarity.

What is intelligence in machines?

- Difference between rule-based systems and learning systems
- Data as the core primitive

Supervised Learning Fundamentals

- Linear Regression
- Polynomial Regression
- Multiple Linear Regression

AI FRONTIER ENGINEERING

03

CLASSIFICATION SYSTEMS

We move from prediction to decision-making

- Logistic Regression
- Support Vector Machines (SVM)
- Naive Bayes
- K-Nearest Neighbors (KNN)
- Decision Trees

Each algorithm is taught with:

- Mathematical intuition
- Decision boundaries
- Strengths, weaknesses, and failure cases

04

ENSEMBLE LEARNING

Real-world models rarely survive alone

Ensemble Techniques

- Bagging
- Boosting
- Stacking
- Blending
- Cascading

AI FRONTIER ENGINEERING

05

DEEP LEARNING FOUNDATIONS

From algorithms to learning
architectures

- Neural Networks
- Perceptron
- Multi-Layer Perceptron (MLP)
- Activation functions
- Backpropagation

06

COMPUTER VISION & SEQUENTIAL MODELS

Convolutional Neural Networks (CNNs)

- Image feature extraction
- Filters, kernels, pooling
- Advanced image processing
models

Recurrent & Sequential Models

- RNNs
- LSTM
- Bi-LSTM

AI FRONTIER ENGINEERING

07

TRANSFORMERS & MODERN AI

We transition to state-of-the-art architectures.

- Transformer architecture
- Attention mechanisms
- Self-attention vs recurrence

08

AGENTIC AI & SYSTEM ARCHITECTURES

Beyond models — into intelligence systems.

Convolutional Neural Networks (CNNs)

- Image feature extraction
- Filters, kernels, pooling
- Advanced image processing models

Recurrent & Sequential Models

- RNNs
- LSTM
- Bi-LSTM

AI FRONTIER ENGINEERING

09

APPLICATION BUILDING & DEPLOYMENT

Students build real applications
using learned models.

- API development
- Model serving
- Integrations
- System orchestration



SYSTEMS AND SOFTWARE ENGINEERING

01

CORE

PROGRAMMING FOUNDATIONS

Before systems, before scale, code must be solid.

- Programming fundamentals
- Variables, control flow, functions, recursion
- Memory concepts: stack vs heap
- Object-Oriented Programming (OOP)

Abstraction, Encapsulation, Inheritance, Polymorphism

SOLID principles

02

DATA

STRUCTURES & ALGORITHMS (DSA)

Problem-solving is the backbone of engineering.

- Core Data Structures
- Arrays, Strings
- Linked Lists
- Stacks & Queues
- Hash Tables
- Trees (Binary Trees, BSTs)
- Heaps
- Graphs
- Algorithms
- Searching & Sorting
- Recursion & Backtracking
- Greedy Algorithms
- Divide and Conquer
- Dynamic Programming
- Graph Algorithms (BFS, DFS, Shortest Paths)
- Emphasis on time complexity, space complexity, and trade-offs.

SYSTEMS AND SOFTWARE ENGINEERING

03

OPERATING SYSTEMS & LOW-LEVEL SYSTEMS

We move from prediction to
decision-making.

- Processes & Threads
- Scheduling algorithms
- Inter-process communication (IPC)
- Deadlocks & concurrency issues
- Memory management & virtual memory
- File systems

04

COMPUTER NETWORKS & DISTRIBUTED BASICS

Software doesn't live alone.

- Networking fundamentals
- TCP/IP, HTTP/HTTPS
- DNS, Load Balancers
- Client-server architecture
- RESTful communication

SYSTEMS AND SOFTWARE ENGINEERING

05

DATABASES & DATA MANAGEMENT

Data is the backbone of every system.

- Relational databases (SQL)
- Schema design & normalization
- Indexing & query optimization
- Transactions & ACID properties
- NoSQL databases (concepts)

06

SOFTWARE ENGINEERING PRINCIPLES

From writing code to building software.

- Software Development Life Cycle (SDLC)
- Agile & Scrum
- Version control (Git)
- Code reviews & collaboration
- Testing fundamentals
 - Unit testing
 - Integration testing

SYSTEMS AND SOFTWARE ENGINEERING

07

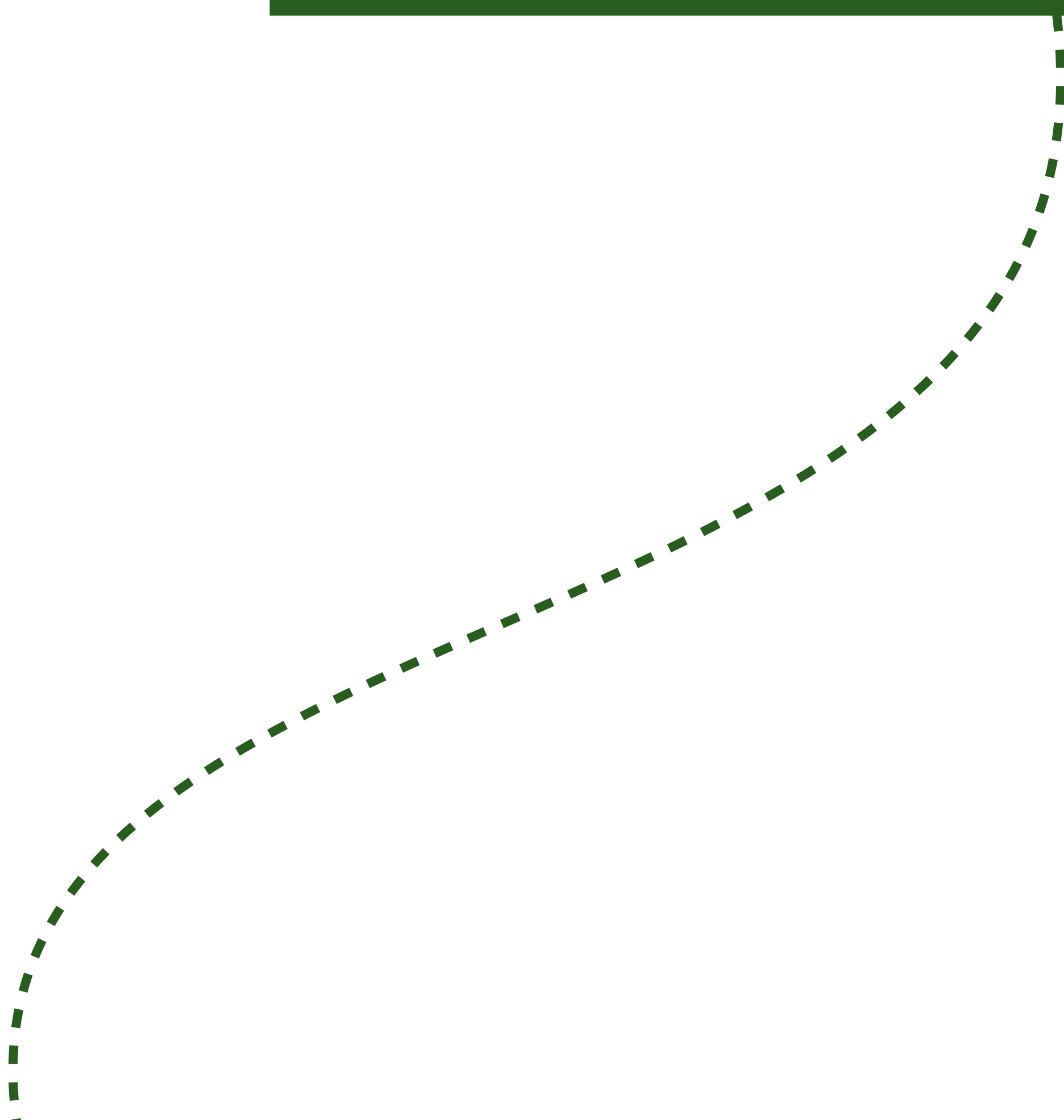
SYSTEM DESIGN & ARCHITECTURE

This is where engineers separate themselves

- System design fundamentals
- Scalability, reliability, availability
- Monolith vs Microservices
- API design principles
- Caching strategies
- Message queues

08

CLOUD, DEVOPS & DEPLOYMENT

- Cloud computing basics
 - Containers & Docker
 - CI/CD pipelines
 - Monitoring & logging
 - Infrastructure concepts
- 

SYSTEMS AND SOFTWARE ENGINEERING

09

BUILDING PRODUCTION-GRADE APPLICATIONS

End-to-end execution.

- Backend development
- API construction & integration
- Authentication & security basics
- Performance optimization





END NOTE

Skillpoint programs are demanding by design.

**We don't promise shortcuts. We
promise capability.**

If you complete this program seriously, you won't just clear interviews — you'll understand what you're building.

CONTACT & CONNECT



Chennai



97897 10659



skillpoint.in

Skillpoint

Engineering-Focused Learning Academy

In collaboration with Query