Logicraft.ai

CRAFTING CODE . ARCHITECTING SCALE

AI Engineer Advanced
Complete Syllabus

A 32-week production syllabus for software / data engineers with 2–8 years of experience.

6	32	96
modules	weeks	interactive, hands-on classes

Curriculum focus: self-hosted inference (vLLM, TensorRT-LLM), advanced multi-tenant retrieval, GraphRAG, stateful LangGraph agents, PEFT/QLoRA fine-tuning and enterprise LLMOps, engineered for senior engineers who need to deploy robust, cost-effective and observable AI systems at scale.
How you learn: 3 classes a week · 1.5–2 hour sessions · 96 interactive, hands-on production sessions · production milestone projects and architectural reviews · a four-week capstone with architectural defense.
Duration – 32 weeks (7.5–8 months)

MODULE 01 · WEEKS 01–06 · LOW-LEVEL LLM SYSTEMS, DISTRIBUTED INFERENCE & GPU KERNELS

Focus: Demystifying GPU memory hierarchies, serving engines, token throughput, and high-scale inference.

01 Modern LLM Architecture & Computational Hardware Constraints

Week 01 · Classes 01–03

- **Class 01 · Hardware Architectures** — Tensor Cores, HBM3/HBM3e bandwidth, NVLink topologies, memory-bandwidth bound vs. compute-bound operations.
- **Class 02 · FlashAttention Deep Dive** — IO awareness, GPU SRAM tiling, and avoiding HBM roundtrips in FlashAttention-2 and FlashAttention-3.
- **Class 03 · Modern Attention Topologies** — MHA (Multi-Head), MQA (Multi-Query), and GQA (Grouped-Query Attention) compute and memory trade-offs.

02 KV Cache Mechanics, PagedAttention & Continuous Batching

Week 02 · Classes 04–06

- **Class 04 · The KV Cache Bottleneck** — Memory footprint formulas, dynamic context allocation, and generation bottlenecks.
- **Class 05 · PagedAttention & vLLM Internals** — Virtual memory paging, block table managers, logical vs physical memory, and prefix caching.
- **Class 06 · Dynamic Batching Mechanics** — Static vs continuous batching, prefill vs decode phase scheduling, request queues, and preemption.

03 Distributed Inference Engines (vLLM & TensorRT-LLM)

Week 03 · Classes 07–09

- **Class 07 · Production vLLM Configuration** — Engine flags, max_model_len, GPU memory utilization thresholds, swap space, and chunked prefill.
- **Class 08 · TensorRT-LLM Architecture** — Engine compilation, kernel fusion, weight-only quantization, and low-latency pipeline optimization.
- **Class 09 · Distributed Parallelism Strategies** — Tensor Parallelism (TP) across GPUs vs Pipeline Parallelism (PP) and Megatron-LM communication primitives.

04 Model Quantization & Speculative Decoding

Week 04 · Classes 10–12

- **Class 10 · Post-Training Quantization (PTQ)** — Math and runtime trade-offs of FP8, INT8, INT4, AWQ (Activation-aware Weight Quantization), and GPTQ.
- **Class 11 · Speculative Decoding Systems** — Draft-verification model pipelines, medusa heads, acceptance rates, and token throughput gains.
- **Class 12 · Structured Output Acceleration** — Constrained decoding, Outlines, Guidance, and grammar-guided token masking at the logit level.

05 Cloud & Kubernetes-Based GPU Autoscaling

Week 05 · Classes 13–15

- **Class 13 · Containerizing LLM Runtimes** — NVIDIA Container Toolkit, multi-stage production Dockerfiles, and cold-start reduction techniques.
- **Class 14 · Orchestrating GPU Workloads on K8s** — EKS/GKE device plugins, GPU time-slicing vs MIG (Multi-Instance GPU), and Karpenter provisioning.
- **Class 15 · Metric-Based Dynamic Autoscaling** — Autoscaling on TTFT (Time to First Token), TPOT (Time Per Output Token), and request queue concurrency.

06 Production Milestone Project 1

Week 06 · Classes 16–18

- **Class 16 · Production Lab 1** — Multi-GPU vLLM cluster deployment on Kubernetes with prefix caching and continuous batching enabled.
- **Class 17 · Production Lab 2** — Load testing using Locust and Triton perf-analyzer; benchmarking latency vs. throughput curves under concurrency.
- **Class 18 · Architectural Review** — Code review, architectural defense, failure-recovery drills, and milestone sign-off.

MODULE 02 · WEEKS 07–12 · ADVANCED ENTERPRISE RETRIEVAL & VECTOR ARCHITECTURES

Focus: Scaling beyond basic vector search to deterministic, multi-tenant enterprise knowledge retrieval.

07 Deterministic Ingestion & Semantic Chunking

Week 07 · Classes 19–21

- **Class 19 · Ingestion Failure Modes** — Pitfalls of fixed-size chunking, boundary fragmentation, and document layout destruction.
- **Class 20 · Semantic & Structure-Aware Chunking** — Embedding distance splitting, sliding windows, and Markdown/HTML hierarchy preservation.
- **Class 21 · Agentic & Contextual Chunking** — Contextual chunk headers, pre-pending LLM-generated document/section summaries, and metadata injection.

08 Hybrid Search & Reciprocal Rank Fusion (RRF)

Week 08 · Classes 22–24

- **Class 22 · Dense vs Sparse Retrieval** — Dense vector embeddings vs sparse keyword algorithms (BM25, SPLADE) and vocabulary mismatch.
- **Class 23 · Reciprocal Rank Fusion (RRF)** — RRF mathematical formulations, dynamic k tuning, and convex score combination techniques.
- **Class 24 · Cross-Encoder Re-rankers** — Cohere Rerank, BGE-Reranker, bi-encoder vs cross-encoder cost-latency trade-offs, and top-k filtering.

09 Vector Database Internals at Scale

Week 09 · Classes 25–27

- **Class 25 · Vector Indexing Math** — Flat vs IVF (Inverted File) vs HNSW (Hierarchical Navigable Small World); tuning M and efConstruction.
- **Class 26 · PgVector in Enterprise Production** — HNSW indexing in PostgreSQL, IVFFlat, halfvec (FP16), vacuuming, and memory allocation.
- **Class 27 · Milvus & Qdrant Enterprise Clustering** — Partitioning, payload indexing, write-ahead logging (WAL), and distributed query coordination.

10 Multi-Tenancy & Row-Level Security (RLS)

Week 10 · Classes 28–30

- **Class 28 · Multi-Tenancy Isolation Models** — Schema-per-tenant vs collection-per-tenant vs metadata-filtered partitions.
- **Class 29 · Row-Level Security (RLS) in RAG** — Syncing Active Directory/OAuth roles into vector payloads and deterministic filter enforcement.
- **Class 30 · Context-Window Optimizations** — Selective token pruning, contextual compression, summarization trees, and 'lost-in-the-middle' mitigation.

11 GraphRAG & Relational Knowledge Retrieval

Week 11 · Classes 31–33

- **Class 31 · Limits of Semantic Similarity** — Why pure vectors fail at multi-hop reasoning; introduction to Knowledge Graphs for RAG.
- **Class 32 · Graph Extraction Pipelines** — Automated entity-relation extraction using LLMs into Neo4j/Amazon Neptune.
- **Class 33 · Hybrid GraphRAG Implementation** — Combining vector search with graph traversal queries (Cypher + Vector similarity).

12 Production Milestone Project 2

Week 12 · Classes 34–36

- **Class 34 · Production Lab 1** — Architecting a multi-tenant enterprise RAG pipeline supporting 100K+ complex PDF/DOCX files.
- **Class 35 · Production Lab 2** — Implementing GraphRAG + RRF reranking with strict metadata RBAC validation.
- **Class 36 · Evaluation & Defense** — Benchmarking retrieval against golden datasets using Hit Rate@K and MRR; architectural defense.

MODULE 03 · WEEKS 13–18 · STATEFUL AGENTIC SYSTEMS & MODEL CONTEXT PROTOCOL (MCP)

Focus: Complex multi-agent coordination, deterministic execution graphs, and tool protocols.

13 Deep Dive into State Graphs with LangGraph

Week 13 · Classes 37–39

- **Class 37 · Cyclic Execution Graphs** — Beyond linear DAGs: cyclical execution loops, conditional routers, and branch merges in LangGraph.
- **Class 38 · State Reducer Architecture** — State management, channel reducers, typed schemas, and immutability patterns in long-running agents.
- **Class 39 · Production Checkpointing** — PostgreSQL and Redis state checkpointers for fault recovery, session continuity, and pause/resume.

14 Human-in-the-Loop & Time-Travel Debugging

Week 14 · Classes 40–42

- **Class 40 · Human-in-the-Loop (HITL) Workflows** — Manual approval gates, review queues, webhook interruptions, and escalation thresholds.
- **Class 41 · State Editing & Time-Travel** — Rewinding graph execution, editing past agent state, branching workflows, and replaying history.
- **Class 42 · Subgraphs & Agent Decomposition** — Modular agent architecture, nested subgraphs, and context isolation across agent teams.

15 Multi-Agent Collaboration & Supervision

Week 15 · Classes 43–45

- **Class 43 · Multi-Agent Topologies** — Supervisor patterns, hierarchical command trees, and choreographed peer-to-peer collaboration.
- **Class 44 · Shared vs Isolated Agent Memory** — Working memory scratchpads, cross-agent message passing, and token window conservation.
- **Class 45 · Deadlock & Loop Circuit Breakers** — Circular dependency detection, maximum iteration enforcement, and token exhaustion circuit breakers.

16 Model Context Protocol (MCP) in Production

Week 16 · Classes 46–48

- **Class 46 · MCP Architecture & Specifications** — Client-server protocol specification, JSON-RPC messaging, stdio vs Server-Sent Events (SSE) transports.
- **Class 47 · Building Production MCP Servers** — Exposing databases, microservices, and file management tools safely through MCP endpoints.
- **Class 48 · Building Enterprise MCP Hosts** — Dynamic tool discovery, access permission enforcement, and authentication propagation.

17 Determinism & Schema Governance

Week 17 · Classes 49–51

- **Class 49 · Enforcing Deterministic Outputs** — Pydantic v2 schemas, Instructor library, automatic self-correction loops, and structured validation.
- **Class 50 · Agent Evaluation Frameworks** — Trajectory evaluation, tool-selection precision/recall, and step-efficiency scoring.
- **Class 51 · Agent Sandboxing & Execution Safety** — Running untrusted agent-generated code inside isolated microVMs (gVisor/Firecracker/Docker).

18 Production Milestone Project 3

Week 18 · Classes 52–54

- **Class 52 · Production Lab 1** — Building a stateful multi-agent automated code review and QA system using LangGraph and custom MCP servers.
- **Class 53 · Production Lab 2** — Implementing PostgreSQL checkpointers, HITL approval webhooks, and sandboxed code execution.
- **Class 54 · Review & Chaos Testing** — Chaos testing agent recovery under node failure and comprehensive code review.

MODULE 04 · WEEKS 19–24 · DOMAIN ADAPTATION, SUPERVISED FINE-TUNING & ALIGNMENT

Focus: When prompting and RAG fall short: custom model training, PEFT, and preference alignment.

19 Dataset Engineering & Synthetic Generation

Week 19 · Classes 55–57

- **Class 55 · Fine-Tuning Decision Matrix** — When to Fine-Tune vs RAG vs Prompt Engineering: technical boundaries, latency, and ROI modeling.
- **Class 56 · Enterprise Dataset Curation** — Data cleansing, MinHash LSH deduplication, decontamination, and format standardization.
- **Class 57 · Synthetic Data Generation** — LLM-as-a-Judge pipelines for dataset expansion, rejection sampling, and diversity filtering.

20 Parameter-Efficient Fine-Tuning (PEFT)

Week 20 · Classes 58–60

- **Class 58 · LoRA Mathematical Foundations** — Low-Rank Adaptation math, intrinsic dimensionality, rank r, and scaling factor α.
- **Class 59 · QLoRA & 4-Bit Quantization** — NormalFloat4 (NF4), double quantization, and paged optimizers for memory-constrained training.
- **Class 60 · Accelerated Fine-Tuning with Unsloth** — Custom CUDA kernels, gradient checkpointing, and high-throughput SFT using Hugging Face TRL.

21 Training Infrastructure & Supervised Fine-Tuning (SFT)

Week 21 · Classes 61–63

- **Class 61 · Hyperparameter Tuning for LLMs** — Cosine learning rate schedules with warmup, weight decay, batch sizes, and gradient accumulation.
- **Class 62 · Distributed SFT Infrastructure** — PyTorch FSDP (Fully Sharded Data Parallel) and DeepSpeed ZeRO-1/2/3 strategies for multi-GPU training.
- **Class 63 · Training Monitoring & Diagnostics** — Loss curve analysis, perplexity tracking, gradient norm monitoring, and catastrophic forgetting prevention.

22 Alignment: DPO & ORPO

Week 22 · Classes 64–66

- **Class 64 · Preference Data Architecture** — Constructing paired preference datasets (chosen vs rejected responses) for domain alignment.
- **Class 65 · Direct Preference Optimization (DPO)** — Mathematical derivation of DPO vs RLHF/PPO; eliminating the reward model overhead.
- **Class 66 · ORPO & Safe Alignment** — Odds Ratio Preference Optimization and instruction-following alignment without degradation.

23 Merging, Exporting & Multi-LoRA Serving

Week 23 · Classes 67–69

- **Class 67 · Weight Merging & Precision Handling** — Merging LoRA adapters into base weights, 16-bit de-quantization, and GGUF/AWQ export.
- **Class 68 · Dynamic Multi-LoRA Serving** — Configuring vLLM to serve dozens of specialized LoRA adapters concurrently on a single base model.
- **Class 69 · Comprehensive Model Benchmarking** — Running task-specific evaluation suites, MMLU regression checks, and domain accuracy metrics.

24 Production Milestone Project 4

Week 24 · Classes 70–72

- **Class 70 · Production Lab 1** — Fine-tuning an open-weights model on proprietary structured domain documents using QLoRA.
- **Class 71 · Production Lab 2** — Aligning the model with DPO, validating benchmark regressions, and deploying as a multi-LoRA endpoint on vLLM.
- **Class 72 · Model Evaluation & Sign-Off** — Final evaluation against unseen validation datasets and checkpoint auditing.

MODULE 05 · WEEKS 25–28 · LLMOPS, CONTINUOUS EVALUATION, SECURITY & OBSERVABILITY

Focus: Industrializing AI deployments with enterprise security, telemetry, and automated CI/CD.

25 Continuous Evaluation-Driven Development (EDD)

Week 25 · Classes 73–75

- **Class 73 · Synthetic Benchmark Suites** — Creating representative golden ground-truth test datasets for continuous evaluation.
- **Class 74 · RAG Evaluation Frameworks** — Deep dive into Ragas and DeepEval: Faithfulness, Answer Relevance, Context Precision, and Recall.
- **Class 75 · Custom LLM Judges & CI Evals** — Designing calibrated LLM-as-a-Judge evaluators and integrating them into automated test suites.

26 Distributed Observability & Telemetry

Week 26 · Classes 76–78

- **Class 76 · OpenTelemetry for GenAI** — Semantic conventions for GenAI, spans, traces, and metadata propagation across microservices.
- **Class 77 · Full-Stack Tracing Tools** — Deploying LangSmith and Arize Phoenix: tracing nested tool executions, latency sinks, and error rates.
- **Class 78 · Production Metric Dashboards** — Tracking token usage, cost-per-user, TTFT, TPOT, and latency percentiles in Prometheus and Grafana.

27 AI Security & OWASP Top 10 for LLMs

Week 27 · Classes 79–81

- **Class 79 · Prompt Injection Defense** — Detecting and preventing direct/indirect prompt injection, jailbreaks, and payload smuggling.
- **Class 80 · Guardrail Frameworks** — Implementing NeMo Guardrails, Llama Guard, and regex/PII masking engines in the inference path.
- **Class 81 · Model Supply Chain Security** — Safetensors validation, poisoned dataset prevention, and API key token rotation architectures.

28 LLMOps CI/CD Pipelines & Cost Governance

Week 28 · Classes 82–84

- **Class 82 · Automated Prompt CI/CD** — GitHub Actions workflows for prompt regression testing and threshold validation before deployment.
- **Class 83 · Semantic Caching & Cost Routing** — Redis semantic caching, prompt caching, model cascades (routing queries by complexity), and budget throttles.
- **Class 84 · Canary & Blue/Green Deployments** — Zero-downtime routing strategies for new prompt templates and fine-tuned model versions.

MODULE 06 · WEEKS 29–32 · CAPSTONE ARCHITECTURE & DEFENSE

Focus: End-to-end delivery of an enterprise-grade AI system with strict SLA and cost boundaries.

29 System Scoping, SLA Definition & Architecture Blueprint

Week 29 · Classes 85–87

- **Class 85 · Non-Functional Scoping** — Defining strict latency floors (P95/P99), dollar cost ceilings, and uptime targets.
- **Class 86 · Retrieval & Ingestion Blueprint** — Architecting high-throughput data pipelines and hybrid vector/graph schemas.
- **Class 87 · Serving & High Availability Blueprint** — Architecting GPU clusters, fallback routing, and checkpointer resilience.

30 Implementation & Infrastructure Provisioning

Week 30 · Classes 88–90

- **Class 88 · Core Engine Deployment** — Deploying the multi-agent execution core and low-latency inference cluster.
- **Class 89 · Retrieval & Tool Integration** — Integrating RLS-compliant hybrid retrieval and custom MCP microservices.
- **Class 90 · Telemetry & Guardrail Hooks** — Wiring OpenTelemetry distributed tracing and bidirectional guardrails.

31 Stress Testing, Chaos Engineering & Hardening

Week 31 · Classes 91–93

- **Class 91 · Chaos Testing & Node Failures** — High-concurrency load testing and GPU node failure injection under production loads.
- **Class 92 · Adversarial Red-Teaming** — Penetration testing with automated jailbreak payloads and indirect injection vectors.
- **Class 93 · Cost & Performance Tuning** — Maximizing semantic cache hit rates and tuning continuous batching parameters.

32 Production Rollout & Architectural Defense

Week 32 · Classes 94–96

- **Class 94 · Canary Deployment Execution** — Canary traffic routing, real-time drift monitoring, and automated rollback validation.
- **Class 95 · Boardroom Technical Defense** — Presenting and defending system architecture, trade-offs, and metrics to senior engineering evaluators.
- **Class 96 · Final Audit & Graduation** — Comprehensive code audit, architecture sign-off, and program completion.