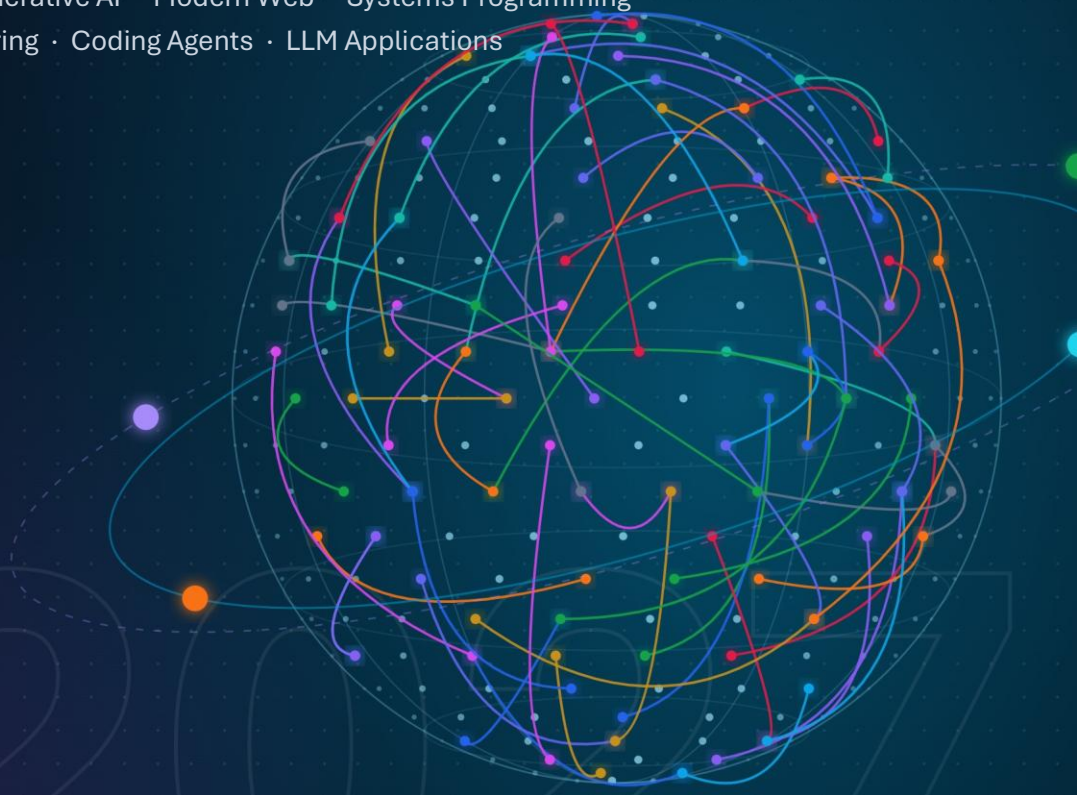
DEEPCLOUDLABS

TRAINING & CONSULTANCY SERVICES

Training & Consultancy Services Catalog 2027

Cloud · Software Architecture · Java SE 27 · Microservices & DevOps
Data Engineering · AI & Generative AI · Modern Web · Systems Programming
AI-Native Software Engineering · Coding Agents · LLM Applications



67 courses · 11 technology domains · 2 bootcamps · 17 consultancy services

Document No: DCL-CRS-01 · Version: 3.1 · Version Date: 18.10.2026 · www.deepcloudlabs.com

Contents

How to Use This Catalog 3
About DEEPCLOUDLABS 4
What's New in the 2027 Edition 5
Our References 6
How We Deliver Training 7
Portfolio at a Glance 8
Course Index 9
Strategic Technology Areas 11
Security Across the Curriculum 12
Learning Paths & Bootcamps 13
01 Cloud Platforms & Architecture 15
02 Microservices, DevOps & Distributed Systems 20
03 Java SE 27 31
04 Spring Ecosystem & Kotlin 42
05 Jakarta EE 11 47
06 Python, Data Engineering & Machine Learning 51
07 Deep Learning & Generative AI 57
08 AI-Native Software Engineering 70
09 JavaScript & Modern Web 90
10 C/C++ & Linux Systems 96
11 Executive Programs 102
Application Development Consultancy 107
Project Management Consultancy 108
Private Group Classes 109
Company Information 110

DOCUMENT CONTROL

Document	Training & Consultancy Services Catalog — 2027 Edition
Document No	DCL-CRS-01
Version	3.1 — supersedes version 3.0 of 09.10.2026
Version Date	18.10.2026
Publisher	DEEPCLOUDLABS · İstanbul, Türkiye · www.deepcloudlabs.com

READING GUIDE

How to Use This Catalog

A consistent structure on every page — so you can compare courses, plan learning paths and find what your teams need in minutes.



CATALOG STRUCTURE

PART I	**About DEEPCLOUDLABS**	Who we are, what is new in 2027, our references and how we deliver training.
PART II	**Training Portfolio**	Portfolio overview, learning paths & bootcamps, and eleven color-coded technology domains with full course pages.
PART III	**Consultancy Services**	Application development and project management consultancy.
PART IV	**Engage with Us**	Private group classes, contact details and company information.

COURSE LEVELS

Introductory	No or little prior experience in the topic.
Intermediate	Working knowledge of the fundamentals.
Advanced	Solid practical experience; deep-dive content.
Executive	Leaders and managers; no technical background needed.

COURSE CODE RANGES

DCL-1xx	C/C++, Linux, Python, Kotlin
DCL-2xx	Java SE 27
DCL-3xx	Web, Jakarta EE, Spring, microservices, messaging
DCL-4xx	Data, AI, executive programs, edge AI
DCL-5xx	Cloud platforms (new in 2027)
DCL-6xx	AI-native software engineering (new in 2027)

Every course can be delivered as published, slightly modified, or completely customized for a private group. Durations are given in training days. Course codes are stable identifiers: use them when requesting proposals or schedules. AI-native courses (DCL-6xx and DCL-454) are followed by a course specification page with measurable learning outcomes, practical components, assessment, expected competencies and recommended textbooks.

PART I · ABOUT US

About DEEPCLOUDLABS

Instructor-led technical training and engineering consultancy for the information technology industry.

DEEPCLOUDLABS offers instructor-led technical classroom training for the information technology industry. Our immersive, instructor-led ecosystem is designed to bridge the talent gap and accelerate digital transformation through high-impact technical mastery. Our proven training solution helps companies enhance organizational capabilities by empowering their employees with technical skills. Our fully configured lab environment gives participants hands-on access to the applications taught in our classrooms, enabling them to learn on their own schedules, and our mentoring service helps participants learn at their own pace with our highly skilled instructors in their workplace.

Research & development focus

DEEPCLOUDLABS is an innovative company whose research and development teams focus on all aspects of the following topics:

- **Cloud Computing**
- **Big Data Analytics**
- **Artificial Intelligence and Machine Learning**
- **Image and Video Analytics**
- **Blockchain and Cryptocurrency**
- **Algorithmic and High-Frequency Trading**
- **Project Management and Software Process Enhancement**

Professional services

DEEPCLOUDLABS services provide access to the talent and systems you need to innovate faster and deliver real business value. We offer a full range of professional services:

Consulting
Advice, expertise and consulting services for blockchain technology, AI and machine learning, and software development.

Corporate Training
Hands-on training for real-world problems: in-house and external corporate training, seminars, workshops and talks.

Research & Development
Innovation and prototyping around data analytics, AI and machine learning, and blockchain technologies.

Software Development
Agile implementation of advanced big data analytics applications, using cognitive technology to increase accuracy and productivity.

Outsourced Developers
Hire our talented developers for a defined period.

OUR TEAM

Our engineering team comprises specialists holding Ph.D. and M.Sc. degrees, with the engineering experience to innovate and to transform those innovations into products.

PART I · 2027 UPDATES

What's New in the 2027 Edition

A redesigned catalog and a curriculum upgraded for the technologies enterprises are adopting in 2027.

08 · AI-NATIVE ENGINEERING

AI-native software engineering

New domain 08 (version 3.1): coding agents, specification-driven development, AI-assisted testing, LLM applications, RAG, agents, LLMOps and AI security.

11 · EXECUTIVE

AI-native delivery for leaders

New one-day program DCL-454: operating model, impact metrics, policy and governance for AI-assisted software engineering.

03 · JAVA SE 27

Java SE 27 curriculum

All Java courses upgraded to Java SE 27 (GA 15 Sep 2026): compact object headers, G1 everywhere, post-quantum TLS, structured concurrency.

04 · SPRING & KOTLIN

Spring Framework 7 & Boot 4

Spring courses aligned with the November 2025 generation: API versioning, built-in resilience, JSpecify null-safety, Jackson 3.

07 · DEEP LEARNING & GENAI

LLM engineering

Generative AI courses add RAG, vector databases, AI agents with tool calling (MCP), LoRA fine-tuning, evaluation and guardrails.

02 · MICROSERVICES & DEVOPS

Kafka 4 · RabbitMQ 4

Messaging courses updated to KRaft-only Kafka 4 and to RabbitMQ 4 quorum queues and streams.

01 · CLOUD

Cloud catalog codes

Cloud courses now carry DCL-5xx codes and full module outlines; terminology updated (Entra ID, Cloud Run functions).

10 · C/C++ & LINUX

Modern C++ (C++23/26)

C++ courses add C++26 previews: std::execution, hazard pointers, RCU, plus std::expected and std::print.

VERSION UPGRADE SUMMARY

TECHNOLOGY	2026 EDITION	2027 EDITION	COURSES
Java SE	Java SE 26	Java SE 27	All Java courses
JUnit	JUnit 5	JUnit 6	DCL-215
Spring Framework / Boot	Spring 6 / Boot 3	Spring 7 / Boot 4	DCL-370, DCL-374, DCL-350
Jakarta EE	Jakarta EE 10 / 11	Jakarta EE 11 throughout	DCL-342, DCL-364, DCL-390
Apache Kafka	Not version-specific	Kafka 4 (KRaft only)	DCL-380, DCL-358
RabbitMQ	Not version-specific	RabbitMQ 4 (quorum queues, streams)	DCL-382
ECMAScript	ES6–ES13	ES2015–ES2025	DCL-302, DCL-304
Vue tooling	Vue CLI	Vite / create-vue	DCL-318
C++	C++23	C++23 with C++26 preview	DCL-112, DCL-113, DCL-115
Apache Spark	Spark Streaming	Spark 4, Structured Streaming, lakehouse	DCL-400
Edge AI hardware	Jetson Nano	Jetson Orin Nano + TensorRT	DCL-460
Cloud terminology	Azure AD, Cloud Functions	Microsoft Entra ID, Cloud Run functions	DCL-502, DCL-504, DCL-506
On-device ML	TensorFlow Lite	LiteRT (formerly TensorFlow Lite)	DCL-428

Also new: standardized course pages (overview, audience, prerequisites, outcomes, modules, duration, delivery, technologies), course-specific technical illustrations, technology maps for every domain, role-based learning paths and a cross-reference of strategic technology areas. AI-native courses add a course specification page with learning outcomes, practical components, assessment and recommended textbooks.

PART I · TRUSTED BY

Our References

Companies to which we have delivered training courses and consultancy services.

İŞ YATIRIM | GE | softtech | SIEMENS
ERICSSON | Garanti BBVA | Türk Telekom Değerli Hissettirir | HAVELSAN
TURKISH AIRLINES | EPİAŞ | aselsan | TEB
ULAK HABERLEŞME A.Ş. | TAKAS İSTANBUL | TACİRLER MENKUL DEĞERLER A.Ş. | TOFAŞ TÜRK OTOMOBİL FABRİKASI A.Ş.
LINKTERA | arçelik | IBTECH | KoçSistem
TÜBİTAK BİLGEM | BKM BANKALARARASI KART MERKEZİ | ECZACIBAŞI BİLİŞİM | Türkiye Sigorta Birliği

Logos are trademarks of their respective owners and are shown to identify organizations to which DEEPCLOUDLABS has delivered services.

PART I · DELIVERY MODELS

How We Deliver Training

Hands-on, instructor-led learning — delivered where and how your organization needs it.

Instructor-led classroom training

Immersive, instructor-led technical training designed to bridge the talent gap through high-impact technical mastery.

Fully configured lab environment

Participants get hands-on access to the applications taught in class, so they can learn on their own schedule.

Mentoring at your workplace

Our highly skilled instructors mentor participants at their own pace, in their own workplace.

Private group classes

Train an entire team or department with one unified learning experience — everyone gains the same knowledge and skills.

Flexible locations

At any of our locations, on-site at your offices, or at a location of your choice.

CUSTOMIZATION SPECTRUM

01
Off the shelf
Delivered exactly as published in this catalog.

02
Slightly modified
Module emphasis, examples and labs adapted to your stack.

03
Completely customized
A program designed around your organization's learning initiatives.

EVERY COURSE PAGE SHOWS

- Duration in training days and the recommended level
- Delivery: instructor-led with hands-on labs, at our locations, on-site or at a location of your choice
- Target audience, prerequisites and related courses to plan a learning path

PART II · TRAINING PORTFOLIO

Portfolio at a Glance

Eleven technology domains, one consistent standard of hands-on, instructor-led training.

67	11	218	2	17
COURSES	TECHNOLOGY DOMAINS	TRAINING DAYS IN TOTAL	BOOTCAMPS	CONSULTANCY SERVICES

TECHNOLOGY DOMAINS

01 4 COURSES · 19 DAYS

Cloud Platforms & Architecture

AWS · EC2 · S3 · IAM · Bicep

02 9 COURSES · 27 DAYS

Microservices, DevOps & Distributed Systems

DDD · Spring Boot 4 · Apache Kafka 4 · JUnit 6 · Jenkins

03 9 COURSES · 29 DAYS

Java SE 27

Records · Virtual threads · Streams & Gatherers · G1 · Clean Architecture

04 4 COURSES · 13 DAYS

Spring Ecosystem & Kotlin

IoC / DI · Spring MVC · Spring Data JPA · Actuator · Kotlin 2

05 3 COURSES · 10 DAYS

Jakarta EE 11

Servlet 6.1 · REST 4.0 · CDI 4.1 · Persistence 3.2 · Messaging 3.1

06 5 COURSES · 17 DAYS

Python, Data Engineering & Machine Learning

Python 3 · NumPy · Hadoop · Apache Iceberg · scikit-learn

07 11 COURSES · 39 DAYS

Deep Learning & Generative AI

PyTorch 2 · CNNs · Transformers · RAG · TensorRT

08 9 COURSES · 30 DAYS

AI-Native Software Engineering

Claude Code · Context engineering · LLM APIs · MCP · Evaluation

09 5 COURSES · 15 DAYS

JavaScript & Modern Web

ES2025 · Node.js LTS · Signals · React 19 · Vue 3

10 5 COURSES · 16 DAYS

C/C++ & Linux Systems

C17/C23 · C++23 · std::jthread · POSIX · GCC

11 3 COURSES · 3 DAYS

Executive Programs

Data-driven decisions · Design thinking · Generative AI

PART II · ALL COURSES

Course Index

Every course in the 2027 catalog with level, duration and page. Codes link to the course page in the digital edition.

CODE	COURSE	LEVEL	DAYS	PAGE
01 CLOUD PLATFORMS & ARCHITECTURE				
DCL-500	AWS Essentials	Introductory	3	16
DCL-502	Microsoft Azure Fundamentals	Introductory	3	17
DCL-504	Azure Cloud for Architects	Advanced	10	18
DCL-506	Google Cloud Platform Essentials	Introductory	3	19
02 MICROSERVICES, DEVOPS & DISTRIBUTED SYSTEMS				
DCL-350	Implementing Microservices Architecture using Spring Cloud	Intermediate	5	22
DCL-351	Microservices Testing	Intermediate–Advanced	4	23
DCL-352	Domain-Driven Design Essentials	Intermediate	2	24
DCL-353	DevOps Engineering	Intermediate	3	25
DCL-355	Microservices Patterns with Examples in Spring Boot	Advanced	4	26
DCL-356	Cloud Architecture Patterns	Intermediate	2	27
DCL-358	Implementing Event-Driven Microservices Architecture using Spring Boot and Apache Kafka	Advanced	3	28
DCL-380	Apache Kafka: Architecture and Development	Intermediate	2	29
DCL-382	RabbitMQ: Architecture and Administration	Intermediate	2	30
03 JAVA SE 27				
DCL-200	Java Performance Tuning and Optimization	Advanced	3	33
DCL-204	Java SE 27 Programming	Introductory	5	34
DCL-206	Advanced Java Development with Modern Design and Programming Paradigms	Advanced	4	35
DCL-208	Clean Architecture and Code (Java SE and Spring Boot)	Intermediate	3	36
DCL-210	Advanced Java Programming	Advanced	3	37
DCL-215	Test-Driven Development with JUnit 6	Intermediate	3	38
DCL-220	Object-Oriented Programming Principles and Design Patterns with Java SE 27	Intermediate	3	39
DCL-235	Effective Java Programming	Advanced	3	40
DCL-252	New Features in Java SE 8–27	Intermediate	2	41
04 SPRING ECOSYSTEM & KOTLIN				
DCL-168	Kotlin Programming	Intermediate	3	43
DCL-370	Spring Framework 7	Intermediate	4	44
DCL-374	Developing Spring Boot 4 Applications	Intermediate	3	45
DCL-376	Spring Data	Intermediate	3	46
05 JAKARTA EE 11				
DCL-342	Jakarta Persistence 3.2	Intermediate	3	48
DCL-364	Best Practices in Jakarta EE 11	Advanced	3	49
DCL-390	Developing Enterprise Applications on Jakarta EE 11	Intermediate	4	50
06 PYTHON, DATA ENGINEERING & MACHINE LEARNING				
DCL-160	Python Programming	Introductory	4	52
DCL-162	Advanced Python Programming	Intermediate	3	53

CODE	COURSE	LEVEL	DAYS	PAGE
DCL-400	Big Data Essentials	Introductory	3	54
DCL-402	Data Analytics using Python	Introductory	3	55
DCL-410	Practical Machine Learning using Python	Intermediate	4	56
07 DEEP LEARNING & GENERATIVE AI				
DCL-422	Deep Learning for Computer Vision	Intermediate	4	59
DCL-424	Deep Learning for Medical Image Analysis	Intermediate	4	60
DCL-426	Deep Learning with PyTorch	Intermediate	4	61
DCL-428	Deep Learning with TensorFlow	Intermediate	4	62
DCL-430	Foundation of Deep Learning with Backbone Architecture Papers	Advanced	4	63
DCL-432	Deep Learning with Backbone Learning Paradigm Papers	Advanced	4	64
DCL-434	Introduction to Generative AI with Python	Intermediate	3	65
DCL-436	Advanced Generative AI with Python	Advanced	4	66
DCL-438	Introduction to ChatGPT	Introductory	2	67
DCL-440	Introduction to Image Generation with Stable Diffusion	Introductory	2	68
DCL-460	Edge Computing: Deployment & Inference with NVIDIA Jetson	Intermediate	4	69
08 AI-NATIVE SOFTWARE ENGINEERING				
DCL-600	AI-Assisted Software Engineering	Intermediate	3	72
DCL-602	Specification-Driven Development with AI Agents	Intermediate–Advanced	3	74
DCL-604	Agentic Software Engineering	Advanced	4	76
DCL-606	AI-Assisted Software Testing and Quality Engineering	Intermediate	3	78
DCL-610	Generative AI Application Development	Intermediate	4	80
DCL-612	Retrieval-Augmented Generation for Enterprise Applications	Advanced	3	82
DCL-614	AI Agent and Multi-Agent Systems Engineering	Advanced	4	84
DCL-616	LLMOps: Evaluating and Operating Production AI Systems	Advanced	3	86
DCL-618	Secure and Responsible AI Application Engineering	Advanced	3	88
09 JAVASCRIPT & MODERN WEB				
DCL-302	Node.js Programming	Intermediate	3	91
DCL-304	Advanced JavaScript Programming	Advanced	3	92
DCL-305	Developing Angular Applications	Intermediate	3	93
DCL-306	Developing React Applications	Intermediate	3	94
DCL-318	Developing Vue 3 Applications	Intermediate	3	95
10 C/C++ & LINUX SYSTEMS				
DCL-100	C Programming	Introductory	3	97
DCL-112	Object-Oriented Programming in Modern C++ (C++23/26)	Intermediate	4	98
DCL-113	Functional Programming in Modern C++ (C++23/26)	Advanced	2	99
DCL-115	Multi-Threaded Programming in Modern C++ (C++23/26)	Advanced	3	100
DCL-140	Linux System Programming	Advanced	4	101
11 EXECUTIVE PROGRAMS				
DCL-450	Big Data and Artificial Intelligence for Executives	Executive	1	103
DCL-452	Cloud Computing and Microservice Architecture for Executives	Executive	1	104
DCL-454	AI-Native Software Delivery for Technology Leaders	Executive	1	105

PART II · CROSS-REFERENCE

Strategic Technology Areas

Where each strategic technology area is covered across the portfolio — useful for building programs around a transformation initiative. Click a code to open the course.

Cloud platforms & architecture

5 COURSES · 19 DAYS

- DCL-504 Azure for Architects
- DCL-500 AWS Essentials
- DCL-506 Google Cloud Essentials
- DCL-356 Cloud Architecture Patterns
- DCL-452 Cloud for Executives

Containers, Kubernetes & CI/CD

4 COURSES · 22 DAYS

- DCL-353 DevOps Engineering
- DCL-350 Spring Cloud Microservices
- DCL-351 Microservices Testing
- DCL-504 Azure for Architects

Software architecture

5 COURSES · 15 DAYS

- DCL-208 Clean Architecture & Code
- DCL-206 Modern Java Paradigms
- DCL-220 OOP & Design Patterns
- DCL-352 Domain-Driven Design
- DCL-364 Jakarta EE Best Practices

Microservices

4 COURSES · 16 DAYS

- DCL-350 Spring Cloud Microservices
- DCL-355 Microservices Patterns
- DCL-358 Event-Driven Microservices
- DCL-351 Microservices Testing

Distributed systems & messaging

4 COURSES · 9 DAYS

- DCL-380 Apache Kafka
- DCL-382 RabbitMQ
- DCL-358 Event-Driven Microservices
- DCL-356 Cloud Architecture Patterns

Java

6 COURSES · 20 DAYS

- DCL-204 Java SE 27 Programming
- DCL-206 Modern Java Paradigms
- DCL-200 Java Performance Tuning
- DCL-210 Advanced Java
- DCL-235 Effective Java
- DCL-252 New Features Java 8–27

Security

6 COURSES · 27 DAYS

- DCL-210 Advanced Java
- DCL-370 Spring Framework 7
- DCL-353 DevOps Engineering
- DCL-504 Azure for Architects
- DCL-390 Jakarta EE 11 Development
- DCL-618 Secure & Responsible AI

Data engineering

4 COURSES · 11 DAYS

- DCL-400 Big Data Essentials
- DCL-402 Data Analytics (Python)
- DCL-162 Advanced Python
- DCL-380 Apache Kafka

AI / machine learning

5 COURSES · 20 DAYS

- DCL-410 Practical Machine Learning
- DCL-422 DL for Computer Vision
- DCL-426 DL with PyTorch
- DCL-428 DL with TensorFlow
- DCL-460 Edge AI on Jetson

Generative AI

4 COURSES · 11 DAYS

- DCL-434 Intro to Generative AI
- DCL-436 Advanced Generative AI
- DCL-438 Introduction to ChatGPT
- DCL-440 Stable Diffusion

AI-native software engineering

5 COURSES · 14 DAYS

- DCL-600 AI-Assisted Engineering
- DCL-602 Spec-Driven Development
- DCL-604 Agentic Engineering
- DCL-606 AI-Assisted Testing & QE
- DCL-454 AI-Native Delivery for Leaders

LLM applications & agents

5 COURSES · 17 DAYS

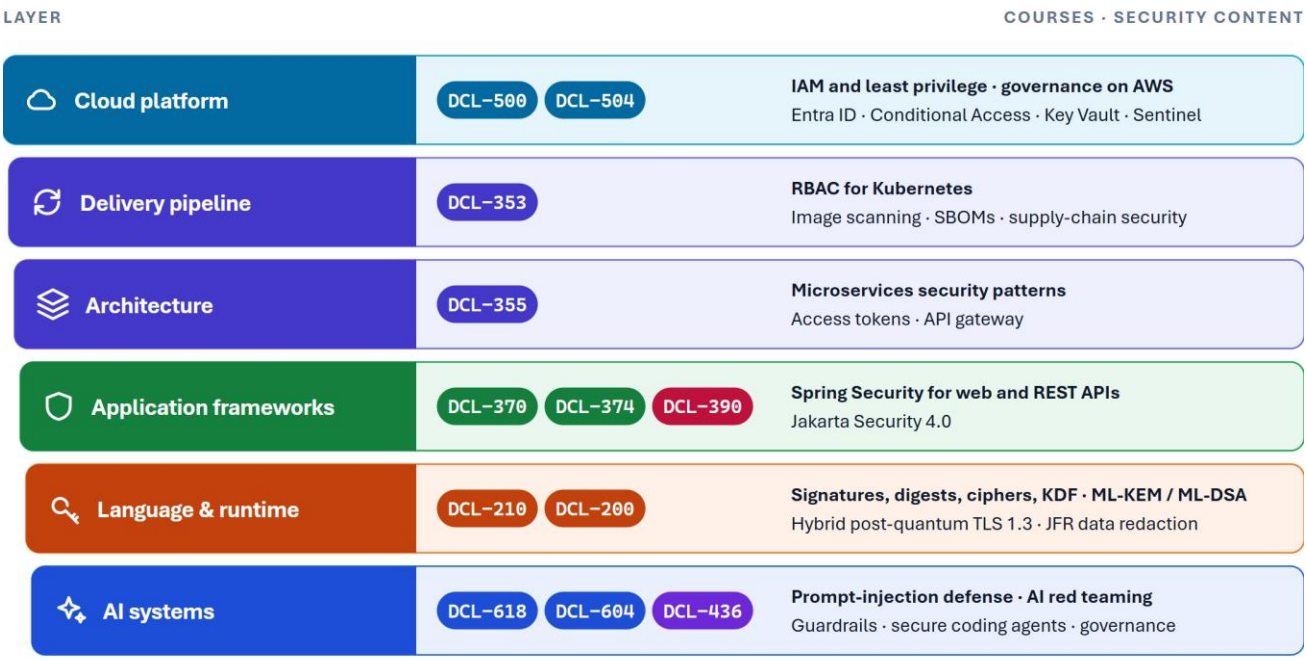
- DCL-610 GenAI App Development
- DCL-612 Enterprise RAG
- DCL-614 Agent & Multi-Agent Systems
- DCL-616 LLMOps
- DCL-618 Secure & Responsible AI

Coverage bar: one block per course, colored by technology domain.

PART II · CROSS-REFERENCE

Security Across the Curriculum

Security is embedded throughout the portfolio — from cloud platforms and delivery pipelines down to cryptography in the Java runtime — and AI systems, with their new attack surfaces, also have a dedicated course (DCL-618).



SECURITY CONTENT BY COURSE

COURSE	SECURITY CONTENT
DCL-210	Cryptography APIs, post-quantum algorithms (ML-KEM, ML-DSA), hybrid PQ key exchange for TLS 1.3
DCL-370 / DCL-374	Spring Security; securing REST APIs
DCL-355	Microservices security patterns (access tokens, API gateway)
DCL-353	RBAC, image scanning, SBOMs and supply-chain security
DCL-500	IAM, least privilege, governance and security best practices on AWS
DCL-504	Entra ID, Conditional Access, Key Vault and Microsoft Sentinel
DCL-390	Jakarta Security 4.0
DCL-618	OWASP Top 10 for LLM and agentic applications, prompt-injection defense, AI red teaming, NIST AI RMF, ISO/IEC 42001, EU AI Act
DCL-600 / DCL-604	Secure use of coding agents: secrets, sandboxing, least-privilege permissions, hallucinated packages, review gates
DCL-612 / DCL-614	Permission-aware retrieval; MCP servers with least-privilege scopes and human approval of high-impact actions
DCL-434 / DCL-436	Responsible and secure GenAI, LLM guardrails
DCL-200	JFR in-process data redaction of sensitive information (Java SE 27)

PART II · PLAN YOUR JOURNEY

Learning Paths & Bootcamps

Role-based sequences built from catalog courses — and two integrated bootcamps for intensive, end-to-end programs.

Each path lists courses in the recommended order; course colors indicate the technology domain. Paths can be shortened, combined or extended for a private group. Bootcamps follow the course outlines in this catalog; schedule, pacing and capstone projects are agreed per engagement.



Machine Learning Bootcamp

From first Python program to deep learning research papers — a complete AI engineering pathway. **7 courses · 26 training days**

#	CODE	COURSE	DOMAIN	DAYS
1	**DCL-160**	Python Programming	Data & ML	4
2	**DCL-162**	Advanced Python Programming	Data & ML	3
3	**DCL-402**	Data Analytics using Python	Data & ML	3
4	**DCL-410**	Practical Machine Learning using Python	Data & ML	4
5	**DCL-426**	Deep Learning with PyTorch	Deep Learning & GenAI	4
6	**DCL-428**	Deep Learning with TensorFlow	Deep Learning & GenAI	4
7	**DCL-430**	Foundation of Deep Learning with Backbone Architecture Papers	Deep Learning & GenAI	4

Full-stack Development Bootcamp

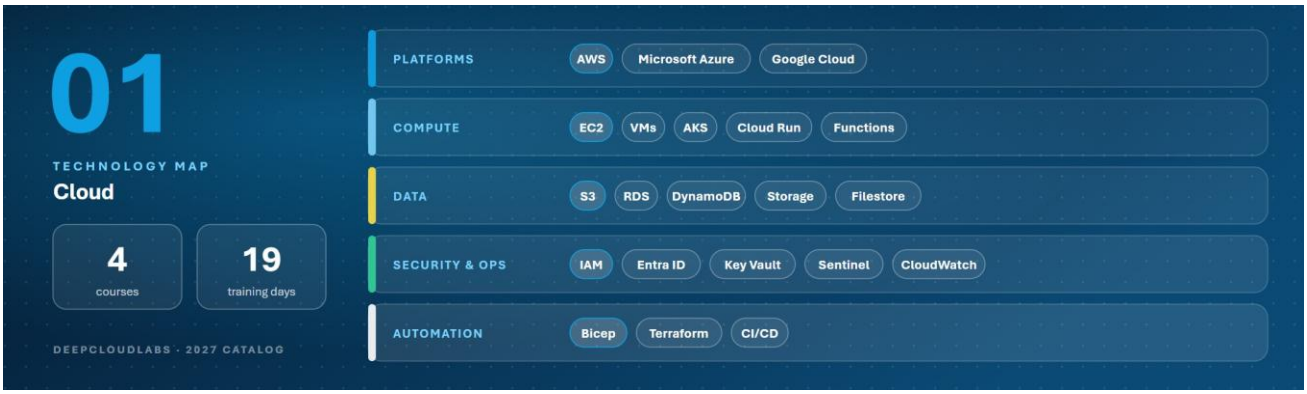
Java SE 27, Spring, React and Spring Cloud — the full enterprise stack, front to back. **4 courses · 17 training days**

#	CODE	COURSE	DOMAIN	DAYS
1	**DCL-204**	Java SE 27 Programming	Java SE 27	5
2	**DCL-370**	Spring Framework 7	Spring & Kotlin	4
3	**DCL-306**	Developing React Applications	JavaScript & Web	3
4	**DCL-350**	Implementing Microservices Architecture using Spring Cloud	Microservices & DevOps	5

TRAINING DOMAIN 01 · 4 COURSES · 19 TRAINING DAYS

01 Cloud Platforms & Architecture

Hands-on fluency on AWS, Microsoft Azure and Google Cloud — from core services to architect-level design.



Cloud is now the default runtime for enterprise software. These programs build practical fluency on the three leading cloud platforms — Amazon Web Services, Microsoft Azure and Google Cloud — from core compute, storage and networking services to identity, resilience, governance and cost-aware architecture. Every course is lab-driven and connects platform services to the architecture patterns taught in our microservices and DevOps curriculum.

WHAT'S NEW IN 2027

- Identity content updated to Microsoft Entra ID (formerly Azure Active Directory).
- Google Cloud serverless content updated to Cloud Run and Cloud Run functions (formerly Cloud Functions).
- Each cloud course now carries a catalog code (DCL-500 series) and a full module outline.

PROGRAM OUTLINE

CODE	COURSE	LEVEL	DAYS	PAGE
DCL-500	AWS Essentials	Introductory	3	16
DCL-502	Microsoft Azure Fundamentals	Introductory	3	17
DCL-504	Azure Cloud for Architects	Advanced	10	18
DCL-506	Google Cloud Platform Essentials	Introductory	3	19
Total	4 courses		**19**	

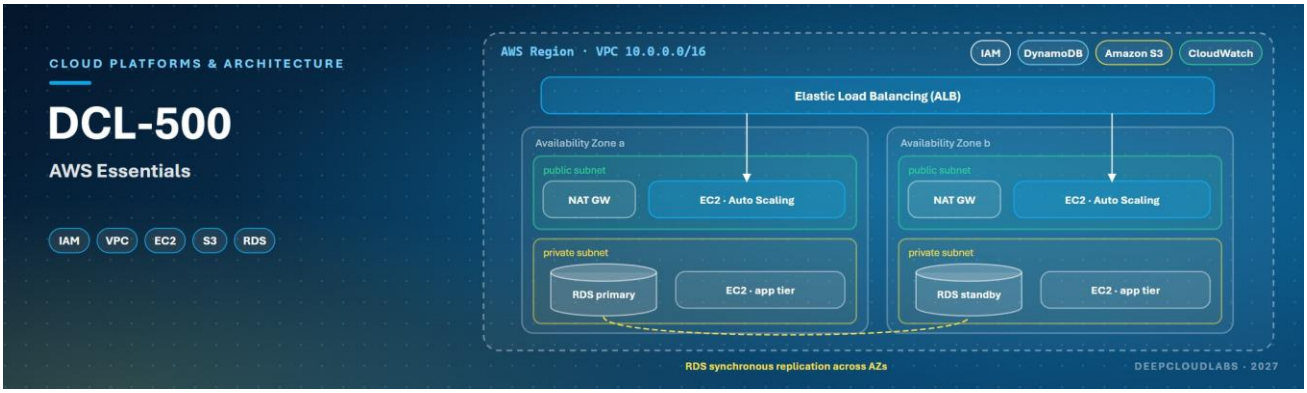
Choosing the right cloud course

COURSE	PLATFORM	LEVEL	DAYS	BEST FOR
DCL-500 AWS Essentials	Amazon Web Services	Introductory	3	Teams deploying their first production workloads on AWS
DCL-502 Azure Fundamentals	Microsoft Azure	Introductory	3	Azure newcomers and AZ-900 exam candidates
DCL-504 Azure for Architects	Microsoft Azure	Advanced	10	Architects designing secure, resilient enterprise solutions
DCL-506 Google Cloud Essentials	Google Cloud	Introductory	3	Multi-cloud teams adding Google Cloud to AWS/Azure skills

DCL-500 CLOUD PLATFORMS & ARCHITECTURE

AWS Essentials

Provision, secure and operate core AWS workloads — hands-on from day one.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Introductory	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A hands-on introduction to provisioning, securing and operating core AWS workloads. Participants build a well-structured AWS environment step by step: identity and access with IAM, network isolation with Amazon VPC, compute with Amazon EC2, object storage with Amazon S3, and managed databases with Amazon RDS and Amazon DynamoDB. Elastic Load Balancing and Auto Scaling deliver elasticity and high availability, while Amazon CloudWatch and governance and security best practices keep workloads observable, secure and cost-aware.

WHAT YOU WILL LEARN

- Navigate the AWS global infrastructure and the shared responsibility model
- Secure access with IAM users, roles and least-privilege policies
- Design a VPC with public and private subnets, routing and security groups
- Deploy elastic, highly available workloads with EC2, ELB and Auto Scaling
- Select S3, RDS or DynamoDB for a workload and monitor it with CloudWatch

WHO SHOULD ATTEND

Cloud engineers and developers new to AWS; system administrators and architects moving workloads to AWS.

PREREQUISITES

Basic familiarity with networking (IP addressing, DNS), operating systems and the command line. No prior AWS experience required.

TECHNOLOGIES

IAM · Amazon VPC · Amazon EC2 · Amazon S3 · Amazon RDS · DynamoDB · ELB · Auto Scaling · CloudWatch

RELATED COURSES

DCL-356 Cloud Architecture Patterns
DCL-353 DevOps Engineering

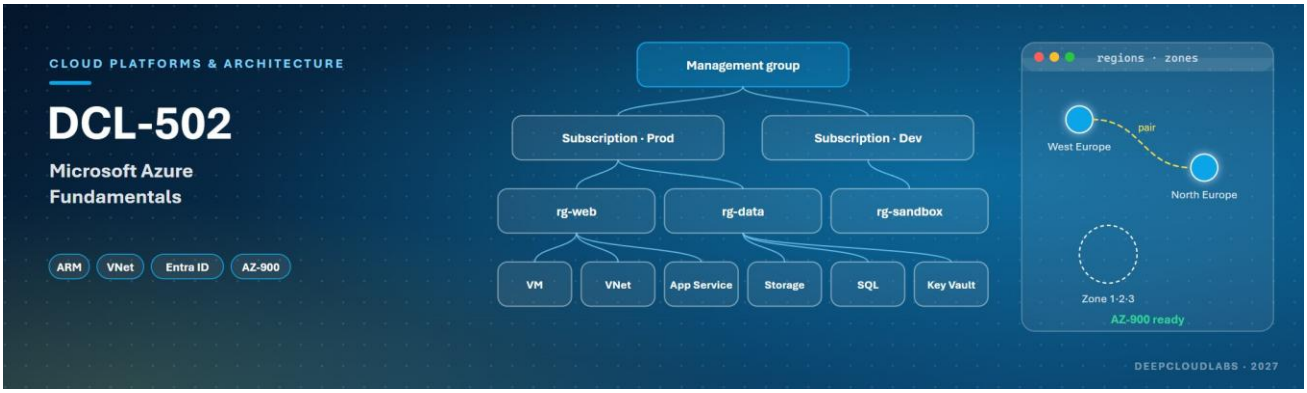
COURSE MODULES

01	Cloud Concepts and AWS Global Infrastructure	06	Databases: Amazon RDS and Amazon DynamoDB
02	Identity and Access Management (IAM)	07	Elasticity and High Availability: ELB and Auto Scaling
03	Networking with Amazon VPC	08	Monitoring with Amazon CloudWatch
04	Compute with Amazon EC2	09	Governance and Security Best Practices
05	Object Storage with Amazon S3	10	Labs: WordPress on AWS, Windows on EC2, CloudWatch Monitoring

DCL-502 CLOUD PLATFORMS & ARCHITECTURE

Microsoft Azure Fundamentals

The fastest path to Azure fluency — and to the AZ-900 credential.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Introductory	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A structured introduction to Microsoft Azure that builds the conceptual and practical foundation every Azure practitioner needs. Participants explore Azure's global architecture — regions, availability zones, resource groups and Azure Resource Manager (ARM) — and work hands-on with virtual machines, virtual networks, storage and identity with Microsoft Entra ID. The course covers business continuity, high availability, scalability and elasticity, and prepares participants for the AZ-900 Microsoft Azure Fundamentals exam across all objective domains.

WHAT YOU WILL LEARN

- Explain cloud concepts, service models (IaaS, PaaS, SaaS) and Azure pricing
- Organize resources with management groups, subscriptions, resource groups and ARM
- Deploy and connect virtual machines, virtual networks and storage accounts
- Apply identity, access and governance controls with Entra ID, RBAC and Azure Policy
- Prepare effectively for the AZ-900 exam

WHO SHOULD ATTEND

IT professionals, developers and technical stakeholders starting with Azure; AZ-900 candidates.

PREREQUISITES

General IT literacy. No prior cloud experience required.

TECHNOLOGIES

Azure Resource Manager · Virtual Machines · Virtual Network · Azure Storage · Microsoft Entra ID · Azure Policy

RELATED COURSES

DCL-504 Azure Cloud for Architects

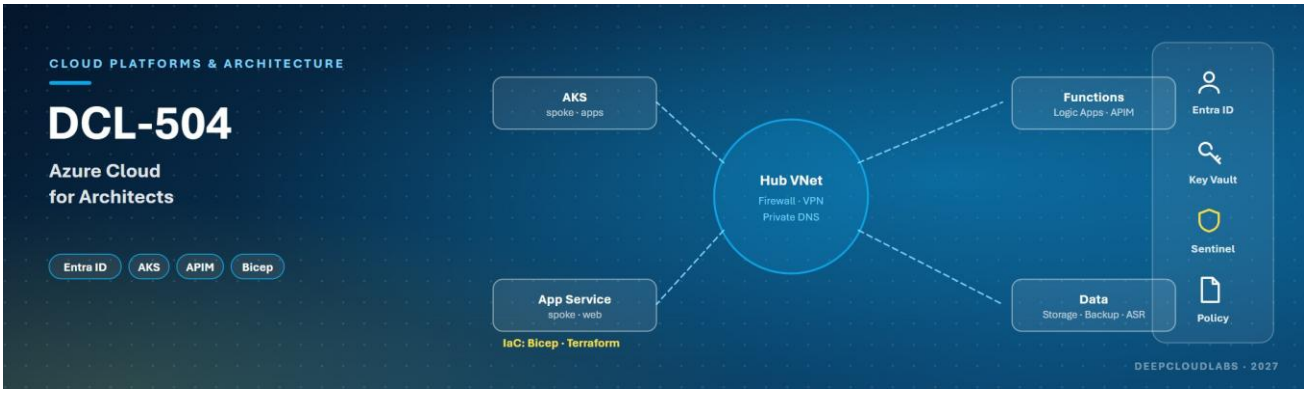
COURSE MODULES

01	Cloud Concepts and Service Models	06	Identity with Microsoft Entra ID
02	Azure Architecture: Regions, Availability Zones, Resource Groups and ARM	07	Business Continuity: High Availability, Scalability and Elasticity
03	Compute: Virtual Machines, App Service and Containers	08	Governance, Compliance and Cost Management
04	Networking: Virtual Networks, Peering and DNS	09	AZ-900 Exam Preparation and Practice
05	Storage Accounts, Access Tiers and Redundancy		

DCL-504 CLOUD PLATFORMS & ARCHITECTURE

Azure Cloud for Architects

An architect-level masterclass for secure, resilient, cost-optimized Azure solutions.



DURATION	LEVEL	DELIVERY	WHERE
10 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

An intensive ten-day program for designing enterprise-grade Azure solutions. Participants design identity and security foundations with Microsoft Entra ID, role-based access control, Conditional Access, Azure Key Vault and Microsoft Sentinel; select and integrate application platforms including Azure Kubernetes Service (AKS), App Service, Azure Functions, Logic Apps and API Management; and architect storage, backup and site-recovery solutions. The program closes with DevOps and infrastructure as code using Bicep and Terraform, and with governance and compliance at scale.

WHAT YOU WILL LEARN

- Design identity, access and security architectures on Azure
- Choose and integrate the right compute and integration platforms
- Architect resilient storage, backup and disaster-recovery solutions
- Automate environments with Bicep, Terraform and CI/CD pipelines
- Govern Azure at scale with policy, compliance and cost controls

WHO SHOULD ATTEND

Solution and cloud architects, senior engineers and experienced IT professionals designing Azure solutions.

PREREQUISITES

Azure fundamentals knowledge (DCL-502 / AZ-900 level) and practical infrastructure or development experience.

TECHNOLOGIES

Microsoft Entra ID · Key Vault · Microsoft Sentinel · AKS · App Service · Azure Functions · Logic Apps · API Management · Bicep · Terraform

RELATED COURSES

DCL-356 Cloud Architecture Patterns
DCL-353 DevOps Engineering

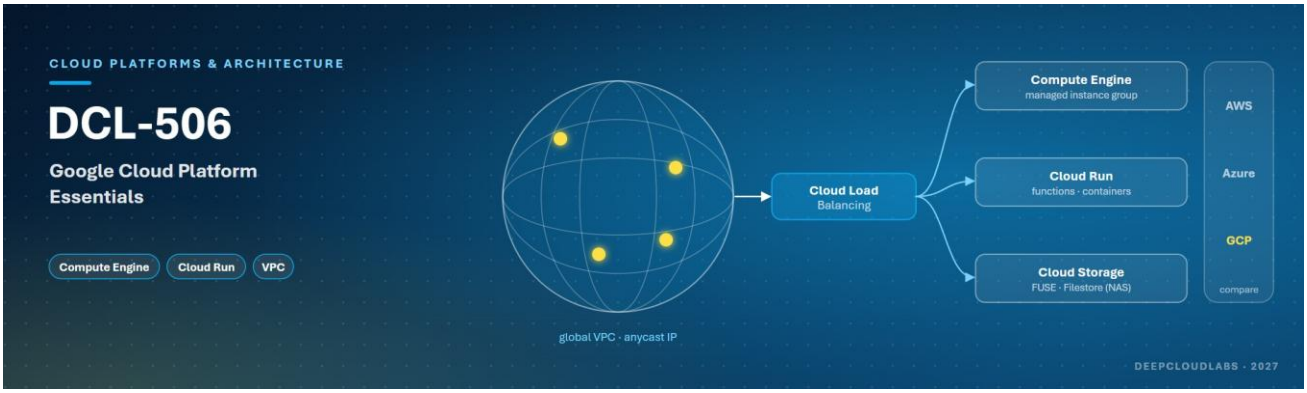
COURSE MODULES

01	Azure Architecture Foundations and Design Principles	07	Application Platforms: App Service and Azure Functions
02	Identity and Access: Microsoft Entra ID, RBAC and Conditional Access	08	Integration: Logic Apps and API Management
03	Secrets, Keys and Certificates with Azure Key Vault	09	Storage, Backup and Site-Recovery Architectures
04	Security Operations with Microsoft Sentinel	10	DevOps and Infrastructure as Code (Bicep, Terraform)
05	Network Architecture: Hub-and-Spoke, Private Endpoints and Hybrid Connectivity	11	Governance, Compliance and Cost Optimization
06	Containers and Kubernetes with AKS	12	Architecture Case Study and Design Review

DCL-506 CLOUD PLATFORMS & ARCHITECTURE

Google Cloud Platform Essentials

Navigate Google Cloud confidently — and compare it intelligently with AWS and Azure.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Introductory	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A practical introduction to Google Cloud for practitioners adding it to their stack. Participants deploy compute with Compute Engine and managed instance groups and run serverless workloads with Cloud Run and Cloud Run functions (formerly Cloud Functions); compare object, FUSE-mounted and NAS storage patterns; and design networks with VPC, peering, Cloud VPN, Cloud Interconnect and load balancing. The course closes with monitoring, logging and SRE release-management practices, with side-by-side comparisons to equivalent AWS and Azure services throughout.

WHAT YOU WILL LEARN

- Organize projects, IAM and billing in the Google Cloud resource hierarchy
- Run workloads on Compute Engine, managed instance groups and Cloud Run
- Select the right storage pattern: Cloud Storage, Cloud Storage FUSE or Filestore
- Design VPC networks with hybrid connectivity and global load balancing
- Map Google Cloud services to their AWS and Azure equivalents

WHO SHOULD ATTEND

Practitioners adding Google Cloud to a multi-cloud skill set; engineers and architects evaluating Google Cloud.

PREREQUISITES

Basic cloud and networking knowledge. Experience with another cloud provider is helpful but not required.

TECHNOLOGIES

Compute Engine · Cloud Run · Cloud Storage · Filestore · VPC · Cloud Load Balancing · Cloud Monitoring

RELATED COURSES

DCL-356 Cloud Architecture Patterns

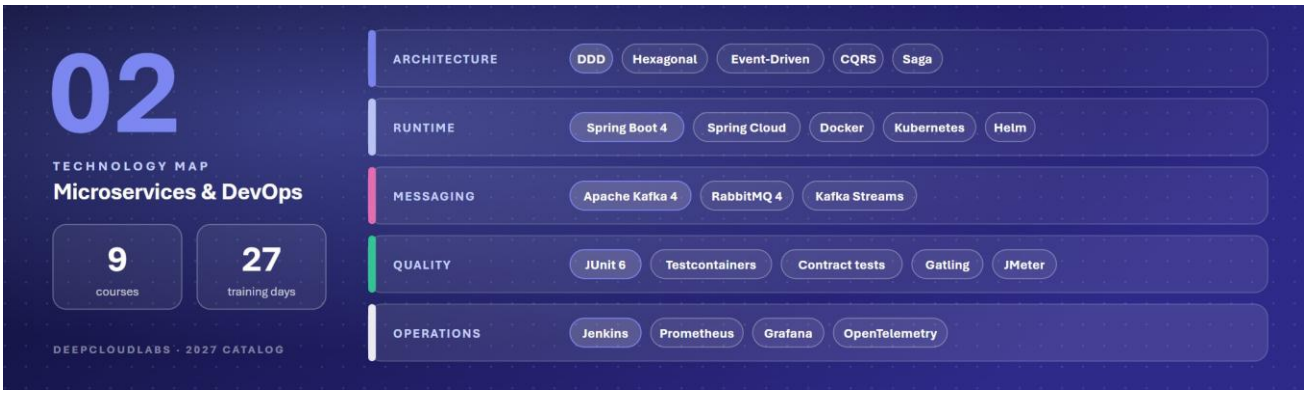
COURSE MODULES

01	Resource Hierarchy, IAM and Billing	06	Cloud Load Balancing
02	Compute Engine and Managed Instance Groups	07	Monitoring and Logging
03	Serverless with Cloud Run and Cloud Run functions	08	SRE Practices and Release Management
04	Storage Patterns: Cloud Storage, Cloud Storage FUSE and Filestore (NAS)	09	Google Cloud vs AWS vs Azure: Service Mapping
05	VPC Networking, Peering, Cloud VPN and Cloud Interconnect		

TRAINING DOMAIN 02 · 9 COURSES · 27 TRAINING DAYS

02 Microservices, DevOps & Distributed Systems

Design, build, test and deliver distributed systems that are resilient, observable and evolvable.



Modern enterprise platforms are distributed by design. This curriculum covers the full lifecycle of microservices-based systems: domain modeling with Domain-Driven Design, microservices and cloud architecture patterns, implementation with Spring Boot and Spring Cloud, event-driven integration with Apache Kafka and RabbitMQ, testing strategies, and automated CI/CD delivery to Kubernetes.

WHAT'S NEW IN 2027

- Apache Kafka content updated to Kafka 4 (KRaft-only, ZooKeeper removed).
- RabbitMQ content updated to RabbitMQ 4 (quorum queues and streams replace classic mirrored queues).
- Spring Cloud content aligned with Spring Boot 4; tracing updated to Micrometer Tracing and OpenTelemetry.
- DevOps Engineering extended with supply-chain security (image scanning, SBOMs) and GitOps delivery patterns.

PROGRAM OUTLINE

CODE	COURSE	LEVEL	DAYS	PAGE
DCL-350	Implementing Microservices Architecture using Spring Cloud	Intermediate	5	22
DCL-351	Microservices Testing	Intermediate–Advanced	4	23
DCL-352	Domain-Driven Design Essentials	Intermediate	2	24
DCL-353	DevOps Engineering	Intermediate	3	25
DCL-355	Microservices Patterns with Examples in Spring Boot	Advanced	4	26
DCL-356	Cloud Architecture Patterns	Intermediate	2	27
DCL-358	Implementing Event-Driven Microservices Architecture using Spring Boot and Apache Kafka	Advanced	3	28
DCL-380	Apache Kafka: Architecture and Development	Intermediate	2	29
DCL-382	RabbitMQ: Architecture and Administration	Intermediate	2	30
Total	9 courses		**27**	

Curriculum map: the distributed-systems lifecycle

Use this map to assemble a program around the stage your teams are in — or combine stages into a complete microservices learning path.

STAGE	FOCUS	COURSES
Model	Find service boundaries and a shared language with the business	DCL-352 Domain-Driven Design Essentials
Design	Choose patterns for decomposition, data consistency, communication and cloud scale	DCL-355 Microservices Patterns · DCL-356 Cloud Architecture Patterns
Build	Implement services, configuration, discovery, routing and resilience	DCL-350 Microservices with Spring Cloud
Integrate	Connect services through events, streams and queues	DCL-358 Event-Driven Microservices · DCL-380 Apache Kafka · DCL-382 RabbitMQ
Verify	Test at the right levels — unit, integration, contract, end-to-end, performance	DCL-351 Microservices Testing
Deliver & operate	Automate CI/CD to Kubernetes; secure and observe the pipeline	DCL-353 DevOps Engineering

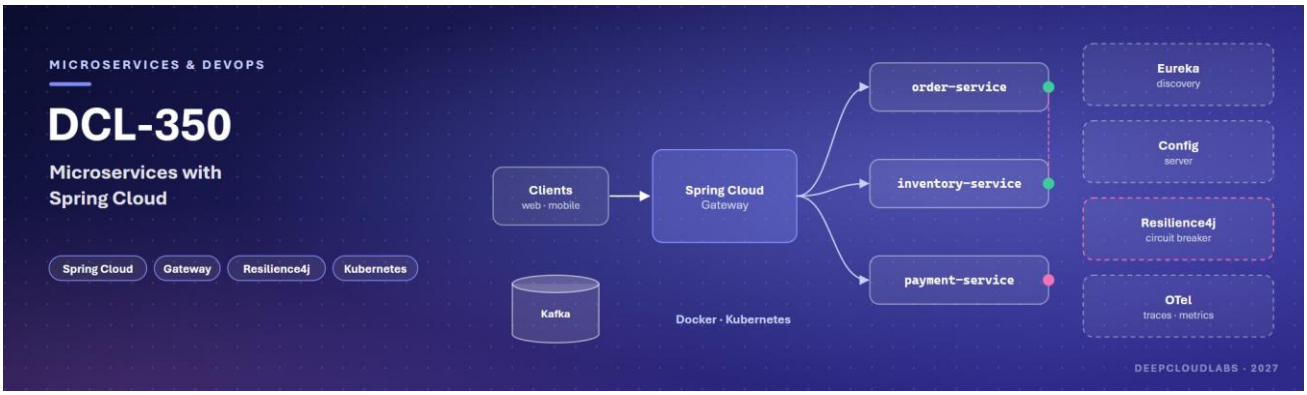
Kafka or RabbitMQ? Choosing a messaging course

ASPECT	DCL-380 APACHE KAFKA	DCL-382 RABBITMQ
Core model	Distributed, partitioned, replicated commit log	Message broker with exchanges, bindings and queues
Typical use	Event streaming, event sourcing, data pipelines	Work queues, flexible routing, request/reply
2027 version focus	Kafka 4: KRaft-only metadata, new consumer rebalance protocol	RabbitMQ 4: quorum queues and streams for replication
Course emphasis	Producers/consumers, CLI, Connect, Kafka Streams	Administration, Spring AMQP & Node.js, clustering, HA, tuning

DCL-350 MICROSERVICES, DEVOPS & DISTRIBUTED SYSTEMS

Implementing Microservices Architecture using Spring Cloud

Build, secure, observe and deploy containerized microservices with Spring Boot 4 and Spring Cloud.



DURATION	LEVEL	DELIVERY	WHERE
5 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This course gives developers and architects the tools and techniques to build, manage and deploy containerized microservices with Spring Framework, Spring Boot and Spring Cloud. Beyond implementation, it focuses on the key considerations for a well-planned microservices architecture: service boundaries, evolutionary architecture, centralized configuration, service discovery, client resiliency, routing, data and messaging integration, and distributed observability — finishing with deployment on Docker and Kubernetes.

WHAT YOU WILL LEARN

- Decompose a domain into well-bounded, independently deployable services
- Externalize configuration and implement service discovery and routing
- Apply client resiliency patterns with Resilience4j
- Integrate services through REST, WebSocket, data and messaging
- Trace, monitor and deploy microservices on Docker and Kubernetes

WHO SHOULD ATTEND

Java developers, software architects and technical leads building distributed systems.

PREREQUISITES

Solid Java and Spring Boot experience (DCL-374 or equivalent).

TECHNOLOGIES

Spring Boot 4 · Spring Cloud · Eureka · Resilience4j · Spring Cloud Gateway · Micrometer · OpenTelemetry · Docker · Kubernetes

RELATED COURSES

DCL-355 Microservices Patterns with Examples in Spring Boot

DCL-358 Implementing Event-Driven Microservices Architecture using Spring Boot and Apache Kafka

DCL-351 Microservices Testing

COURSE MODULES

01	Introduction to Microservices Architecture	09	Spring Cloud Config
02	The Evolutionary Architecture	10	Service Discovery with Spring Cloud Netflix Eureka
03	Modeling Services	11	Client Resiliency Patterns with Resilience4j
04	Spring Boot Bootcamp	12	Service Routing with Spring Cloud Gateway
05	Integrating Services with Spring MVC and HTTP Interface Clients	13	Data Integration with Spring Data
06	Integrating Services with Spring WebSocket	14	Data Integration with Spring Messaging
07	Spring Cloud and Microservices	15	Distributed Logging and Tracing (Micrometer, OpenTelemetry)
08	Spring Boot Actuator	16	Microservices Deployment with Docker and Kubernetes

DCL-351 MICROSERVICES, DEVOPS & DISTRIBUTED SYSTEMS

Microservices Testing

Test strategies that keep distributed systems reliable, scalable and release-ready.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Intermediate–Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This four-day course gives developers, QA engineers and SREs a practical understanding of testing strategies for microservices-based systems. Participants examine the architectural differences between monolithic, SOA and microservices approaches, together with REST communication, service registries and API gateways such as Eureka and MuleSoft. The course covers unit, integration, contract, end-to-end and UI testing, performance testing with JMeter and Gatling, and observability-driven validation with Grafana on Azure Kubernetes Service (AKS) — enabling participants to assess reliability, scalability and operational readiness.

WHAT YOU WILL LEARN

- Choose the right test types and balance for a microservices test pyramid
- Write reliable integration tests with real dependencies using Testcontainers
- Prevent breaking API changes with consumer-driven contract tests
- Design and run performance tests with JMeter and Gatling
- Validate production readiness through observability with Grafana

WHO SHOULD ATTEND

Developers, QA engineers and SREs working on service-based systems.

PREREQUISITES

Experience developing or testing service-based applications; Java familiarity recommended.

TECHNOLOGIES

JUnit 6 · Testcontainers · Contract testing · JMeter · Gatling · Grafana · AKS · Eureka · MuleSoft

RELATED COURSES

DCL-353 DevOps Engineering
DCL-215 Test-Driven Development with JUnit 6
DCL-606 AI-Assisted Software Testing and Quality Engineering

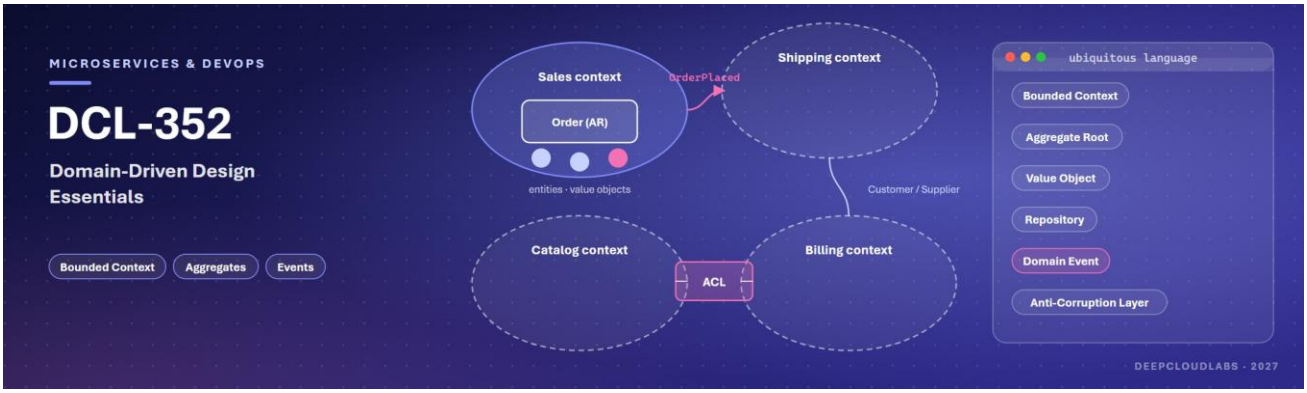
COURSE MODULES

01	Monolith vs SOA vs Microservice Decomposition	05	End-to-End and UI Testing
02	REST, Service Registries and API Gateways (Eureka, MuleSoft)	06	Performance Testing with JMeter and Gatling
03	Unit and Integration Testing with Testcontainers	07	Observability-Driven Validation with Grafana on AKS
04	Consumer-Driven Contract Testing		

DCL-352 MICROSERVICES, DEVOPS & DISTRIBUTED SYSTEMS

Domain-Driven Design Essentials

Model complex business domains so software and the business speak the same language.



DURATION	LEVEL	DELIVERY	WHERE
2 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This two-day course introduces the fundamental principles and practices of Domain-Driven Design (DDD) for developing complex, business-oriented software systems. Participants learn to model business problems using domain models, entities, value objects, services and bounded contexts, and examine aggregates, repositories, domain events and anti-corruption layers as core mechanisms for preserving consistency and isolating domain logic. Participants leave able to structure scalable architectures, collaborate effectively with domain experts and evolve domain models as requirements change.

WHAT YOU WILL LEARN

- Build a ubiquitous language with domain experts
- Identify bounded contexts and map their relationships
- Design aggregates, entities and value objects that protect invariants
- Use repositories, domain events and anti-corruption layers effectively
- Apply DDD to structure microservice boundaries

WHO SHOULD ATTEND

Software developers, architects, technical leads and business analysts working on complex domains.

PREREQUISITES

Software development experience in an object-oriented language.

TECHNOLOGIES

Bounded contexts · Context mapping · Aggregates · Domain events · Hexagonal architecture

RELATED COURSES

DCL-355 Microservices Patterns with Examples in Spring Boot

DCL-358 Implementing Event-Driven Microservices Architecture using Spring Boot and Apache Kafka

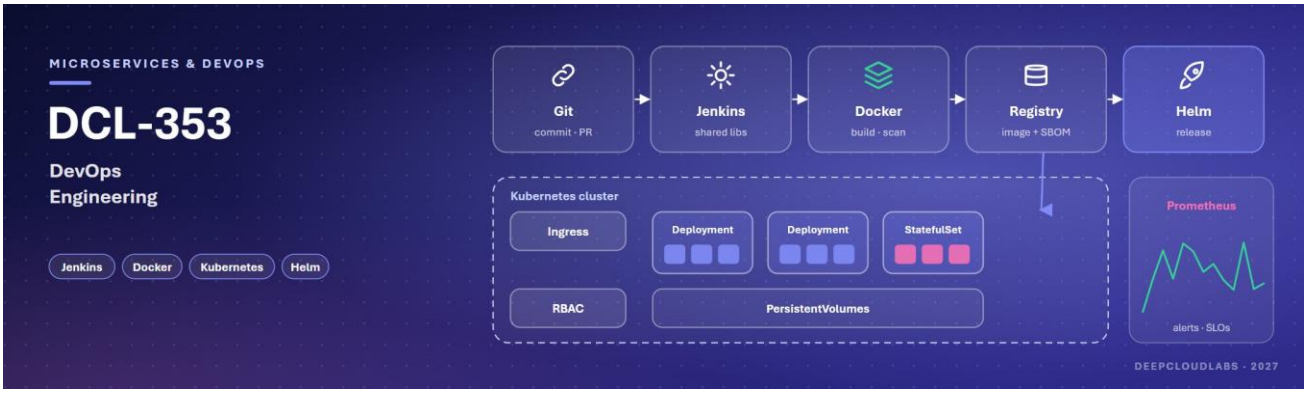
COURSE MODULES

01	Introduction to DDD	05	Repositories
02	DDD: Modeling Problems in Software	06	Domain Events and Anti-Corruption Layers
03	Elements of a Domain Model	07	Extending Domain-Driven Design
04	Aggregates in Domain-Driven Design		

DCL-353 MICROSERVICES, DEVOPS & DISTRIBUTED SYSTEMS

DevOps Engineering

Build a complete CI/CD pipeline — from Git commit to production Kubernetes.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This three-day course gives developers, operations engineers and release engineers the practical knowledge to build and manage an end-to-end CI/CD pipeline from source-code commit to production deployment on Kubernetes. Participants explore Git and Bitbucket workflows and branching strategies; Docker image creation, Compose environments, persistent volumes and container debugging; Jenkins pipelines with shared libraries; and Kubernetes deployments, ingress controllers, StatefulSets and Helm-based release management. The course closes with secure, observable delivery: role-based access control, software supply-chain security and production-grade monitoring with Prometheus.

WHAT YOU WILL LEARN

- Apply Git branching strategies that support continuous delivery
- Build, debug and optimize container images with Docker
- Automate pipelines with Jenkins and shared libraries
- Deploy and release to Kubernetes with Helm
- Secure and observe the delivery pipeline with RBAC, SBOMs and Prometheus

WHO SHOULD ATTEND

Developers, operations engineers and release engineers.

PREREQUISITES

Basic Linux command-line skills and familiarity with software build processes.

TECHNOLOGIES

Git · Bitbucket · Docker · Jenkins · Kubernetes · Helm · Prometheus · RBAC

RELATED COURSES

DCL-351 Microservices Testing
DCL-504 Azure Cloud for Architects
DCL-604 Agentic Software Engineering

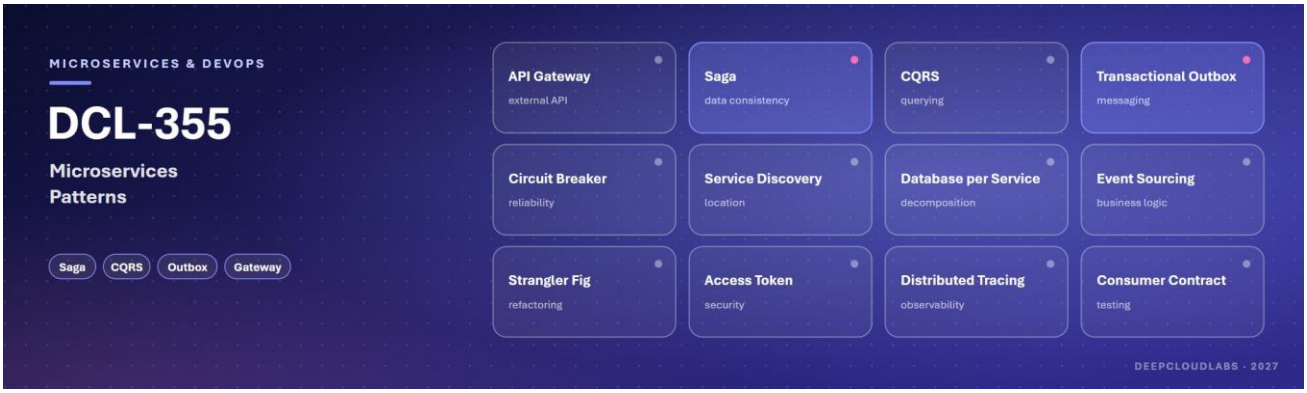
COURSE MODULES

01	Git, Bitbucket and DevOps Branching Strategies	06	Release Management with Helm
02	Docker: Images, Compose, Volumes and Debugging	07	Supply-Chain Security: Image Scanning, SBOMs and RBAC
03	CI Pipelines with Jenkins and Shared Libraries	08	Observability with Prometheus
04	Kubernetes Deployments, Services and Ingress	09	GitOps and Progressive Delivery Overview
05	Stateful Workloads with StatefulSets		

DCL-355 MICROSERVICES, DEVOPS & DISTRIBUTED SYSTEMS

Microservices Patterns with Examples in Spring Boot

The pattern language of microservices — implemented, compared and refactored in Spring Boot.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A comprehensive, practice-oriented examination of microservices architecture patterns using Spring Boot examples. Participants explore service decomposition, messaging, reliable communication, service discovery, transactional messaging, data consistency, business logic, querying, external APIs, testing and security patterns, and address cross-cutting concerns, observability, deployment strategies and the architectural trade-offs of resilient, scalable distributed systems. Participants leave able to select, implement and evaluate the right patterns and to systematically refactor monoliths into maintainable Spring Boot microservices.

WHAT YOU WILL LEARN

- Select decomposition, communication and data patterns for a given context
- Maintain consistency across services with sagas and transactional messaging
- Implement CQRS, API gateway and security patterns in Spring Boot
- Design for observability and safe deployment
- Refactor a monolith incrementally with the Strangler Fig approach

WHO SHOULD ATTEND

Software architects, senior developers and technical leads.

PREREQUISITES

Spring Boot experience and familiarity with microservices concepts (DCL-350 recommended).

TECHNOLOGIES

Spring Boot 4 · Saga · CQRS · Transactional Outbox · API Gateway · Circuit Breaker

RELATED COURSES

DCL-358 Implementing Event-Driven Microservices Architecture using Spring Boot and Apache Kafka

DCL-356 Cloud Architecture Patterns

COURSE MODULES

01	Microservices Architecture Basics	10	Querying Patterns
02	Application Architecture Patterns	11	External API Patterns
03	Decomposition Patterns	12	Testing Patterns
04	Messaging Style Patterns	13	Security Patterns
05	Reliable Communications Patterns	14	Cross-Cutting Concerns Patterns
06	Service Discovery Patterns	15	Observability Patterns
07	Transactional Messaging Patterns	16	Deployment Patterns
08	Data Consistency Patterns	17	Refactoring to Microservices Patterns
09	Business Logic Design Patterns		

DCL-356 MICROSERVICES, DEVOPS & DISTRIBUTED SYSTEMS

Cloud Architecture Patterns

Proven patterns for scalable, resilient and high-performance cloud systems.



DURATION	LEVEL	DELIVERY	WHERE
2 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This two-day course introduces essential cloud architecture patterns for designing scalable, resilient and high-performance distributed systems. Participants explore horizontal scaling, queue-centric workflows, auto-scaling, eventual consistency, MapReduce, database sharding, multi-tenancy and commodity-hardware strategies, and examine failure and latency management through the busy signal, node failure, colocation, valet key, CDN and multi-site deployment patterns. Participants leave able to evaluate architectural trade-offs and apply the right patterns to improve availability, scalability, security and operational efficiency.

WHAT YOU WILL LEARN

- Reason about scalability, eventual consistency and network latency
- Apply compute, data and workflow patterns to cloud workloads
- Design for node failure, throttling and multi-site resilience
- Secure direct resource access with the valet key pattern
- Evaluate trade-offs across cloud providers

WHO SHOULD ATTEND

Software and cloud architects, senior developers and technical leads.

PREREQUISITES

Experience with distributed or web-based applications; basic cloud concepts.

TECHNOLOGIES

Auto-scaling · Queues · Sharding · CDN · Multi-region

RELATED COURSES

DCL-504 Azure Cloud for Architects

DCL-355 Microservices Patterns with Examples in Spring Boot

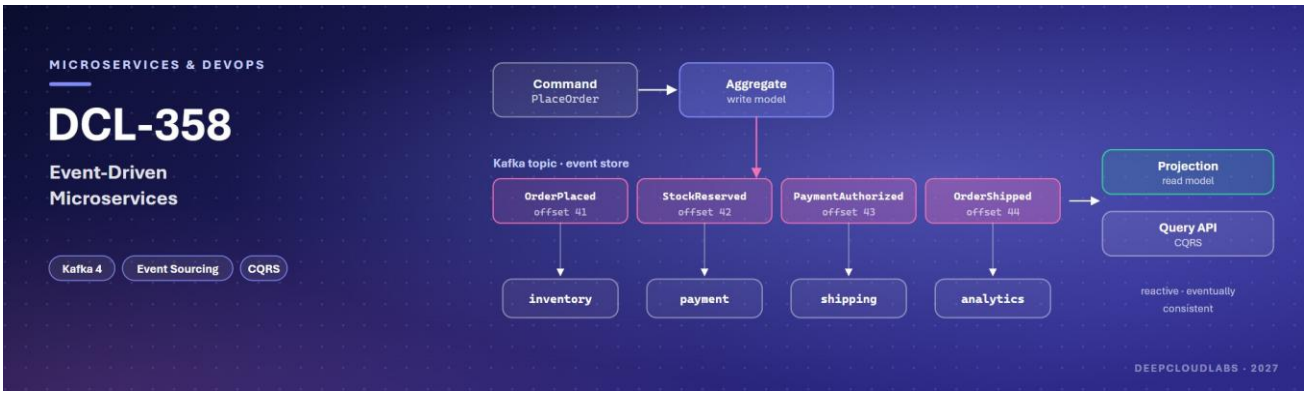
COURSE MODULES

01	Cloud Design Patterns	09	Multi-Tenancy and Commodity Hardware Pattern
02	Scalability Primer	10	Busy Signal Pattern
03	Horizontally Scaling Compute Pattern	11	Node Failure Pattern
04	Queue-Centric Workflow Pattern	12	Network Latency Primer
05	Auto-Scaling Pattern	13	Colocate Pattern
06	Eventual Consistency Primer	14	Valet Key Pattern
07	MapReduce Pattern	15	CDN Pattern
08	Database Sharding Pattern	16	Multi-Site Deployment Pattern

DCL-358 MICROSERVICES, DEVOPS & DISTRIBUTED SYSTEMS

Implementing Event-Driven Microservices Architecture using Spring Boot and Apache Kafka

Loosely coupled, reactive microservices that communicate through reliable event streams.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A practical course on designing and implementing event-driven microservices with Spring Boot and Apache Kafka. Participants explore event-driven architecture patterns, domain modeling with event sourcing, commands and events, CQRS and the core concepts of Kafka brokers and messaging, and address reactive programming, consistency, concurrency and transaction management in distributed event-driven systems. Participants leave able to design and implement scalable, resilient and loosely coupled microservices that communicate through Kafka-based event streams.

WHAT YOU WILL LEARN

- Model domains with events, commands and event sourcing
- Separate write and read models with CQRS
- Produce and consume events reliably with Kafka and Spring Boot
- Handle consistency, concurrency and transactions across services
- Build reactive, event-driven services end to end

WHO SHOULD ATTEND

Java developers and architects building event-driven systems.

PREREQUISITES

Spring Boot experience; microservices fundamentals (DCL-350 recommended).

TECHNOLOGIES

Spring Boot 4 · Apache Kafka 4 · Event Sourcing · CQRS · Project Reactor

RELATED COURSES

DCL-380 Apache Kafka: Architecture and Development

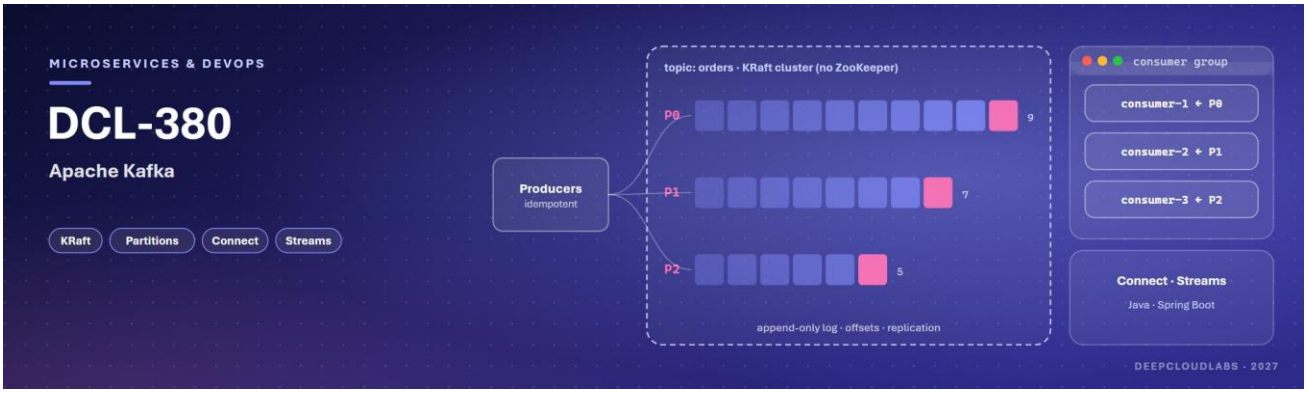
COURSE MODULES

01	Software Architecture and Microservices	06	Event Sourcing and CQRS
02	Event Patterns and Event-Driven Architecture	07	Reactive Event-Driven Systems and Reactive Programming
03	Designing the Domain Model using Event Sourcing	08	Consistency, Concurrency and Transactions in Event-Driven Systems
04	Overview of Apache Kafka 4 and Kafka Brokers (KRaft)	09	Implementing Event-Driven Microservices with Spring Boot and Apache Kafka
05	Events and Commands		

DCL-380 MICROSERVICES, DEVOPS & DISTRIBUTED SYSTEMS

Apache Kafka: Architecture and Development

Kafka 4 from the inside out — architecture, producers, consumers, Connect and Streams.



DURATION	LEVEL	DELIVERY	WHERE
2 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This course introduces Apache Kafka and provides a detailed tour of its architecture so you can build solutions on Kafka using Java and Spring Boot. It is aligned with Apache Kafka 4, which runs exclusively in KRaft mode without ZooKeeper and makes the next-generation consumer rebalance protocol generally available. Participants develop producers and consumers, operate Kafka with the command-line tools, build data pipelines with Kafka Connect and implement stream processing with Kafka Streams.

WHAT YOU WILL LEARN

- Explain Kafka 4 architecture: brokers, KRaft controllers, topics, partitions and replication
- Develop reliable producers and consumers in Java and Spring Boot
- Choose commit strategies and tune producer and consumer configuration
- Build data pipelines with Kafka Connect
- Implement stream processing with Kafka Streams

WHO SHOULD ATTEND

Java developers, data engineers and architects adopting Apache Kafka.

PREREQUISITES

Java programming experience; Spring Boot familiarity is helpful.

TECHNOLOGIES

Apache Kafka 4 · KRaft · Kafka Connect · Kafka Streams · Spring for Apache Kafka

RELATED COURSES

DCL-358 Implementing Event-Driven Microservices Architecture using Spring Boot and Apache Kafka

COURSE MODULES

01 Introduction to Apache Kafka
Kafka 4 architecture (KRaft, no ZooKeeper) · Core concepts and features · Components and installation

02 Developing Kafka Producers
Synchronous and asynchronous sends in Java and Spring Boot · Producer configuration: acks, idempotence, batching

03 Developing Kafka Consumers
Consumers and topic subscription in Java and Spring Boot · Consumer configuration and rebalance protocol · Commit strategies

04 Kafka CLI
Topic CLI · Console producer/consumer CLI · Consumer group CLI

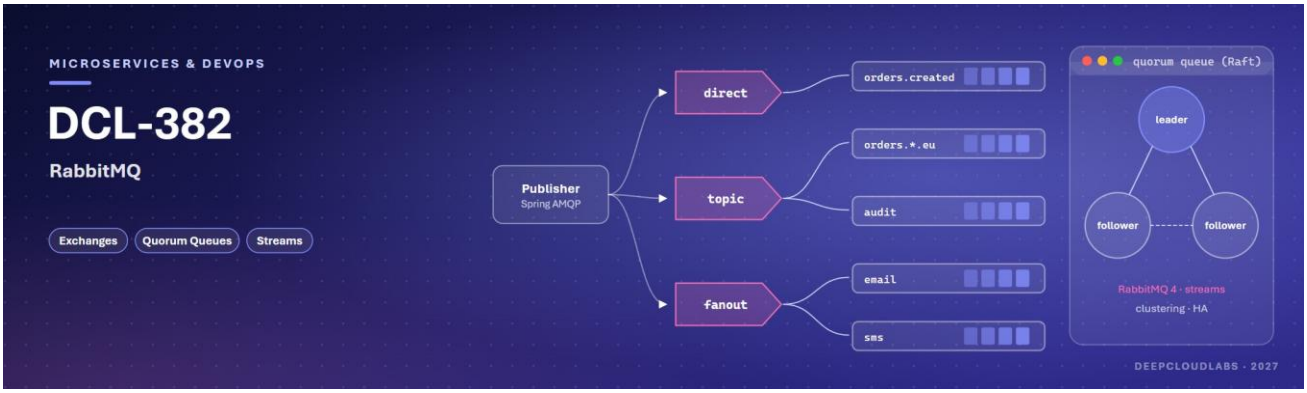
05 Kafka Connect
Architecture and use cases · Building data pipelines

06 Kafka Stream Processing
Kafka Streams architecture and design patterns · Kafka Streams API · Kafka Streams with Spring Boot

DCL-382 MICROSERVICES, DEVOPS & DISTRIBUTED SYSTEMS

RabbitMQ: Architecture and Administration

Install, configure, develop for and operate RabbitMQ 4 — from exchanges to quorum queues.



DURATION	LEVEL	DELIVERY	WHERE
2 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A deep dive into installing, configuring and developing applications that leverage RabbitMQ messaging. The course begins with installation and general configuration, continues with developing messaging applications using Spring AMQP and Node.js, and delves into advanced topics including clustering, high availability and performance tuning. Content is aligned with RabbitMQ 4, where quorum queues and streams provide replication and data safety and classic mirrored queues have been removed.

WHAT YOU WILL LEARN

- Apply enterprise messaging patterns with exchanges, bindings and queues
- Install, configure and administer RabbitMQ
- Develop producers and consumers with Spring AMQP and Node.js
- Build clustered, highly available deployments with quorum queues and streams
- Tune performance and troubleshoot production issues

WHO SHOULD ATTEND

Developers, system administrators and architects running RabbitMQ.

PREREQUISITES

Basic messaging concepts; Java or JavaScript programming experience.

TECHNOLOGIES

RabbitMQ 4 · AMQP · Quorum queues · Streams · Spring AMQP · Node.js

RELATED COURSES

DCL-380 Apache Kafka: Architecture and Development

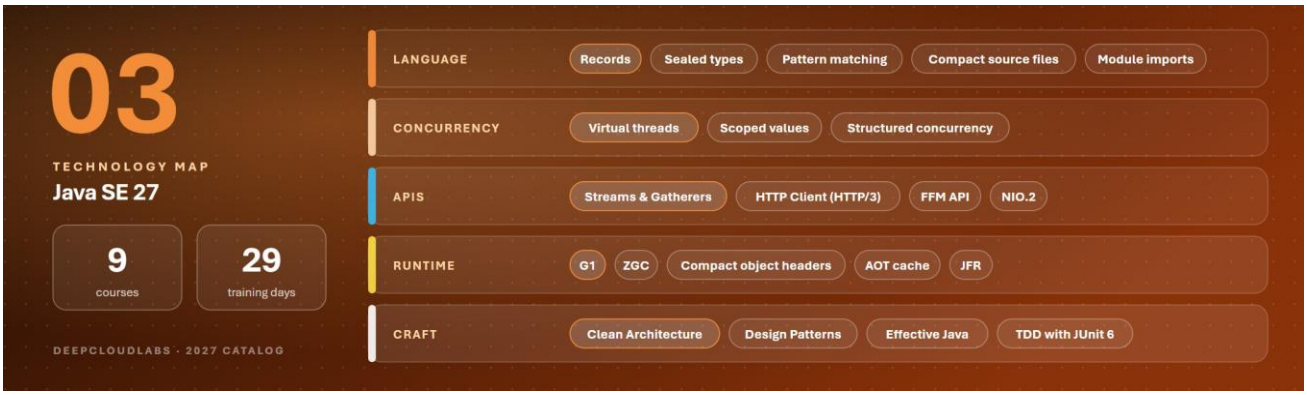
COURSE MODULES

01	Enterprise Messaging and RabbitMQ	05	Clustering
02	Messaging Patterns in RabbitMQ	06	High Availability with Quorum Queues and Streams
03	Administration and Configuration	07	Performance Tuning and Troubleshooting
04	Developing Messaging Applications using Spring AMQP and Node.js		

TRAINING DOMAIN 03 · 9 COURSES · 29 TRAINING DAYS

03 Java SE 27

The complete Java curriculum, upgraded to Java SE 27 — the language, the runtime and the craft.



Java Platform, Standard Edition lets you develop and deploy Java applications on desktops, servers and in the cloud, with the performance, versatility, portability and security today's applications require. Our Java curriculum is aligned with Java SE 27 (general availability 15 September 2026), built on Java SE 25, the current long-term-support release. It covers data-oriented programming with records, sealed types and pattern matching; scalable concurrency with virtual threads, scoped values and structured concurrency; and the latest runtime advances — compact object headers and G1 as the default collector in every environment. Courses are developed in collaboration with our engineering team, which uses Java in production projects.

WHAT'S NEW IN 2027

- All Java SE titles, modules and examples upgraded from Java SE 26 to Java SE 27.
- New Java SE 27 runtime content: G1 default everywhere (JEP 523), compact object headers by default (JEP 534), JFR in-process data redaction (JEP 536).
- New security content: post-quantum hybrid key exchange for TLS 1.3 (JEP 527), ML-KEM/ML-DSA, KDF API.
- Preview features clearly identified: structured concurrency, lazy constants, primitive types in patterns, PEM encodings.
- TDD course upgraded from JUnit 5 to JUnit 6.

PROGRAM OUTLINE

CODE	COURSE	LEVEL	DAYS	PAGE
DCL-200	Java Performance Tuning and Optimization	Advanced	3	33
DCL-204	Java SE 27 Programming	Introductory	5	34
DCL-206	Advanced Java Development with Modern Design and Programming Paradigms	Advanced	4	35
DCL-208	Clean Architecture and Code (Java SE and Spring Boot)	Intermediate	3	36
DCL-210	Advanced Java Programming	Advanced	3	37
DCL-215	Test-Driven Development with JUnit 6	Intermediate	3	38
DCL-220	Object-Oriented Programming Principles and Design Patterns with Java SE 27	Intermediate	3	39
DCL-235	Effective Java Programming	Advanced	3	40
DCL-252	New Features in Java SE 8–27	Intermediate	2	41
Total	9 courses		**29**	

Java SE 27 at a glance

Java SE 27 reached general availability on 15 September 2026 with nine JEPs. It is a feature release; Java SE 25 (September 2025) remains the most recent long-term-support release. Our courses teach final features as production-ready and clearly label previews and incubators.

JEP	FEATURE	STATUS IN 27	COVERED IN
JEP 523	Make G1 the Default Garbage Collector in All Environments	Final	DCL-200, DCL-252
JEP 527	Post-Quantum Hybrid Key Exchange for TLS 1.3	Final	DCL-210
JEP 534	Compact Object Headers by Default	Final	DCL-200, DCL-252
JEP 536	JFR In-Process Data Redaction	Final	DCL-200
JEP 531	Lazy Constants	Third preview	DCL-252
JEP 532	Primitive Types in Patterns, instanceof, and switch	Fifth preview	DCL-204, DCL-252
JEP 533	Structured Concurrency	Seventh preview	DCL-206, DCL-252
JEP 537	Vector API	Twelfth incubator	DCL-200
JEP 538	PEM Encodings of Cryptographic Objects	Third preview	DCL-210

Modern Java feature map

FEATURE	SINCE	STATUS	WHY IT MATTERS
Records · sealed classes	16 · 17	Final	Data-oriented domain models
Pattern matching for switch · record patterns	21	Final	Expressive, exhaustive branching
Virtual threads	21	Final	Millions of concurrent tasks; no pinning since 24
Sequenced collections	21	Final	Uniform first/last access
Foreign Function & Memory API	22	Final	Safe native interop without JNI
Stream gatherers · Class-File API	24	Final	Custom stream operations; bytecode tooling
Scoped values	25	Final	Context propagation for virtual threads
Compact source files · instance main · module imports	25	Final	Faster start for scripts and learners
Flexible constructor bodies	25	Final	Validation before super()
HTTP/3 for the HTTP Client	26	Final	Modern, low-latency networking
Compact object headers by default	27	Final	Lower heap footprint out of the box

DCL-200 JAVA SE 27

Java Performance Tuning and Optimization

Measure, profile and tune the JVM — from garbage collection to JIT, AOT and object layout.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A practitioner's course in Java performance engineering on Java SE 27. Participants learn the basic principles of performance, monitor operating-system performance on Linux and Windows, and observe the JVM and JIT compiler at work. Using VisualVM, JDK Flight Recorder (JFR) and JDK Mission Control, they profile real applications, then compare and tune the G1, ZGC and Shenandoah collectors. The course covers Java SE 27 runtime advances — G1 as the default collector in all environments, compact object headers enabled by default, ahead-of-time caching and JFR data redaction — and closes with benchmarking and with maximizing startup and footprint using GraalVM and Quarkus.

WHAT YOU WILL LEARN

- Apply a repeatable performance methodology across OS, JVM and application layers
- Profile CPU, allocation, memory and lock contention with JFR and JDK Mission Control
- Select and tune G1, ZGC and Shenandoah for throughput and latency targets
- Reduce footprint and startup with compact object headers, AOT caching and GraalVM
- Build trustworthy benchmarks and avoid common measurement pitfalls

WHO SHOULD ATTEND

Senior Java developers, performance engineers, SREs and architects responsible for production Java systems.

PREREQUISITES

Strong Java programming experience (DCL-204 or equivalent) and basic Linux/Windows operations knowledge.

TECHNOLOGIES

JDK 27 · JFR · JDK Mission Control · VisualVM · G1 · ZGC · Shenandoah · JMH · GraalVM · Quarkus

RELATED COURSES

DCL-210 Advanced Java Programming

DCL-206 Advanced Java Development with Modern Design and Programming Paradigms

COURSE MODULES

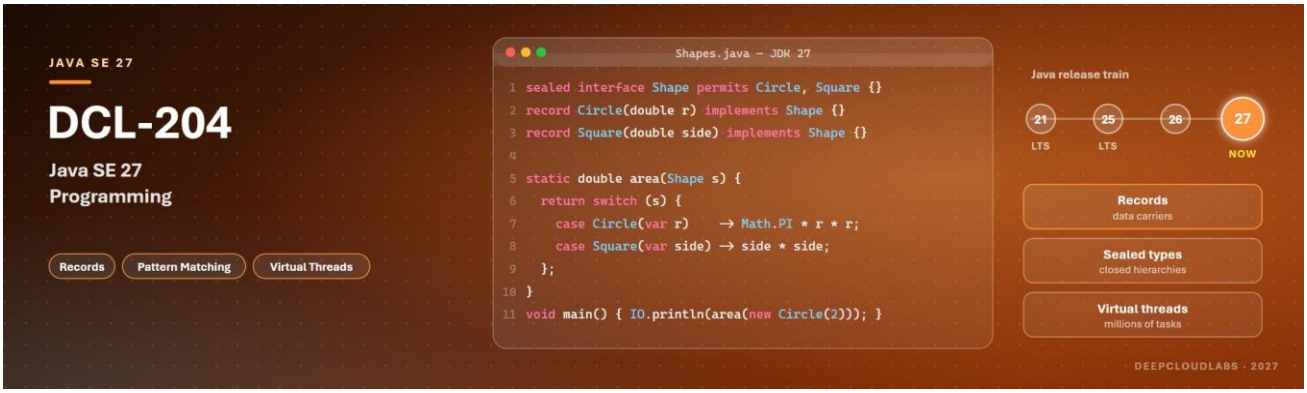
01	JVM Overview and Performance Methodology	07	Language and GC Concerns
02	Monitoring Operating System Performance	08	Performance Tuning at Language Level
03	Monitoring JVM and JIT Performance	09	Performance Tuning at API Level
04	Profiling with VisualVM, JDK Flight Recorder and JDK Mission Control	10	Memory Footprint and Startup: Compact Object Headers and AOT Caching
05	Garbage Collection Schemes: G1, ZGC, Generational Shenandoah, Parallel	11	Benchmarking Java Applications (JMH)
06	Garbage Collection Tuning	12	Maximizing Performance with GraalVM and Quarkus

DCL-204 JAVA SE 27

Java SE 27 Programming

Updated for 2027 · formerly "Java SE 26 Programming"

The core Java SE 27 language and APIs — taught the modern way.



DURATION	LEVEL	DELIVERY	WHERE
5 days	Introductory	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

Java SE 27 Programming covers the core language features and APIs you will use to design object-oriented and data-oriented applications on the Java SE 27 platform. Participants start fast with compact source files and JShell, then progress through classes, interfaces, lambdas, records, sealed hierarchies and pattern matching, the module system, collections, streams and gatherers, exceptions, java.time, NIO.2 and virtual threads — clearly distinguishing final features from Java SE 27 previews.

WHAT YOU WILL LEARN

- Write idiomatic Java SE 27 code with records, sealed types and pattern matching
- Design class hierarchies with encapsulation, inheritance, interfaces and polymorphism
- Process data declaratively with lambdas, streams and stream gatherers
- Handle errors, files and time robustly with exceptions, NIO.2 and java.time
- Write scalable concurrent code with virtual threads and the Concurrency API

WHO SHOULD ATTEND

Developers new to Java, developers moving from older Java versions, and programmers from other languages.

PREREQUISITES

Experience with at least one programming language and basic object-oriented concepts.

TECHNOLOGIES

JDK 27 · JShell · Records · Sealed classes · Pattern matching · Streams · Virtual threads · NIO.2

RELATED COURSES

DCL-235 Effective Java Programming
DCL-220 Object-Oriented Programming Principles and Design Patterns with Java SE 27
DCL-374 Developing Spring Boot 4 Applications

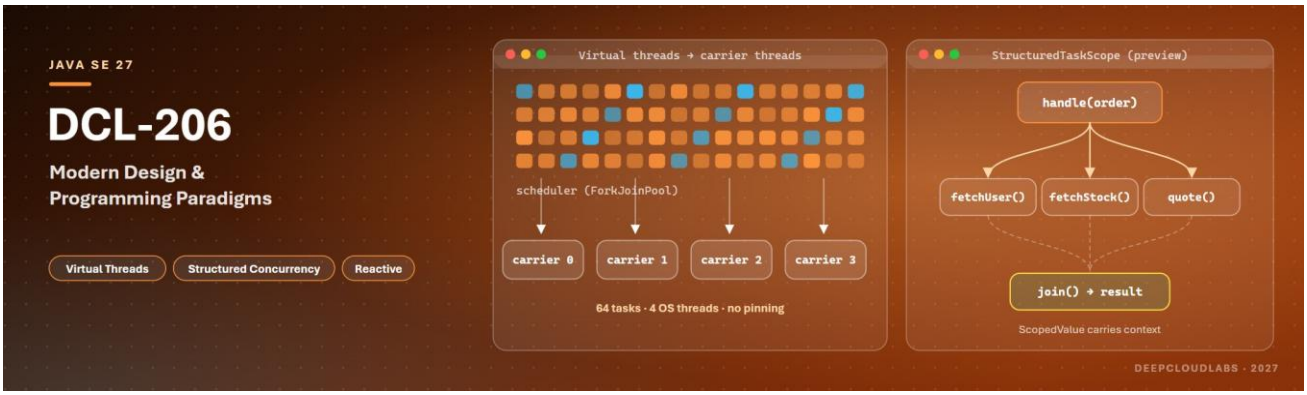
COURSE MODULES

01	Java Platform Overview: JDK 27, Release Cadence and LTS	09	Module System and Module Import Declarations
02	Java Syntax and Class Review: Compact Source Files, Instance Main Methods, JShell	10	Collections, Sequenced Collections and Generics
03	Encapsulation and Subclassing	11	Streams, Filters and Stream Gatherers
04	Overriding Methods, Polymorphism and Static Classes	12	Built-in Functional Interfaces and Lambda Operations
05	Abstract, Nested and Sealed Classes	13	Exceptions and Assertions
06	Records, Enums and Data-Oriented Programming	14	Java Date/Time API
07	Interfaces and Lambda Expressions	15	I/O Fundamentals and NIO.2
08	Pattern Matching: instanceof, switch and Record Patterns	16	Concurrency API and Virtual Threads

DCL-206 JAVA SE 27

Advanced Java Development with Modern Design and Programming Paradigms

Architecture, patterns and the modern Java platform — from Java SE 8 to Java SE 27.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A comprehensive course that equips software engineers, architects and senior developers with a robust understanding of modern software architecture principles, advanced design patterns and the latest advancements in the Java platform from Java SE 8 through Java SE 27. Participants gain both theoretical foundations and practical insight to architect, design and implement scalable, modular and reactive systems in enterprise environments — including the thread programming model built on virtual threads, scoped values and structured concurrency.

WHAT YOU WILL LEARN

- Select architectural styles and patterns for enterprise systems
- Apply GoF design patterns using modern Java features
- Build modular applications with the Java module system
- Combine functional, asynchronous and reactive programming styles
- Design scalable concurrency with virtual threads and structured concurrency

WHO SHOULD ATTEND

Software engineers, architects and senior Java developers.

PREREQUISITES

Solid Java development experience (DCL-204 or equivalent).

TECHNOLOGIES

Java SE 27 · Module system · CompletableFuture · Flow API · Virtual threads · Structured concurrency

RELATED COURSES

DCL-200 Java Performance Tuning and Optimization

DCL-210 Advanced Java Programming

COURSE MODULES

01 Software Architecture and Architectural Patterns
Tiered and layered architecture, MVC, REST · Hexagonal / Clean Architecture · Event patterns and event-driven architecture · SOA, MSA and microservices patterns

02 Design Patterns
Object design fundamentals and OOP principles · Interface, responsibility and construction patterns · Operation and extension patterns

03 New Features in Java SE 8 to 27

04 Modularity and the Java Module System
Motivation and architecture · Creating, compiling and deploying modules · module-info.java, access control, module imports

05 Functional Programming in Java
Higher-order and pure functions, immutability · Lambdas, method references, functional interfaces · Streams and stream gatherers

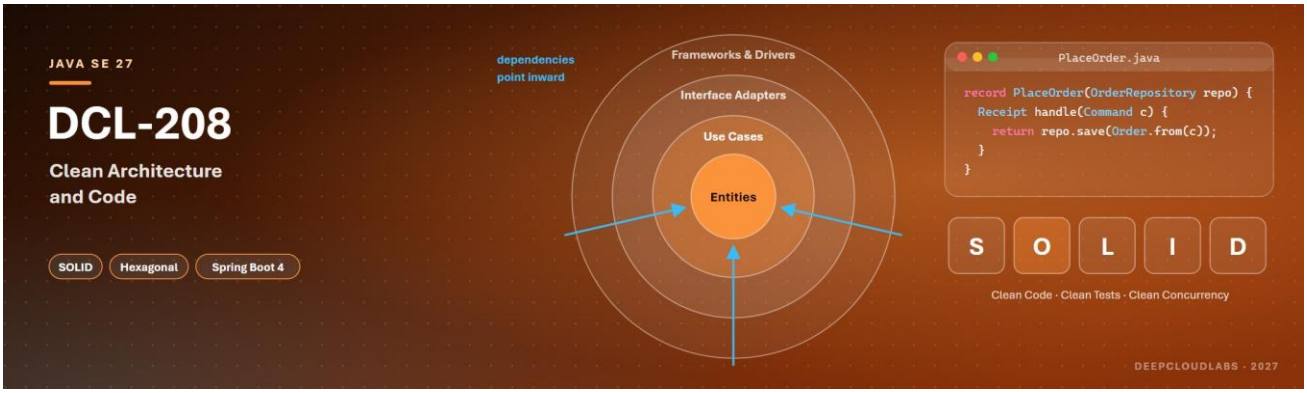
06 Asynchronous and Reactive Programming
CompletableFuture, Future and ExecutorService · Exception handling and chaining · Flow API: Reactive Streams publishers and subscribers

07 Thread Programming Model in Java SE 21+
Platform threads and virtual threads · Scoped values and structured concurrency (preview) · Executors, synchronizers and thread safety

DCL-208 JAVA SE 27

Clean Architecture and Code (Java SE and Spring Boot)

Software that is easy to change is software that lasts.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This course combines the architectural principles of Clean Architecture — the dependency rule, use-case-centric design, boundaries and the SOLID principles — with the day-to-day craft of Clean Code: meaningful names, small functions, honest comments, consistent formatting, careful error handling and clean tests. Examples use Java SE 27 and Spring Boot 4, showing how records, sealed types and pattern matching support clean domain models, and how to keep frameworks at the edges of the system.

WHAT YOU WILL LEARN

- Apply SOLID principles and the dependency rule
- Structure applications around use cases with clear boundaries
- Write readable, intention-revealing code and tests
- Handle errors and concurrency cleanly
- Keep Spring Boot and other frameworks at the edges of the architecture

WHO SHOULD ATTEND

Java developers, technical leads and architects who want maintainable, testable code bases.

PREREQUISITES

Java programming experience; Spring Boot familiarity is helpful.

TECHNOLOGIES

Java SE 27 · Spring Boot 4 · SOLID · Hexagonal architecture · JUnit 6

RELATED COURSES

DCL-220 Object-Oriented Programming Principles and Design Patterns with Java SE 27

DCL-215 Test-Driven Development with JUnit 6

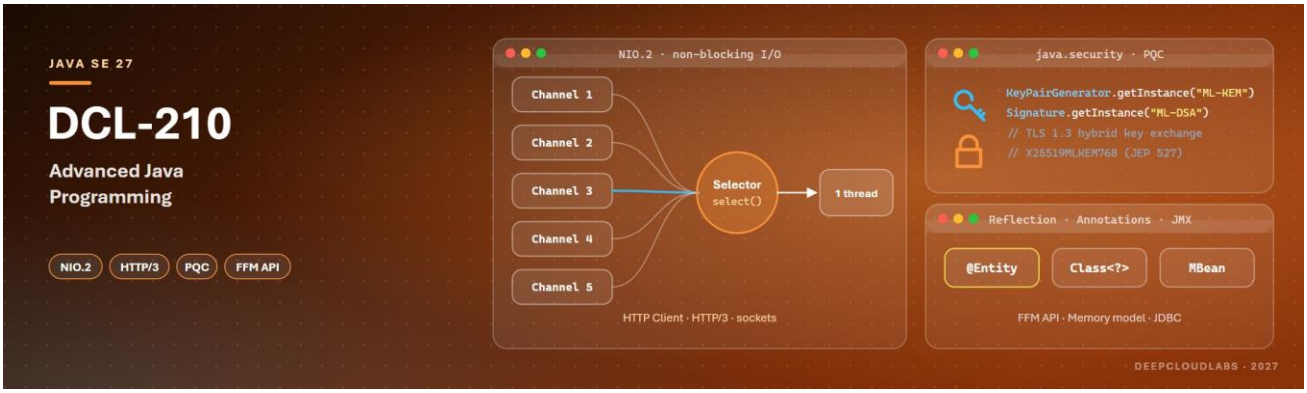
COURSE MODULES

01	Introduction to Software Architectures	08	Comments
02	Introduction to Clean Architecture	09	Formatting
03	SOLID Principles	10	Objects and Data Structures
04	Clean Architecture Components	11	Error Handling
05	Introduction to Clean Code	12	Boundaries
06	Meaningful Names	13	Clean Tests
07	Functions	14	Clean Concurrency

DCL-210 JAVA SE 27

Advanced Java Programming

The APIs and runtime behavior that separate expert Java developers from the rest.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

The goal of this training is to become a better Java programmer and a true master of the Java language: to truly understand threading, Java NIO and the intricacies of the Java memory model, and to use that knowledge to improve the performance of Java applications. The course goes deep into annotations and reflection, collections, streams and the Flow API, distributed programming, JMX, database programming, non-blocking networking with the HTTP Client (now with HTTP/3), the Foreign Function & Memory API, and modern security APIs — including post-quantum algorithms and the hybrid post-quantum key exchange for TLS 1.3 added in Java SE 27.

WHAT YOU WILL LEARN

- Use annotations, reflection and JMX to build introspectable systems
- Master threads, virtual threads and the Java memory model
- Build non-blocking network applications with NIO.2 and the HTTP Client
- Apply modern cryptography, including post-quantum algorithms
- Call native code safely with the Foreign Function & Memory API

WHO SHOULD ATTEND

Experienced Java developers aiming for expert-level platform knowledge.

PREREQUISITES

Solid Java programming experience (DCL-204 or equivalent).

TECHNOLOGIES

Java SE 27 · NIO.2 · HTTP Client · JMX · JDBC · FFM API · ML-KEM · ML-DSA

RELATED COURSES

DCL-200 Java Performance Tuning and Optimization

DCL-206 Advanced Java Development with Modern Design and Programming Paradigms

COURSE MODULES

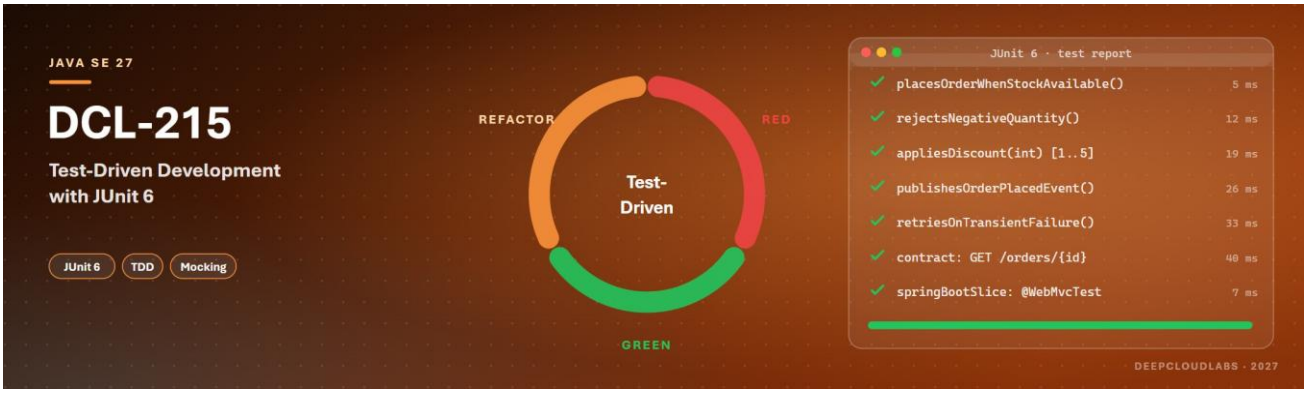
01	Annotations and Reflection API	07	Database Programming: JDBC, JPA, JTA
02	Collection API, Stream API and Flow API	08	NIO and NIO.2
03	XML and JSON Processing	09	Networking: Non-Blocking Sockets, Selectors and the HTTP Client (HTTP/3)
04	RMI and Distributed Programming	10	Security: Signatures, Digests, Ciphers, KDF and Post-Quantum Cryptography
05	JMX and Programming MBeans	11	Foreign Function & Memory API
06	Threads, Virtual Threads and Concurrent Programming	12	New Language Features in Java SE 8–27

DCL-215 JAVA SE 27

Test-Driven Development with JUnit 6

Updated for 2027 · formerly "Test-Driven Development with JUnit 5"

Drive design through tests — with JUnit 6, Java SE 27 and Spring Framework 7.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

JUnit is the standard unit-testing framework for the Java platform and has been central to the practice of test-driven development. In this course participants gain a deep, practical understanding of JUnit 6 — the current generation of the JUnit Jupiter programming model, with a Java 17 baseline, unified versioning and JSpecify nullability annotations — and learn to drive design through tests: from unit and parameterized tests to test doubles, testing strategies for microservices and Spring Boot testing. Teams on JUnit 5 learn a clear migration path.

WHAT YOU WILL LEARN

- Write expressive unit and parameterized tests with JUnit 6
- Practice the red–green–refactor TDD cycle
- Use dummies, stubs, spies and mocks appropriately
- Test microservices and Spring Boot applications at the right levels
- Design for testability and migrate existing JUnit 5 suites

WHO SHOULD ATTEND

Java developers and technical leads adopting or strengthening test-driven development.

PREREQUISITES

Java programming experience (DCL-204 or equivalent).

TECHNOLOGIES

JUnit 6 · Mockito · AssertJ · Spring Boot Test · Testcontainers

RELATED COURSES

DCL-351 Microservices Testing

DCL-208 Clean Architecture and Code (Java SE and Spring Boot)

DCL-606 AI-Assisted Software Testing and Quality Engineering

COURSE MODULES

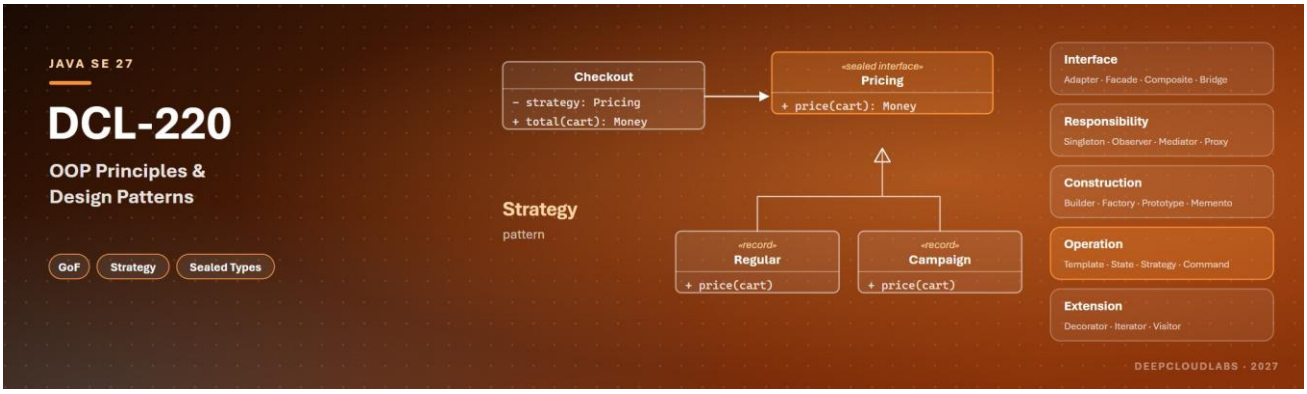
01	Java SE 8–27: New Features Boot Camp	06	Test Doubles: Dummy Object, Stub, Spy, Mock
02	Introduction to JUnit 6 and Migration from JUnit 5	07	Testing Strategies in Microservice Architecture
03	Unit Testing with JUnit 6	08	Spring Boot Testing
04	Microservice Architecture, Domain-Driven Design, Hexagonal Architecture	09	Guidelines for Testable Design (Java SE 27, Spring Framework 7)
05	Test-Driven Development		

DCL-220 JAVA SE 27

Object-Oriented Programming Principles and Design Patterns with Java SE 27

Updated for 2027 · formerly "Object-Oriented Programming Principles and Design Patterns with Java SE 26"

All 23 Gang of Four patterns — expressed with modern Java SE 27.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This course provides an overview of all the Gang of Four (GoF) design patterns as outlined in their seminal book, together with modern-day variations, adjustments and discussions of the intrinsic use of patterns in the Java language. Patterns are expressed with Java SE 27 features — records, sealed interfaces, lambdas and pattern matching — showing where modern Java simplifies or replaces classic implementations.

WHAT YOU WILL LEARN

- Apply core object design principles to real problems
- Recognize and implement all 23 GoF patterns
- Use records, sealed types and lambdas to simplify classic patterns
- Choose between patterns based on forces and trade-offs
- Identify patterns already used inside the Java platform

WHO SHOULD ATTEND

Java developers and designers who want to apply design patterns effectively.

PREREQUISITES

Java programming experience and basic object-oriented concepts.

TECHNOLOGIES

Java SE 27 · UML · GoF patterns · Records · Sealed interfaces

RELATED COURSES

DCL-206 Advanced Java Development with Modern Design and Programming Paradigms

DCL-235 Effective Java Programming

COURSE MODULES

01	Object Design Fundamentals	05	Construction Patterns: Builder, Factory Method, Abstract Factory, Prototype, Memento
02	OOP Principles and Design Patterns	06	Operation Patterns: Template Method, State, Strategy, Command, Interpreter
03	Interface Patterns: Adapter, Facade, Composite, Bridge	07	Extension Patterns: Decorator, Iterator, Visitor
04	Responsibility Patterns: Singleton, Observer, Mediator, Proxy, Chain of Responsibility, Flyweight		

DCL-235 JAVA SE 27

Effective Java Programming

Proven practices for robust, maintainable and efficient Java — updated for Java SE 27.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A three-day course that gives experienced Java developers practical guidance for writing robust, maintainable and efficient Java applications. Participants examine best practices for object creation, classes and interfaces, generics, enums, annotations, methods, exceptions and serialization, and modern techniques including lambdas, streams, concurrency and the correct implementation of methods common to all objects. Guidance is updated for Java SE 27 — including records, sealed types and virtual threads. Participants leave able to apply proven design principles, avoid common pitfalls and improve the reliability, readability and performance of their code.

WHAT YOU WILL LEARN

- Create and destroy objects safely and efficiently
- Design classes, interfaces and generics that are hard to misuse
- Use enums, annotations, lambdas and streams idiomatically
- Handle exceptions and concurrency correctly
- Avoid serialization pitfalls and prefer safer alternatives

WHO SHOULD ATTEND

Experienced Java developers and technical leads.

PREREQUISITES

At least one year of Java development experience.

TECHNOLOGIES

Java SE 27 · Generics · Streams · Records · Virtual threads

RELATED COURSES

DCL-220 Object-Oriented Programming Principles and Design Patterns with Java SE 27

DCL-206 Advanced Java Development with Modern Design and Programming Paradigms

COURSE MODULES

01	Creating and Destroying Objects	07	Methods
02	Methods Common to All Objects	08	General Programming
03	Classes and Interfaces	09	Exceptions
04	Generics	10	Concurrency
05	Enums and Annotations	11	Serialization
06	Lambdas and Streams		

DCL-252 JAVA SE 27

New Features in Java SE 8–27

Updated for 2027 · formerly "New Features in Java SE 8-26"

Everything that changed from Java 8 to Java 27 — and how to upgrade with confidence.



DURATION	LEVEL	DELIVERY	WHERE
2 days	**Intermediate**	**Instructor-led, hands-on**	**Our sites · on-site · your venue**

OVERVIEW

A fast, structured tour of the features and enhancements introduced between Java SE 8 and Java SE 27 — ideal for teams modernizing from Java 8, 11, 17 or 21. The course summarizes language, JVM and API changes release by release and shows how they combine in practice: lambdas and streams, the module system, local-variable type inference, records, sealed classes, pattern matching, virtual threads and scoped values, through to the Java SE 25–27 enhancements, with migration guidance for upgrading to the Java SE 25 LTS release and beyond.

WHAT YOU WILL LEARN

- Summarize the major language, JVM and API changes from Java 8 to 27
- Use lambdas, streams, records, sealed types and pattern matching effectively
- Adopt virtual threads and the newest concurrency APIs safely
- Distinguish final, preview and incubator features
- Plan a low-risk migration to Java SE 25 LTS and Java SE 27

WHO SHOULD ATTEND

Java developers, technical leads and architects planning Java upgrades.

PREREQUISITES

Working knowledge of Java (any version).

TECHNOLOGIES

Java SE 8–27 · Records · Pattern matching · Virtual threads · Modules

RELATED COURSES

DCL-204 Java SE 27 Programming
DCL-206 Advanced Java Development with Modern Design and Programming Paradigms

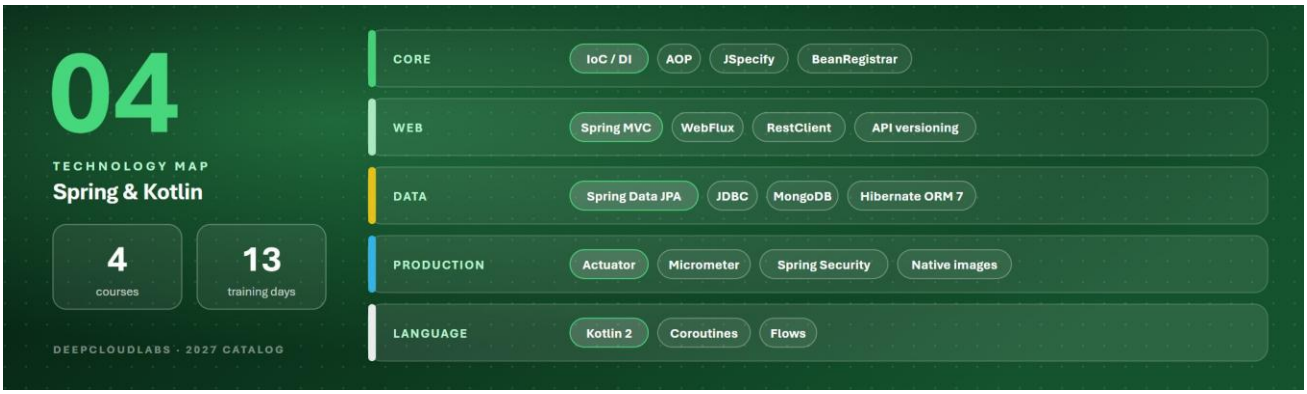
COURSE MODULES

01	Language Changes (Java SE 7–27)	07	Miscellaneous New Features (Java SE 8–27)
02	JVM Changes (Java SE 7–27): GC, AOT and Compact Object Headers	08	Module System (Java SE 9)
03	Changes in APIs (Java SE 8–27)	09	Asynchronous and Reactive Programming (Java SE 9)
04	Using Lambda Expressions and Method Enhancements (Java SE 8)	10	New Language Features in Java SE 9–27
05	Collections and Streams API (Java SE 8–27)	11	What's New in Java SE 25 LTS, 26 and 27
06	Using the Date and Time API (Java SE 8)	12	Migration Strategy: Upgrading from Java 8/11/17/21

TRAINING DOMAIN 04 · 4 COURSES · 13 TRAINING DAYS

04 Spring Ecosystem & Kotlin

Spring Framework 7, Spring Boot 4, Spring Data and Kotlin 2 for production-grade enterprise applications.



Spring technologies help you build enterprise applications that are scalable, secure and robust. Our Spring training empowers developers to solve concrete business problems by mapping application-level issues to Spring-centric solutions. The 2027 curriculum is aligned with Spring Framework 7 and Spring Boot 4, released in November 2025.

WHAT'S NEW IN 2027

- "Spring Framework 6" upgraded to Spring Framework 7.
- "Developing Spring Boot 3 Applications" upgraded to Spring Boot 4.
- Spring Data aligned with Hibernate ORM 7 and Jakarta Persistence 3.2.
- Kotlin course aligned with Kotlin 2 and the K2 compiler.

PROGRAM OUTLINE

CODE	COURSE	LEVEL	DAYS	PAGE
DCL-168	Kotlin Programming	Intermediate	3	43
DCL-370	Spring Framework 7	Intermediate	4	44
DCL-374	Developing Spring Boot 4 Applications	Intermediate	3	45
DCL-376	Spring Data	Intermediate	3	46
Total	4 courses		**13**	

What changes with Spring Framework 7 and Spring Boot 4

TOPIC	2026 EDITION	2027 EDITION
Generation	Spring Framework 6 · Spring Boot 3	Spring Framework 7 · Spring Boot 4 (GA November 2025)
Baselines	Java 17 · Jakarta EE 9/10	Java 17+ (Java 25 supported) · Jakarta EE 11 · Kotlin 2.2
Null safety	Spring-specific annotations	JSpecify annotations across the portfolio
REST APIs	Manual versioning conventions	First-class API versioning in Spring MVC and WebFlux
Resilience	External spring-retry library	Built-in retry and concurrency throttling in Spring Framework
JSON	Jackson 2	Jackson 3 by default
Boot packaging	Single auto-configure module	Modularized auto-configuration

DCL-168 SPRING ECOSYSTEM & KOTLIN

Kotlin Programming

Concise, null-safe, coroutine-powered JVM development with Kotlin 2.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A practical course in Kotlin 2 — the modern, concise and null-safe JVM language supported across the Spring portfolio. Participants master Kotlin's type system, classes and objects, collections and functional programming, then progress to delegates, coroutines and flows for asynchronous programming, and to advanced functional abstractions such as functors, applicatives and monads.

WHAT YOU WILL LEARN

- Write idiomatic, null-safe Kotlin 2 code
- Model data with classes, objects, data classes and delegates
- Apply functional programming techniques to collections and streams
- Write structured asynchronous code with coroutines and flows
- Use functors, applicatives and monads where they add clarity

WHO SHOULD ATTEND

Java developers and JVM engineers adopting Kotlin; Android and back-end developers.

PREREQUISITES

Programming experience in Java or another object-oriented language.

TECHNOLOGIES

Kotlin 2 · K2 compiler · Coroutines · Flow · Gradle

RELATED COURSES

DCL-374 Developing Spring Boot 4 Applications

COURSE MODULES

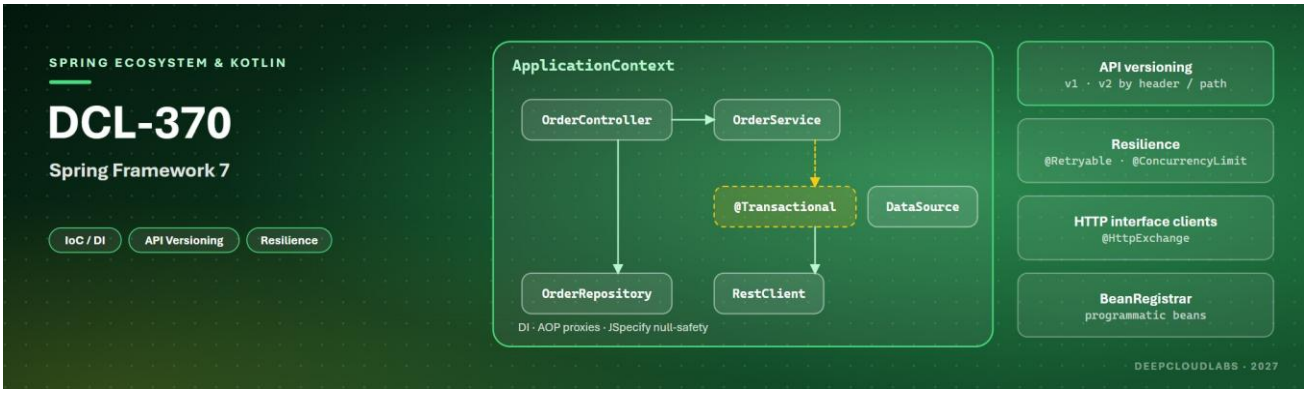
01	Introduction to Kotlin 2 and the K2 Compiler	08	Delegates in Kotlin
02	Control Flow	09	Asynchronous Processing with Coroutines
03	Classes and Objects	10	Collections and Data Operations in Kotlin
04	Collections Framework	11	Functional Programming, OOP and Reactive Programming
05	Getting Started with Functional Programming	12	Monads, Functors and Applicatives
06	Functions: Function Types and Side Effects	13	Working with Streams and Flows
07	Lambdas, Generics, Recursion and Corecursion		

DCL-370 SPRING ECOSYSTEM & KOTLIN

Spring Framework 7

Updated for 2027 · formerly "Spring Framework 6"

The Spring programming model end to end — updated for Spring Framework 7.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

Hands-on experience with Spring Framework 7 and its major features — the IoC container and configuration, AOP, data access and transactions, web and REST applications, testing and Spring Security — to build enterprise-ready applications with Spring Boot 4. The course highlights what is new in Spring Framework 7: JSpecify null-safety across the portfolio, programmatic bean registration, first-class API versioning, HTTP interface clients and built-in resilience features such as retry and concurrency throttling.

WHAT YOU WILL LEARN

- Configure and wire applications with the Spring IoC container
- Apply AOP, transactions and data access with JDBC, ORM and Spring Data
- Build versioned REST APIs and HTTP interface clients
- Add resilience with built-in retry and concurrency limits
- Secure and test Spring applications

WHO SHOULD ATTEND

Java developers building enterprise applications with Spring.

PREREQUISITES

Java programming experience (DCL-204 or equivalent).

TECHNOLOGIES

Spring Framework 7 · Spring Boot 4 · JSpecify · Spring Data · Spring Security · Jackson 3

RELATED COURSES

DCL-374 Developing Spring Boot 4 Applications
DCL-376 Spring Data
DCL-350 Implementing Microservices Architecture using Spring Cloud

COURSE MODULES

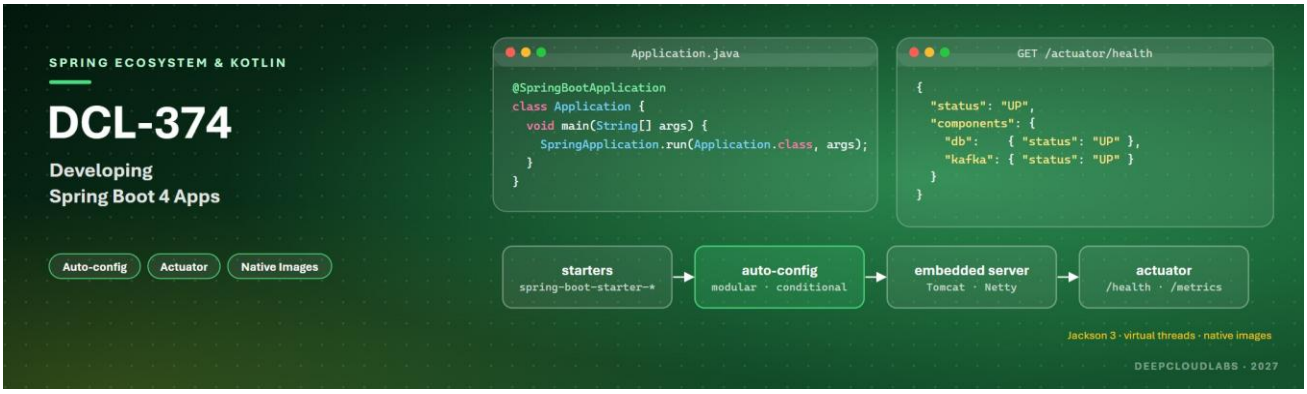
01	Introduction to Spring: Inversion of Control and Containers	08	Spring Data
02	Bean Configuration and Programmatic Bean Registration	09	Spring REST: API Versioning and HTTP Interface Clients
03	Dynamic Proxy and Spring AOP	10	Resilience: Retry and Concurrency Throttling
04	Spring Boot 4	11	Spring Testing
05	Spring JDBC: JdbcTemplate and JdbcClient	12	Spring Web Services
06	Spring Transactions	13	Spring Security
07	Spring ORM		

DCL-374 SPRING ECOSYSTEM & KOTLIN

Developing Spring Boot 4 Applications

Updated for 2027 · formerly "Developing Spring Boot 3 Applications"

Production-ready applications, fast — with Spring Boot 4.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

Hands-on experience with Spring Boot 4 and its core features — configuration, data access, web and REST applications, security, messaging and observability — to build production-ready applications quickly. The course highlights what changed in Spring Boot 4: modularized auto-configuration, Jackson 3 as the default JSON library, JSpecify null-safety, API versioning, HTTP service clients and first-class support for virtual threads, together with Actuator and Micrometer-based observability.

WHAT YOU WILL LEARN

- Bootstrap and configure Spring Boot 4 applications
- Build versioned REST APIs with standard problem details
- Persist data with Spring Data and secure endpoints with Spring Security
- Integrate messaging and observe applications with Actuator and Micrometer
- Package and deploy as containers or native images

WHO SHOULD ATTEND

Java developers building services and web applications with Spring Boot.

PREREQUISITES

Java programming experience; Spring fundamentals are helpful.

TECHNOLOGIES

Spring Boot 4 · Jackson 3 · Spring Data · Spring Security · Actuator · Micrometer · GraalVM

RELATED COURSES

DCL-350 Implementing Microservices Architecture using Spring Cloud

DCL-376 Spring Data

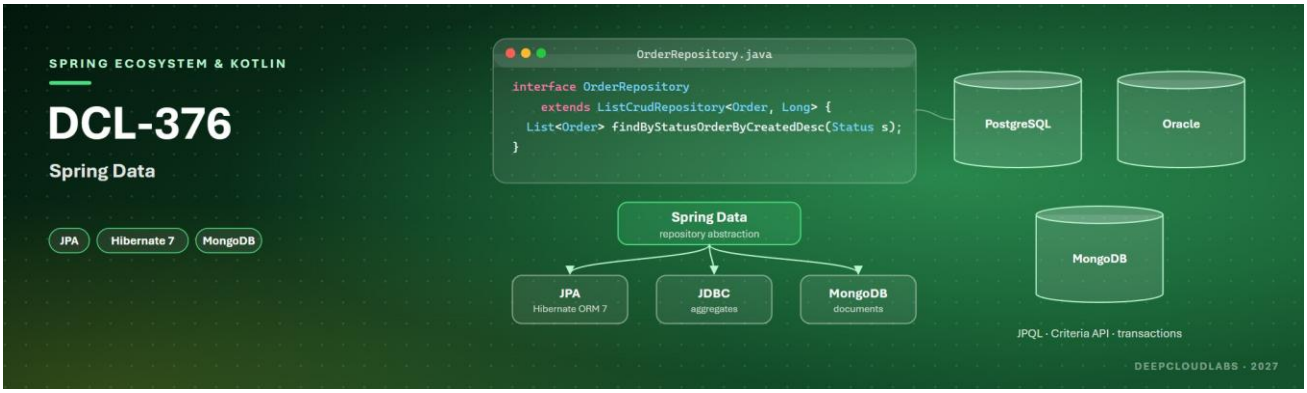
COURSE MODULES

01	Introduction to Spring Boot 4	06	Testing with Spring Boot
02	Auto-Configuration, Starters and Features	07	Spring Security with Spring Boot
03	Spring Boot Essentials: Configuration, Profiles and Logging	08	Spring Messaging with Spring Boot
04	Building REST APIs using Spring Boot	09	Observability with Actuator, Micrometer and OpenTelemetry
05	Spring Data with Spring Boot	10	Deploying Spring Boot Applications: Containers, Virtual Threads, Native Images

DCL-376 SPRING ECOSYSTEM & KOTLIN

Spring Data

Eliminate data-access boilerplate with Spring Data, Hibernate ORM 7 and Jakarta Persistence 3.2.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

Hibernate ORM is the most widely used Jakarta Persistence (JPA) provider, mapping Java classes to relational tables. Spring Data JPA makes Hibernate's powerful features easy to use by removing boilerplate configuration and low-level APIs — and makes it possible to eliminate hand-written DAO implementations entirely. This course covers Spring Data JDBC, JPA with Hibernate ORM 7 and Jakarta Persistence 3.2, and Spring Data MongoDB, on Spring Boot 4.

WHAT YOU WILL LEARN

- Model relational data with JPA entities, relationships and inheritance
- Query with derived queries, JPQL, Criteria API and stored procedures
- Manage transactions correctly with Spring
- Choose between Spring Data JDBC, JPA and MongoDB
- Diagnose and avoid common performance pitfalls

WHO SHOULD ATTEND

Java developers responsible for persistence and data access.

PREREQUISITES

Java and Spring fundamentals; basic SQL.

TECHNOLOGIES

Spring Data · Hibernate ORM 7 · Jakarta Persistence 3.2 · MongoDB · PostgreSQL

RELATED COURSES

DCL-342 Jakarta Persistence 3.2

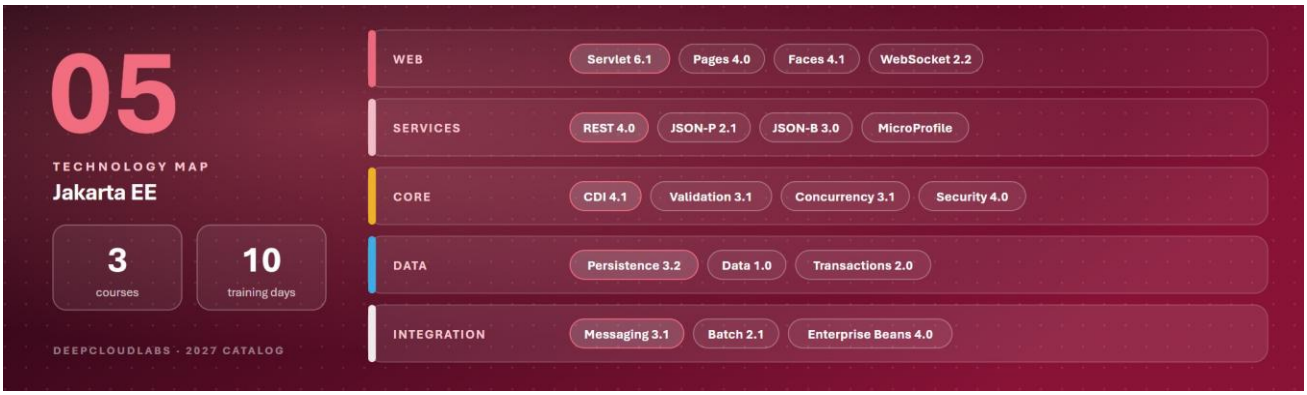
COURSE MODULES

01	Core Spring and Spring Boot Review	08	Mapping Stored Procedures
02	Spring Data JDBC	09	Criteria API
03	Introduction to Jakarta Persistence 3.2	10	Entity Inheritance Relationships
04	Modeling Relational Databases with JPA Entities	11	Spring ORM
05	Working with the Entity Manager	12	Spring Transactions
06	Persisting Enums and Collections	13	Spring Data JPA
07	Java Persistence Query Language (JPQL)	14	Spring Data MongoDB

TRAINING DOMAIN 05 · 3 COURSES · 10 TRAINING DAYS

05 Jakarta EE 11

Standards-based enterprise Java on Jakarta EE 11 — including the new Jakarta Data specification.



Jakarta EE training teaches the concepts, tools and APIs you need to build enterprise and web applications with Jakarta Enterprise Edition. The curriculum is aligned with Jakarta EE 11 (June 2025), which modernizes the platform with a Java SE 17+ baseline, Java 21 support including virtual threads in Jakarta Concurrency, record support, and the new Jakarta Data 1.0 specification, while retiring legacy technologies such as Managed Beans. On completion, participants can create fully functional Jakarta EE applications.

WHAT'S NEW IN 2027

- Program outline unified on Jakarta EE 11 and Jakarta Persistence 3.2.
- Specification versions updated to Jakarta EE 11 (Servlet 6.1, CDI 4.1, Faces 4.1, WebSocket 2.2, Security 4.0 …).
- New module on Jakarta Data 1.0; Jakarta XML Web Services removed from the platform outline.

PROGRAM OUTLINE

CODE	COURSE	LEVEL	DAYS	PAGE
DCL-342	Jakarta Persistence 3.2	Intermediate	3	48
DCL-364	Best Practices in Jakarta EE 11	Advanced	3	49
DCL-390	Developing Enterprise Applications on Jakarta EE 11	Intermediate	4	50
Total	3 courses		**10**	

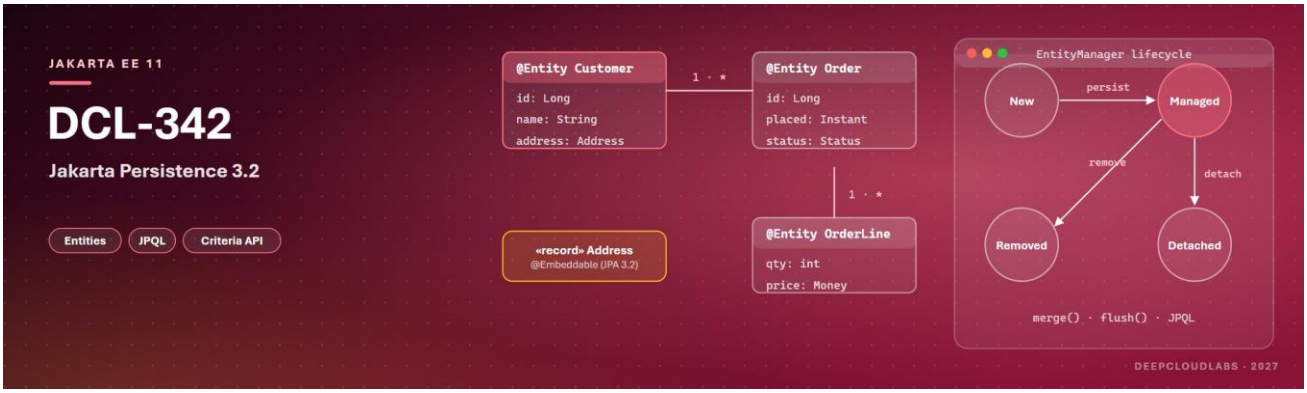
Jakarta EE 11 — what changed

AREA	JAKARTA EE 11
Java baseline	Java SE 17 or higher; Java 21 support including virtual threads
New specification	Jakarta Data 1.0 — repository-based data access
Updated specifications	Servlet 6.1, CDI 4.1, Persistence 3.2, Faces 4.1, Validation 3.1, Concurrency 3.1, Security 4.0, WebSocket 2.2
Removed / retired	Managed Beans; SecurityManager references; optional specifications (e.g. XML Web Services)
Profiles	Core Profile, Web Profile and Platform

DCL-342 JAKARTA EE 11

Jakarta Persistence 3.2

Master object-relational mapping on the Jakarta EE 11 standard.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This training explores the Jakarta Persistence API within the context of a web-based Jakarta EE application, as well as within a stand-alone Java SE application — including its use with Jakarta Enterprise Beans and Contexts and Dependency Injection (CDI). It covers the additions in Jakarta Persistence 3.2, such as Java records as embeddable types, programmatic configuration, schema management and JPQL enhancements.

WHAT YOU WILL LEARN

- Map relational schemas to entities, embeddables and relationships
- Manage entity lifecycles with the EntityManager
- Write type-safe queries with JPQL and the Criteria API
- Apply Bean Validation, transactions and locking
- Optimize persistence performance

WHO SHOULD ATTEND

Java developers building data-driven Jakarta EE or Java SE applications.

PREREQUISITES

Java programming experience and basic SQL.

TECHNOLOGIES

Jakarta Persistence 3.2 · Jakarta EE 11 · CDI · Hibernate ORM · EclipseLink

RELATED COURSES

DCL-390 Developing Enterprise Applications on Jakarta EE 11

DCL-376 Spring Data

COURSE MODULES

01	Introduction to the Jakarta Persistence API	07	Using the Criteria API
02	Working with Jakarta Persistence in a Jakarta EE Environment	08	Implementing Bean Validation with Persistence
03	Modeling Relational Databases with Entities	09	Applying Transactions and Locking
04	Working with the Entity Manager	10	Entity Inheritance Relationships
05	Persisting Enums and Collections	11	Optimizing Persistence Performance
06	Creating Queries with the Jakarta Persistence Query Language		

DCL-364 JAKARTA EE 11

Best Practices in Jakarta EE 11

Patterns, anti-patterns and architecture choices for Jakarta EE 11 and MicroProfile.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This training reviews common and emerging patterns specific to Java SE and Jakarta EE development. Participants learn the depth and evolution of pattern-based techniques in Java, with particular emphasis on Jakarta EE 11 conventions, and progress to Domain-Driven Design and implementing microservices architecture on Jakarta EE 11 with MicroProfile.

WHAT YOU WILL LEARN

- Apply GoF and enterprise integration patterns in Jakarta EE 11
- Implement business and infrastructure patterns with CDI and interceptors
- Recognize and refactor common anti-patterns
- Select patterns that fit the architecture
- Build microservices on Jakarta EE with MicroProfile

WHO SHOULD ATTEND

Senior Java developers and architects working with Jakarta EE.

PREREQUISITES

Jakarta EE development experience (DCL-390 or equivalent).

TECHNOLOGIES

Jakarta EE 11 · CDI 4.1 · MicroProfile · Design patterns · DDD

RELATED COURSES

DCL-352 Domain-Driven Design Essentials

COURSE MODULES

01	Reviewing Object-Oriented Principles in Java	08	Implementing More Infrastructure Patterns
02	Reviewing Gang of Four Patterns	09	Exploring Anti-Patterns
03	Implementing Patterns in Java	10	Selecting Patterns for Architecture
04	Jakarta EE 11: Overview	11	Domain-Driven Design Essentials
05	Implementing Integration Patterns	12	Introduction to Microservices Architecture
06	Implementing Patterns in Business Components	13	Implementing Microservices in Jakarta EE 11 using MicroProfile
07	Implementing Infrastructural Patterns in Jakarta EE 11		

DCL-390 JAKARTA EE 11

Developing Enterprise Applications on Jakarta EE 11

Build and deploy standards-based enterprise applications on the Jakarta EE 11 platform.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This training teaches the skills needed to build and deploy enterprise applications that comply with the Jakarta EE 11 Platform. Participants work across the web, business, persistence, messaging and integration tiers — from Servlets and Faces to CDI, Persistence and the new Jakarta Data specification, RESTful web services, JSON, WebSocket and Jakarta Security — using Java 21 features such as virtual threads through Jakarta Concurrency.

WHAT YOU WILL LEARN

- Build web tiers with Servlet, Pages and Faces
- Implement business logic with CDI and Enterprise Beans
- Persist data with Jakarta Persistence and Jakarta Data
- Expose RESTful, JSON and WebSocket APIs
- Secure, schedule and integrate enterprise applications

WHO SHOULD ATTEND

Java developers building enterprise applications on Jakarta EE application servers.

PREREQUISITES

Java programming experience (DCL-204 or equivalent).

TECHNOLOGIES

Jakarta EE 11 · CDI 4.1 · Persistence 3.2 · Data 1.0 · REST 4.0 · Security 4.0

RELATED COURSES

DCL-342 Jakarta Persistence 3.2
DCL-364 Best Practices in Jakarta EE 11

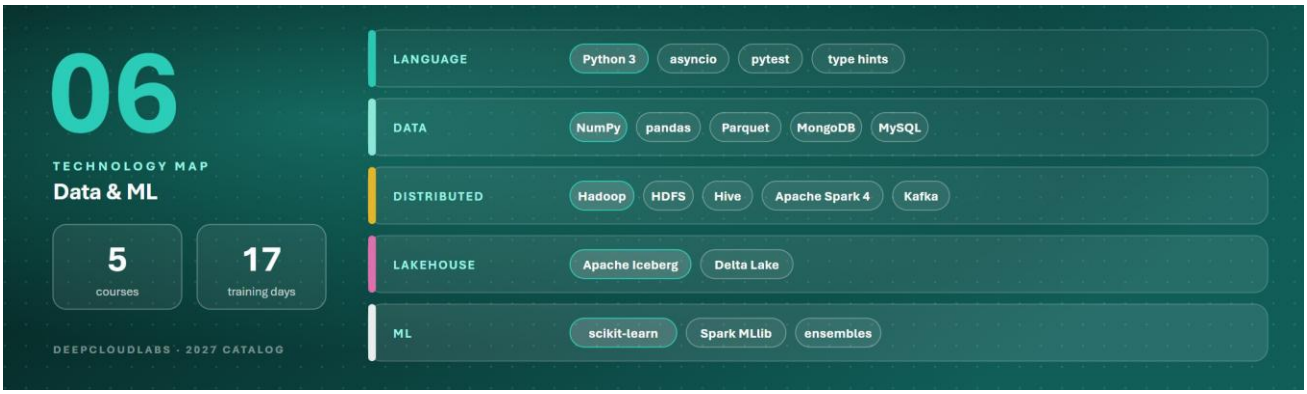
COURSE MODULES

01	Introduction to the Jakarta EE 11 Platform	09	Jakarta Data 1.0
02	Jakarta Servlet 6.1 and Jakarta Pages 4.0	10	Jakarta Transactions 2.0
03	Jakarta Faces 4.1	11	Jakarta Messaging 3.1
04	Jakarta Enterprise Beans 4.0	12	Jakarta Batch 2.1
05	Contexts and Dependency Injection 4.1	13	Jakarta RESTful Web Services 4.0
06	Jakarta Concurrency 3.1 (Virtual Threads)	14	JSON Processing 2.1 and JSON Binding 3.0
07	Jakarta Validation 3.1	15	Jakarta WebSocket 2.2
08	Jakarta Persistence 3.2	16	Jakarta Security 4.0

TRAINING DOMAIN 06 · 5 COURSES · 17 TRAINING DAYS

06 Python, Data Engineering & Machine Learning

From Python fundamentals to distributed data processing and applied machine learning.



Data is the raw material of AI. This track takes participants from Python fundamentals to production-grade analytics and machine learning: idiomatic and advanced Python, distributed data processing with Hadoop and Apache Spark, modern lakehouse table formats, analytics with NumPy and pandas, and applied machine learning with scikit-learn. Together with the Deep Learning & Generative AI track it forms our complete AI engineering curriculum and the basis of the Machine Learning Bootcamp.

WHAT'S NEW IN 2027

- Big Data Essentials extended with Apache Spark 4, Structured Streaming and lakehouse formats (Apache Iceberg, Delta Lake).
- Advanced Python adds FastAPI alongside Flask for REST services.
- Practical Machine Learning adds gradient boosting and model-selection pipelines.

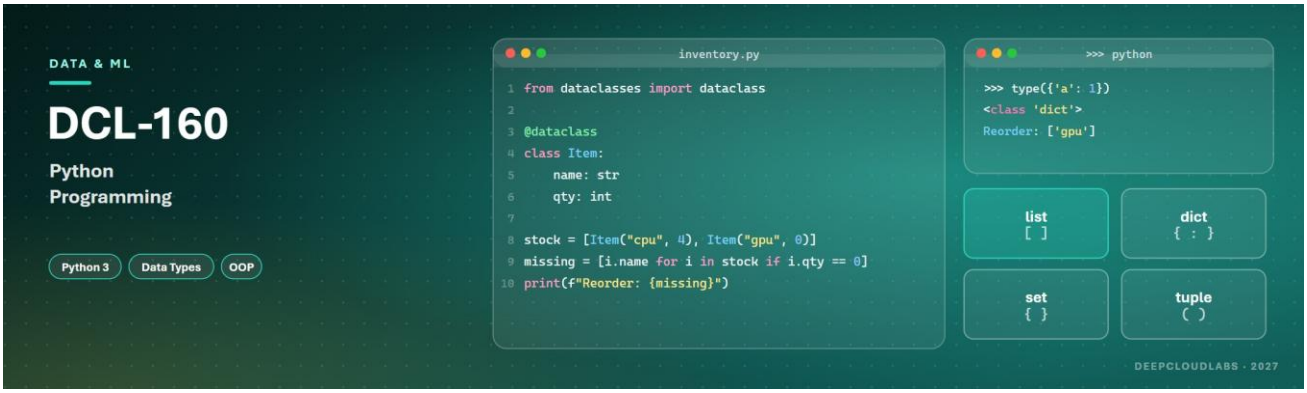
PROGRAM OUTLINE

CODE	COURSE	LEVEL	DAYS	PAGE
DCL-160	Python Programming	Introductory	4	52
DCL-162	Advanced Python Programming	Intermediate	3	53
DCL-400	Big Data Essentials	Introductory	3	54
DCL-402	Data Analytics using Python	Introductory	3	55
DCL-410	Practical Machine Learning using Python	Intermediate	4	56
Total	5 courses		**17**	

DCL-160 PYTHON, DATA ENGINEERING & MACHINE LEARNING

Python Programming

A practical, confident start in Python — no prior programming required.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Introductory	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A practical introduction to Python programming. Participants build a solid foundation in core Python concepts — variables, data structures, control flow, functions, error handling, file operations, object-oriented programming, modules and packages — and learn to use the Python Standard Library effectively, finishing with use-case projects. DCL-160 is designed for beginners with little or no programming experience; prior exposure to programming is helpful but not required.

WHAT YOU WILL LEARN

- Write clear Python programs using core data types and control flow
- Structure code with functions, modules and packages
- Handle errors and work with files
- Apply object-oriented programming in Python
- Use the Standard Library to solve everyday problems

WHO SHOULD ATTEND

Beginners, analysts, engineers and IT professionals starting with Python.

PREREQUISITES

None. Prior exposure to programming is helpful but not required.

TECHNOLOGIES

Python 3 · Standard Library · venv / pip · Jupyter

RELATED COURSES

DCL-162 Advanced Python Programming
DCL-402 Data Analytics using Python

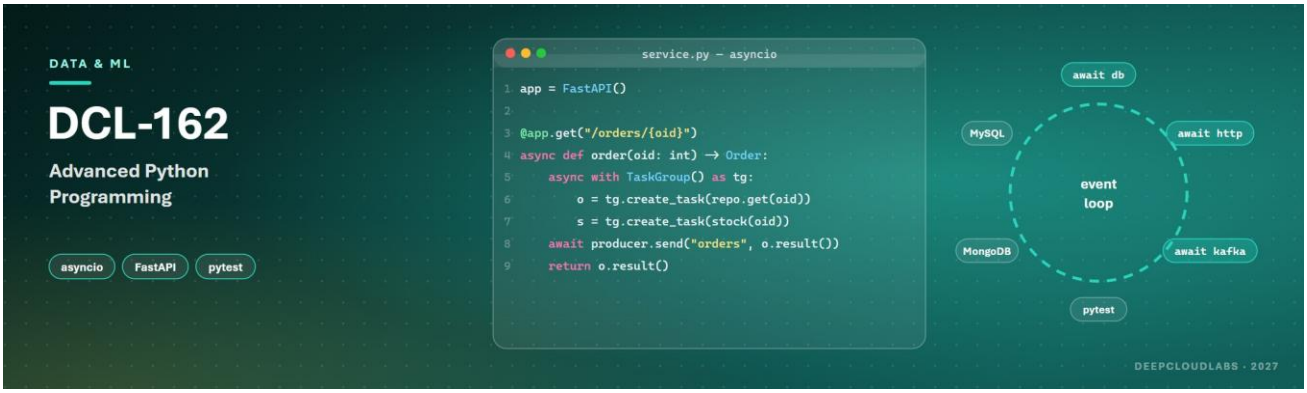
COURSE MODULES

01	Introduction to Programming and Basics of Python	07	Error and Exception Handling
02	Variables and Expressions	08	File Operations
03	Python Data Types	09	Object-Oriented Programming
04	Conditional Control Statements	10	Modules and Packages
05	Loop Control Statements	11	Python Standard Library
06	Defining Functions for Code Reuse	12	Python Projects: Use-Case Examples

DCL-162 PYTHON, DATA ENGINEERING & MACHINE LEARNING

Advanced Python Programming

From scripts to services: OOP, functional style, testing, async I/O, databases and APIs.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

Building on DCL-160, this training expands core concepts in greater depth and introduces advanced topics: an in-depth treatment of classes and object-oriented design, functional programming, file-based data, unit testing with pytest, concurrent and asynchronous programming, database integration with MySQL and MongoDB, building RESTful services with Flask and FastAPI, and event-driven architectures with Apache Kafka.

WHAT YOU WILL LEARN

- Design maintainable object-oriented and functional Python code
- Test code systematically with pytest
- Write concurrent and asynchronous programs with asyncio
- Integrate relational and document databases
- Build REST services and event-driven integrations

WHO SHOULD ATTEND

Python developers moving to professional, production-grade development.

PREREQUISITES

Python fundamentals (DCL-160 or equivalent).

TECHNOLOGIES

Python 3 · pytest · asyncio · FastAPI · Flask · MySQL · MongoDB · Apache Kafka

RELATED COURSES

DCL-402 Data Analytics using Python
DCL-410 Practical Machine Learning using Python

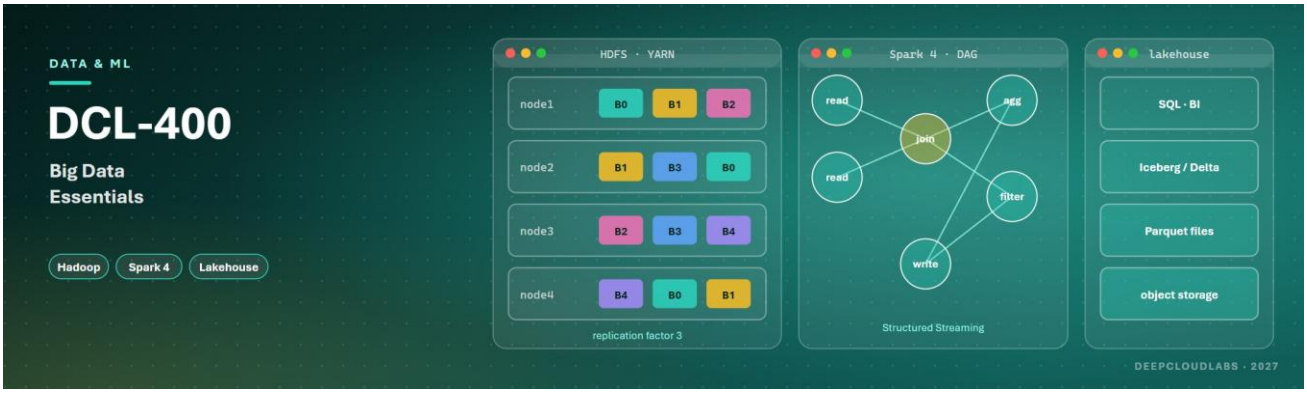
COURSE MODULES

01	Object-Oriented Programming in Python	07	MySQL Programming in Python
02	Functional Programming in Python	08	MongoDB Programming in Python
03	Unit Testing in Python using pytest	09	Introduction to Web Architectures
04	File Operations in Python	10	Designing and Implementing RESTful Services with Flask and FastAPI
05	XML and JSON Processing in Python	11	Event-Driven Architecture using Apache Kafka
06	Concurrent and Asynchronous Programming (asyncio, threading)		

DCL-400 PYTHON, DATA ENGINEERING & MACHINE LEARNING

Big Data Essentials

Big data platforms end to end — Hadoop foundations, Apache Spark 4 and the lakehouse.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Introductory	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A structured, end-to-end overview of big data platforms and analytics, from core concepts to production-ready processing pipelines. Participants explore the Hadoop ecosystem — HDFS, MapReduce, YARN, cluster planning, installation and job management — then move to modern distributed analytics with Apache Spark 4, covering Spark SQL and Structured Streaming for batch and real-time workloads and the lakehouse table formats Apache Iceberg and Delta Lake. The course connects the platform layer to business value with data science and machine learning fundamentals, culminating in machine learning with Spark MLlib.

WHAT YOU WILL LEARN

- Explain distributed storage and processing with HDFS, MapReduce and YARN
- Plan, install and operate a Hadoop cluster and its jobs
- Process batch and streaming data with Spark SQL and Structured Streaming
- Organize analytical data with lakehouse table formats
- Build machine learning workflows with Spark MLlib

WHO SHOULD ATTEND

Data engineers, developers, architects and analysts entering big data.

PREREQUISITES

Basic programming (Python or Java) and SQL knowledge.

TECHNOLOGIES

Hadoop · HDFS · YARN · Hive · Apache Spark 4 · Structured Streaming · Apache Iceberg · Delta Lake · MLlib

RELATED COURSES

DCL-402 Data Analytics using Python
DCL-410 Practical Machine Learning using Python

COURSE MODULES

01	Understanding Big Data	09	Apache Spark 4
02	Understanding Hadoop	10	Apache Spark SQL and DataFrames
03	HDFS	11	Spark Structured Streaming
04	MapReduce	12	Lakehouse Table Formats: Apache Iceberg and Delta Lake
05	Hadoop Ecosystem and YARN	13	Data Science
06	Planning a Hadoop Cluster	14	Machine Learning
07	Hadoop Installation and Managing Jobs	15	Machine Learning with Spark MLlib
08	Apache Hive		

DCL-402 PYTHON, DATA ENGINEERING & MACHINE LEARNING

Data Analytics using Python

End-to-end analytics with NumPy, pandas and visualization — on real-world data.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Introductory	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This course equips participants with the practical skills to perform end-to-end data analytics in Python. After a focused refresher on Python fundamentals, it moves into the analytics stack with NumPy for efficient vectorized computation and pandas for data manipulation at scale. Participants learn to load and store data across common formats, clean and prepare messy datasets, join, reshape and combine data, visualize and explore it, and aggregate it to extract actionable insights — concluding with hands-on analytics practice on real-world datasets.

WHAT YOU WILL LEARN

- Compute efficiently with NumPy arrays and vectorization
- Load, clean and transform data with pandas
- Join, reshape and aggregate datasets
- Create clear visualizations for exploration and reporting
- Apply a reproducible analytics workflow to real data

WHO SHOULD ATTEND

Analysts, engineers and developers who work with data.

PREREQUISITES

Basic programming knowledge; Python fundamentals helpful (DCL-160).

TECHNOLOGIES

Python 3 · NumPy · pandas · Matplotlib · Jupyter · Parquet

RELATED COURSES

DCL-410 Practical Machine Learning using Python

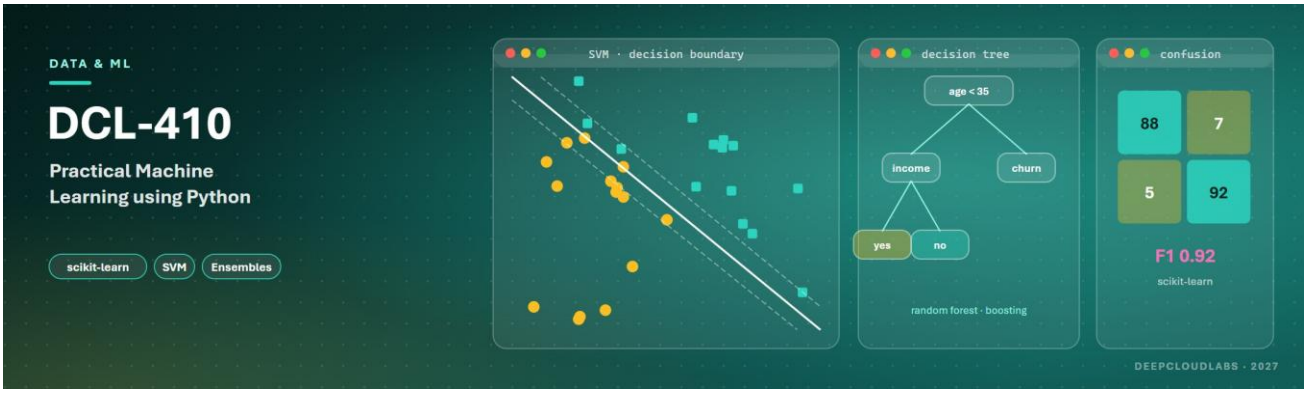
COURSE MODULES

01	Python Language Basics	06	Data Cleaning and Preparation
02	Built-in Data Structures, Functions and Files	07	Data Wrangling: Join, Combine and Reshape
03	NumPy Basics: Arrays and Vectorized Computation	08	Plotting and Visualization
04	pandas	09	Data Aggregation and Group Operations
05	Data Loading, Storage and File Formats	10	Data Analytics Practice with Real-World Data

DCL-410 PYTHON, DATA ENGINEERING & MACHINE LEARNING

Practical Machine Learning using Python

Build, evaluate and improve real machine learning models with scikit-learn.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A hands-on introduction to practical machine learning that guides participants from foundational concepts to building and evaluating real models. It begins with core ML principles anchored in a structured project workflow, develops proficiency in supervised learning with a focus on classification and model training, and explores widely used algorithms — support vector machines, decision trees and ensemble methods including random forests and gradient boosting. The course covers dimensionality reduction, unsupervised learning and model selection, and concludes with an introduction to artificial neural networks.

WHAT YOU WILL LEARN

- Run an end-to-end machine learning project workflow
- Train and evaluate classification and regression models
- Apply SVMs, decision trees and ensemble methods
- Reduce dimensionality and discover structure with unsupervised learning
- Select and tune models with pipelines and cross-validation

WHO SHOULD ATTEND

Developers, analysts and engineers starting with machine learning.

PREREQUISITES

Python programming and basic statistics (DCL-402 recommended).

TECHNOLOGIES

scikit-learn · NumPy · pandas · Gradient boosting · Jupyter

RELATED COURSES

DCL-426 Deep Learning with PyTorch
DCL-428 Deep Learning with TensorFlow

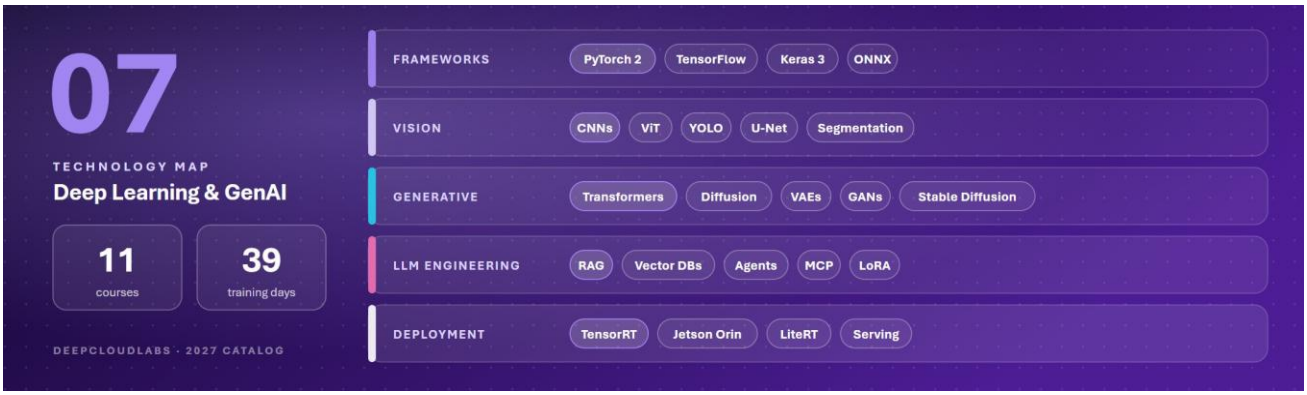
COURSE MODULES

01	Introduction to Machine Learning	07	Ensemble Learning: Random Forests and Gradient Boosting
02	Machine Learning Project	08	Dimensionality Reduction
03	Classification	09	Unsupervised Learning Techniques
04	Training Models	10	Model Selection, Pipelines and Evaluation
05	Support Vector Machines	11	Introduction to Artificial Neural Networks
06	Decision Trees		

TRAINING DOMAIN 07 · 11 COURSES · 39 TRAINING DAYS

07 Deep Learning & Generative AI

Computer vision, landmark architectures, LLMs, diffusion models and AI at the edge.



From convolutional networks to large language models, this track covers the techniques behind today's AI systems — and how to build, adapt and deploy them responsibly. Courses span computer vision and medical imaging, the PyTorch and TensorFlow/Keras frameworks, the landmark research papers that defined modern architectures, generative AI with transformers and diffusion models, LLM applications with retrieval-augmented generation (RAG) and agents, and AI inference at the edge on NVIDIA Jetson.

WHAT'S NEW IN 2027

- Generative AI courses extended with RAG, vector databases, AI agents, tool calling (Model Context Protocol), LoRA fine-tuning, evaluation and guardrails.
- Computer vision adds Vision Transformers and vision foundation models.
- Edge AI course updated to the NVIDIA Jetson Orin Nano platform with JetPack and TensorRT.
- Framework courses updated to PyTorch 2 (torch.compile, torch.export) and Keras 3 / LiteRT.
- Engineering deep dives on LLM applications, RAG, agents, LLMOps and AI security now form domain 08, AI-Native Software Engineering.

PROGRAM OUTLINE

CODE	COURSE	LEVEL	DAYS	PAGE
DCL-422	Deep Learning for Computer Vision	Intermediate	4	59
DCL-424	Deep Learning for Medical Image Analysis	Intermediate	4	60
DCL-426	Deep Learning with PyTorch	Intermediate	4	61
DCL-428	Deep Learning with TensorFlow	Intermediate	4	62
DCL-430	Foundation of Deep Learning with Backbone Architecture Papers	Advanced	4	63
DCL-432	Deep Learning with Backbone Learning Paradigm Papers	Advanced	4	64
DCL-434	Introduction to Generative AI with Python	Intermediate	3	65
DCL-436	Advanced Generative AI with Python	Advanced	4	66
DCL-438	Introduction to ChatGPT	Introductory	2	67
DCL-440	Introduction to Image Generation with Stable Diffusion	Introductory	2	68
DCL-460	Edge Computing: Deployment & Inference with NVIDIA Jetson	Intermediate	4	69
Total	11 courses		**39**	

Generative AI course ladder

COURSE	FOR	LEVEL	DAYS	HANDS-ON FOCUS
DCL-438 Introduction to ChatGPT	Professionals, managers, developers	Introductory	2	Prompting, productivity, OpenAI API
DCL-440 Stable Diffusion	Designers, creatives, developers	Introductory	2	Text-to-image, ControlNet, ComfyUI
DCL-434 Introduction to GenAI	Python developers	Intermediate	3	VAEs, GANs, transformers, diffusion, RAG basics
DCL-436 Advanced GenAI	ML / AI engineers	Advanced	4	Fine-tuning, RAG, agents & MCP, evaluation

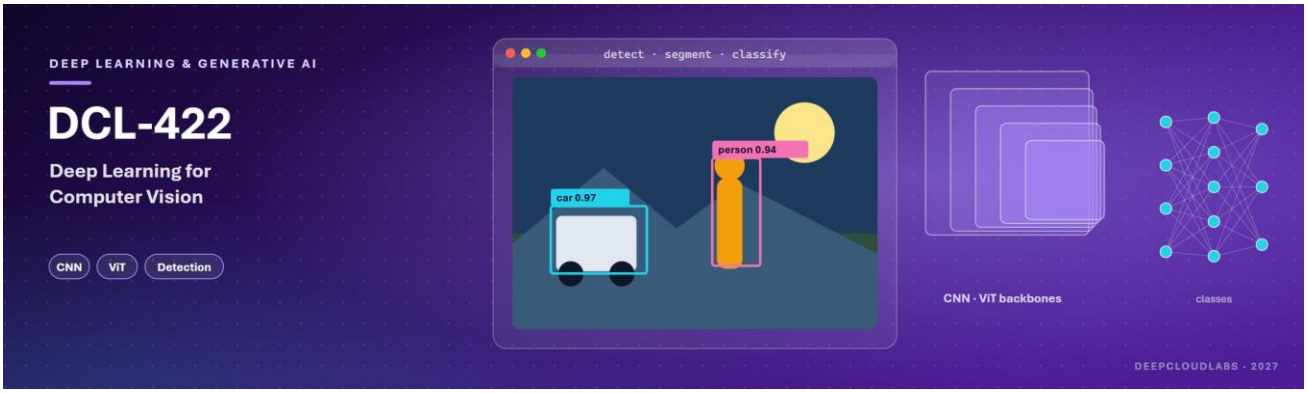
PyTorch or TensorFlow?

ASPECT	DCL-426 PYTORCH	DCL-428 TENSORFLOW
Programming model	Eager, Pythonic; torch.compile for speed	Keras 3 high-level APIs on TensorFlow
Deployment path	torch.export, ONNX, production serving	TensorFlow Serving, LiteRT on device
Shared curriculum	CNNs, sequence models, generative AI, vision and NLP applications	CNNs, sequence models, generative AI, vision and NLP applications

DCL-422 DEEP LEARNING & GENERATIVE AI

Deep Learning for Computer Vision

Classification, detection, segmentation and generation — with CNNs and Vision Transformers.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

An applied, end-to-end introduction to deep learning for computer vision that combines foundational concepts with hands-on model development. Participants ground themselves in machine learning and deep learning, explore core vision tasks and open datasets, and after a focused Python refresher build convolutional neural networks with PyTorch or TensorFlow through the standard training pipeline. They leverage pre-trained models through transfer learning and fine-tuning, are introduced to Vision Transformers and vision foundation models, and apply everything to image classification, object detection, semantic segmentation and image generation.

WHAT YOU WILL LEARN

- Build and train CNNs for vision tasks
- Use transfer learning and fine-tuning to accelerate results
- Apply Vision Transformers and foundation models
- Implement classification, detection and segmentation pipelines
- Evaluate models with task-appropriate metrics

WHO SHOULD ATTEND

Developers, engineers and researchers building vision applications.

PREREQUISITES

Python programming; basic machine learning concepts (DCL-410 recommended).

TECHNOLOGIES

PyTorch · TensorFlow · CNNs · ViT · YOLO · OpenCV

RELATED COURSES

DCL-430 Foundation of Deep Learning with Backbone Architecture Papers

DCL-460 Edge Computing: Deployment & Inference with NVIDIA Jetson

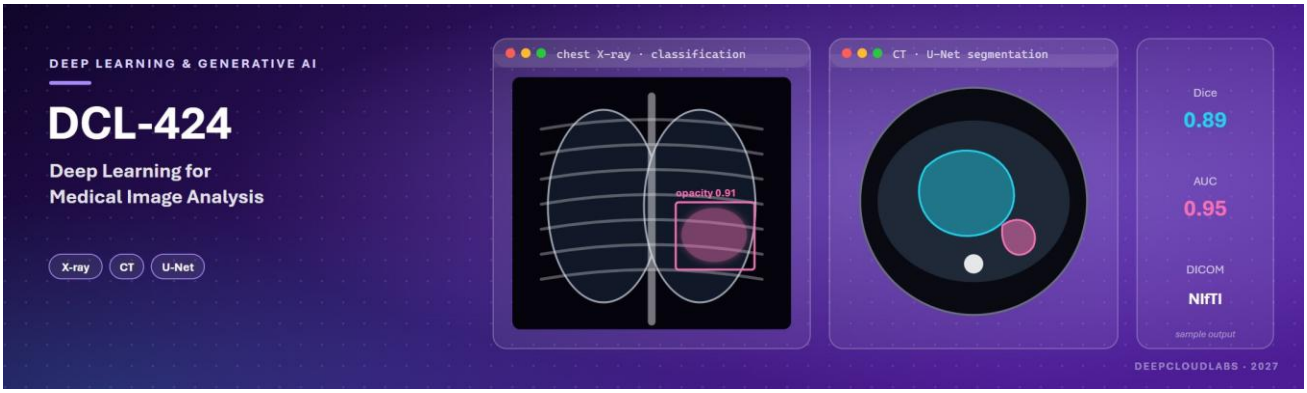
COURSE MODULES

01	Fundamentals of Machine Learning	08	Fundamental Model Development Pipeline
02	Introduction to Deep Learning	09	Pre-trained Models, Transfer Learning and Fine-Tuning
03	Introduction to Computer Vision	10	Vision Transformers and Vision Foundation Models
04	Open Datasets for Computer Vision Tasks	11	Image Classification
05	Python Review for Deep Learning	12	Object Detection
06	Basics of Deep Learning Frameworks (PyTorch or TensorFlow)	13	Semantic Segmentation
07	Convolutional Neural Networks	14	Image Generation

DCL-424 DEEP LEARNING & GENERATIVE AI

Deep Learning for Medical Image Analysis

Deep learning applied to medical imaging — from DICOM data to clinical-grade evaluation.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This course applies deep learning to medical imaging, from data handling to clinically meaningful tasks. Participants work with open medical imaging datasets and standard formats, learn medical image processing, and build a complete model development pipeline on medical data with PyTorch or TensorFlow. Hands-on case studies cover chest X-ray classification and CT segmentation using pre-trained models, transfer learning and fine-tuning, with attention to evaluation metrics and the responsible use of AI in clinical contexts.

WHAT YOU WILL LEARN

- Prepare and process medical images in standard formats
- Build a model development pipeline on medical data
- Classify chest X-rays with transfer learning
- Segment CT images with encoder–decoder networks
- Evaluate medical AI models with appropriate metrics

WHO SHOULD ATTEND

Engineers, researchers and data scientists in healthcare and medical technology.

PREREQUISITES

Python programming; basic machine learning concepts.

TECHNOLOGIES

PyTorch · TensorFlow · DICOM · NIfTI · U-Net · MONAI

RELATED COURSES

DCL-430 Foundation of Deep Learning with Backbone Architecture Papers

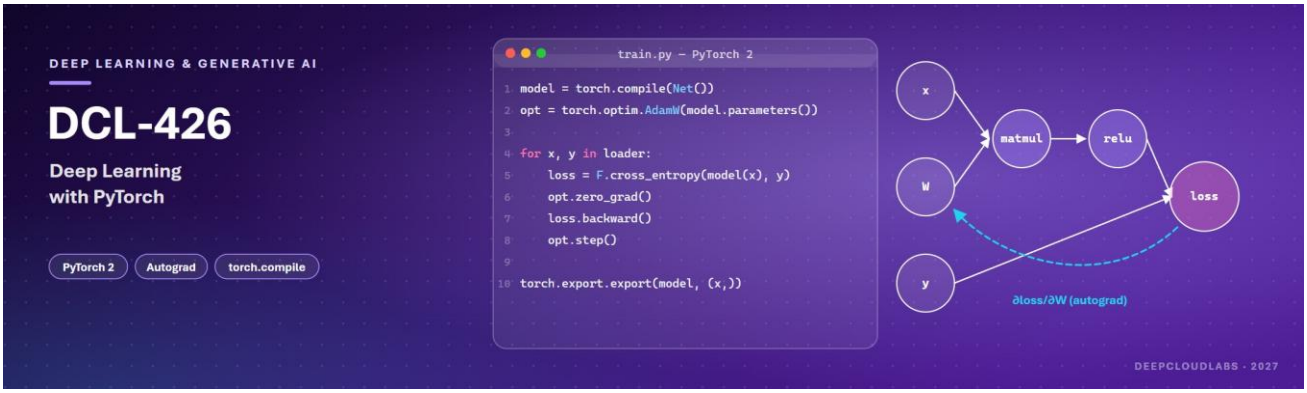
COURSE MODULES

01	Fundamentals of Machine Learning	07	Medical Image Processing
02	Introduction to Deep Learning	08	Fundamental Model Development Pipeline (Medical Image Data)
03	Open Datasets for Medical Imaging Tasks	09	Pre-trained Models, Transfer Learning and Fine-Tuning
04	Python Review for Deep Learning	10	Medical Image Classification (Chest X-ray)
05	Basics of Deep Learning Frameworks (PyTorch or TensorFlow)	11	Medical Image Segmentation (CT)
06	Convolutional Neural Networks		

DCL-426 DEEP LEARNING & GENERATIVE AI

Deep Learning with PyTorch

Deep learning from first principles to deployment with PyTorch 2.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A comprehensive, hands-on course in PyTorch 2. Participants build and train deep neural networks from first principles — tensors, autograd, modules and optimizers — then apply them to convolutional networks, deep sequence models and generative AI, with applications in computer vision and natural language processing. The course closes with model optimization using torch.compile and with exporting and serving trained models.

WHAT YOU WILL LEARN

- Build models with tensors, autograd and nn.Module
- Train deep networks with modern optimizers and regularization
- Implement CNNs, sequence models and generative models
- Apply PyTorch to vision and NLP tasks
- Optimize, export and serve models

WHO SHOULD ATTEND

Developers, ML engineers and researchers adopting PyTorch.

PREREQUISITES

Python programming and machine learning fundamentals (DCL-410 recommended).

TECHNOLOGIES

PyTorch 2 · torch.compile · torch.export · ONNX · Hugging Face

RELATED COURSES

DCL-434 Introduction to Generative AI with Python
DCL-430 Foundation of Deep Learning with Backbone Architecture Papers

COURSE MODULES

01	Short Review of Machine Learning	06	Deep Sequence Modeling
02	Introduction to Deep Learning	07	Generative AI
03	Basics of PyTorch	08	Computer Vision Applications
04	Training Deep Neural Networks	09	Natural Language Processing Applications
05	Convolutional Neural Networks	10	Optimizing, Exporting and Serving Models (torch.compile, torch.export, ONNX)

DCL-428 DEEP LEARNING & GENERATIVE AI

Deep Learning with TensorFlow

Deep learning with TensorFlow and Keras 3 — from first model to production serving.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A comprehensive, hands-on course in TensorFlow and Keras 3. Participants build and train deep neural networks, then apply them to convolutional networks, deep sequence models and generative AI, with applications in computer vision and natural language processing. The course closes with serving models in production with TensorFlow Serving and on-device with LiteRT (formerly TensorFlow Lite).

WHAT YOU WILL LEARN

- Build models with the Keras 3 APIs on TensorFlow
- Train and tune deep networks effectively
- Implement CNNs, sequence models and generative models
- Apply TensorFlow to vision and NLP tasks
- Serve models in the cloud and on device

WHO SHOULD ATTEND

Developers, ML engineers and researchers adopting TensorFlow.

PREREQUISITES

Python programming and machine learning fundamentals (DCL-410 recommended).

TECHNOLOGIES

TensorFlow · Keras 3 · TensorFlow Serving · LiteRT

RELATED COURSES

DCL-434 Introduction to Generative AI with Python

COURSE MODULES

01	Short Review of Machine Learning	06	Deep Sequence Modeling
02	Introduction to Deep Learning	07	Generative AI
03	Basics of TensorFlow and Keras 3	08	Computer Vision Applications
04	Training Deep Neural Networks	09	Natural Language Processing Applications
05	Convolutional Neural Networks	10	Serving Deep Learning Models (TensorFlow Serving, LiteRT)

DCL-430 DEEP LEARNING & GENERATIVE AI

Foundation of Deep Learning with Backbone Architecture Papers

Read, understand and re-implement the papers that defined modern deep learning.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This training gives participants hands-on experience with the backbone papers of deep learning together with their theoretical foundations. Participants gain a solid understanding of commonly used architectures, learn to implement them from scratch, and become familiar with the datasets used for computer vision and image recognition — becoming proficient in PyTorch along the way. The 2027 edition adds the Vision Transformer, the backbone of many modern vision systems.

WHAT YOU WILL LEARN

- Read and critically analyze landmark deep learning papers
- Implement AlexNet, VGG, ResNet, DenseNet and U-Net from scratch
- Explain the design choices behind each architecture
- Visualize and interpret convolutional networks
- Implement and train a Vision Transformer

WHO SHOULD ATTEND

ML engineers, researchers and advanced practitioners.

PREREQUISITES

Python, deep learning fundamentals and linear algebra basics.

TECHNOLOGIES

PyTorch · CNNs · ResNet · U-Net · ViT · ImageNet

RELATED COURSES

DCL-432 Deep Learning with Backbone Learning Paradigm Papers

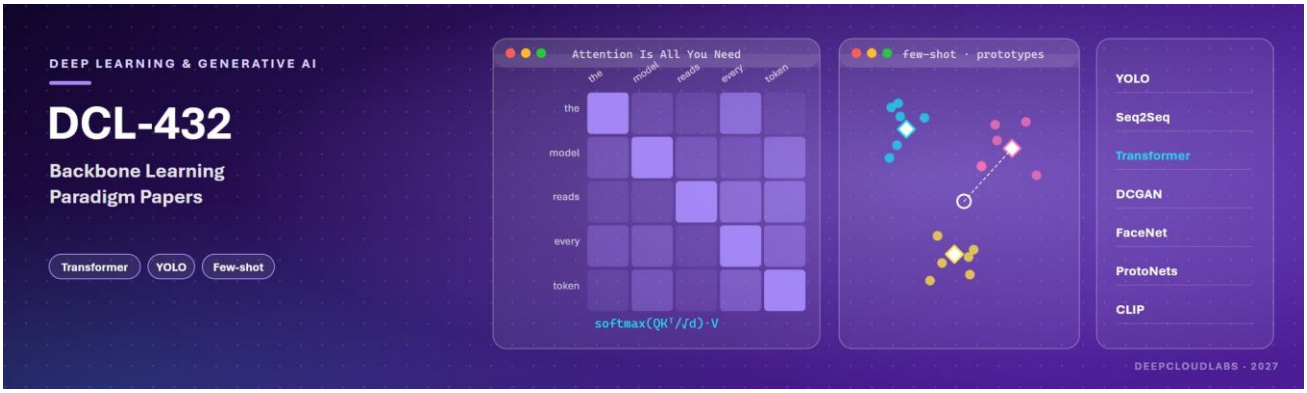
COURSE MODULES

01	Introduction to Deep Learning	06	VGG Net
02	Python Review for Deep Learning	07	ResNet
03	Intensive PyTorch Training	08	DenseNet
04	Visualizing and Understanding Convolutional Networks	09	U-Net
05	AlexNet	10	Vision Transformer (ViT)

DCL-432 DEEP LEARNING & GENERATIVE AI

Deep Learning with Backbone Learning Paradigm Papers

Attention, detection, generation and few-shot learning — straight from the source papers.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This training gives participants hands-on experience with diverse applications of deep learning — natural language processing, computer vision and image generation — as well as advanced CNN features and various learning paradigms, each studied through its landmark paper. Participants gain a broad view of deep learning and its best practices, deep PyTorch knowledge, and an introduction to contrastive and self-supervised learning.

WHAT YOU WILL LEARN

- Implement YOLO-style object detection
- Explain and implement attention and the Transformer
- Build sequence-to-sequence and generative adversarial models
- Apply metric learning and few-shot learning
- Understand contrastive and self-supervised learning

WHO SHOULD ATTEND

ML engineers, researchers and advanced practitioners.

PREREQUISITES

Python and deep learning fundamentals; DCL-430 recommended.

TECHNOLOGIES

PyTorch · Transformers · YOLO · GANs · Metric learning · CLIP

RELATED COURSES

DCL-436 Advanced Generative AI with Python

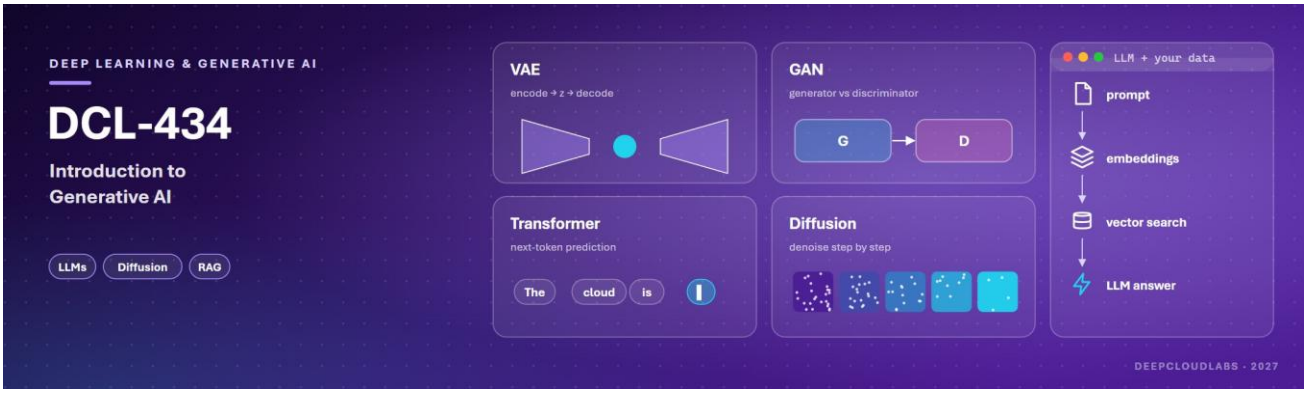
COURSE MODULES

01	Fundamentals of Machine Learning	06	Sequence to Sequence Learning with Neural Networks
02	Introduction to Deep Learning	07	Image Generation with DCGAN
03	Intensive PyTorch Training	08	FaceNet and Metric Learning
04	Object Detection with YOLO	09	Prototypical Networks for Few-Shot Learning
05	Attention Is All You Need	10	Contrastive and Self-Supervised Learning (SimCLR, CLIP)

DCL-434 DEEP LEARNING & GENERATIVE AI

Introduction to Generative AI with Python

How generative models work — and how to build with them in Python.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A practical introduction to generative AI for Python developers. Participants learn how the main families of generative models work — variational autoencoders, generative adversarial networks, transformers and diffusion models — and use them to generate text and images. The course introduces prompt engineering, working with large pre-trained models and LLM APIs, embeddings and retrieval-augmented generation (RAG), and the responsible and secure use of generative AI.

WHAT YOU WILL LEARN

- Explain how VAEs, GANs, transformers and diffusion models generate content
- Engineer effective prompts for text and image models
- Use large pre-trained models and LLM APIs from Python
- Ground model output in your own data with embeddings and RAG
- Apply generative AI responsibly and securely

WHO SHOULD ATTEND

Python developers, data scientists and engineers starting with generative AI.

PREREQUISITES

Python programming; machine learning basics recommended.

TECHNOLOGIES

Python · PyTorch · Transformers · Diffusers · LLM APIs · Vector search

RELATED COURSES

DCL-436 Advanced Generative AI with Python
DCL-610 Generative AI Application Development
DCL-438 Introduction to ChatGPT

07 AI

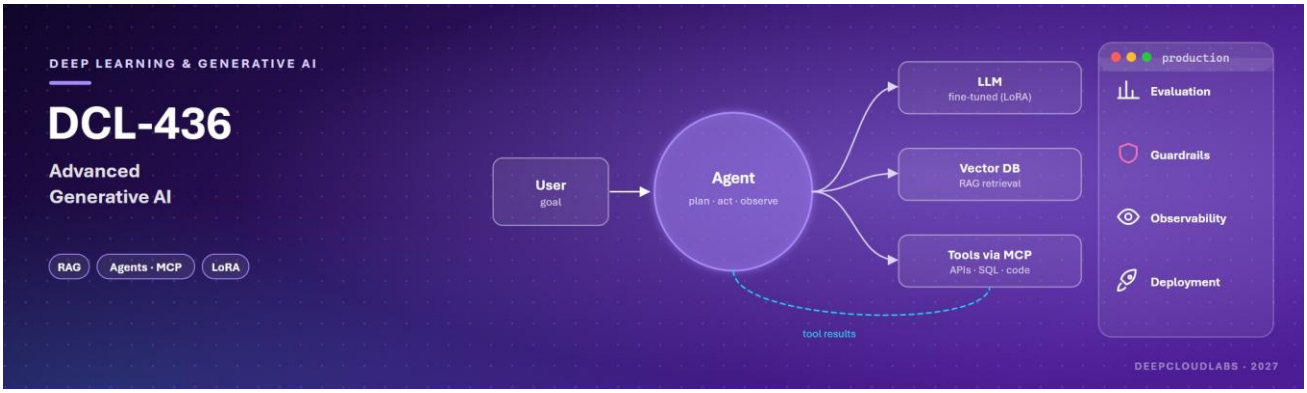
COURSE MODULES

01	Introduction to Generative AI	07	Prompt Engineering
02	Python Review for Generative AI	08	Text Generation
03	Variational Autoencoders (VAEs)	09	Image Generation
04	Generative Adversarial Networks (GANs)	10	Using Large Pre-trained Models and LLM APIs
05	Transformer Neural Networks and Large Language Models	11	Embeddings and Retrieval-Augmented Generation (RAG) Basics
06	Diffusion Models	12	Responsible and Secure Use of Generative AI

DCL-436 DEEP LEARNING & GENERATIVE AI

Advanced Generative AI with Python

Deep generative modeling and production LLM engineering — RAG, agents, fine-tuning and guardrails.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

An advanced, engineering-focused course on building generative AI systems. Part 1 reviews foundations; Part 2 dives into deep generative modeling — VAEs, GANs, autoregressive models, transformers and diffusion models; Part 3 applies them to text, image, audio and multimodal generation; and Part 4 covers what it takes to put large language models into production: parameter-efficient fine-tuning, retrieval-augmented generation with vector databases, AI agents and tool calling with the Model Context Protocol (MCP), evaluation, guardrails and deployment.

WHAT YOU WILL LEARN

- Implement and compare deep generative model families
- Build text, image and audio generation applications
- Fine-tune LLMs efficiently with LoRA/QLoRA
- Engineer RAG pipelines and tool-using AI agents
- Evaluate, guard and deploy LLM applications

WHO SHOULD ATTEND

ML engineers, AI engineers and senior developers building generative AI products.

PREREQUISITES

Python, deep learning fundamentals and DCL-434 or equivalent experience.

TECHNOLOGIES

PyTorch · Transformers · Diffusion · LoRA · Vector databases · MCP · Agents

RELATED COURSES

DCL-612 Retrieval-Augmented Generation for Enterprise Applications

DCL-614 AI Agent and Multi-Agent Systems Engineering

DCL-616 LLMOps: Evaluating and Operating Production AI Systems

DCL-460 Edge Computing: Deployment & Inference with NVIDIA Jetson

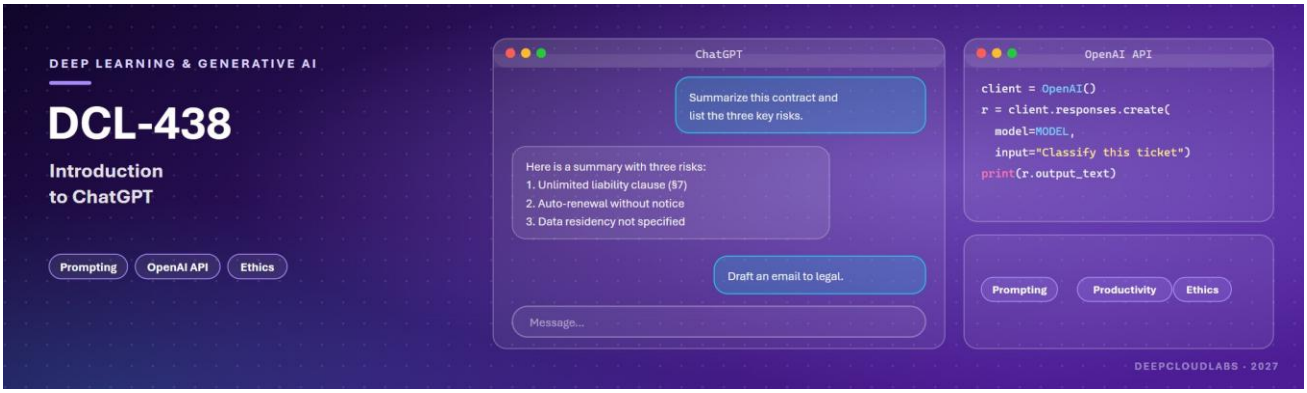
COURSE MODULES

PART 1 · INTRODUCTION AND REVIEW			
01	Introduction to Generative AI	02	Python Review for Generative AI
PART 2 · DEEP GENERATIVE MODELING METHODS			
03	Variational Autoencoders (VAEs)	06	Transformer Neural Networks
04	Generative Adversarial Networks (GANs)	07	Diffusion Models
05	Autoregressive Models		
PART 3 · GENERATIVE AI APPLICATIONS			
08	Prompt Engineering	11	Audio and Music Generation
09	Text Generation with Transformer Neural Networks	12	Using Large Pre-trained Models
10	Image Generation with Multimodal Models		
PART 4 · LLM ENGINEERING IN PRACTICE			
13	Parameter-Efficient Fine-Tuning (LoRA, QLoRA)	15	AI Agents, Tool Calling and the Model Context Protocol (MCP)
14	Retrieval-Augmented Generation and Vector Databases	16	Evaluation, Guardrails and LLM Deployment

DCL-438 DEEP LEARNING & GENERATIVE AI

Introduction to ChatGPT

Use ChatGPT productively and responsibly — and build your first applications with the OpenAI API.



DURATION	LEVEL	DELIVERY	WHERE
2 days	Introductory	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A practical two-day introduction to ChatGPT for professionals and developers. Participants learn how large language models work and where their limits lie, practice prompt engineering, and apply ChatGPT as a personal assistant and for software development, problem solving, creative ideation and research. The final part introduces the OpenAI API and shows how to build applications grounded in your own documents — with attention to generative AI ethics, privacy and responsible use throughout.

WHAT YOU WILL LEARN

- Explain what ChatGPT can and cannot do
- Write effective prompts for everyday and technical tasks
- Use ChatGPT for development, problem solving and research
- Call the OpenAI API from code
- Build a simple application over your own documents

WHO SHOULD ATTEND

Professionals, managers and developers who want to use ChatGPT effectively.

PREREQUISITES

None for Parts 1–2; basic Python for Part 3.

TECHNOLOGIES

ChatGPT · OpenAI API · Prompt engineering · RAG

RELATED COURSES

DCL-434 Introduction to Generative AI with Python
DCL-600 AI-Assisted Software Engineering
DCL-610 Generative AI Application Development

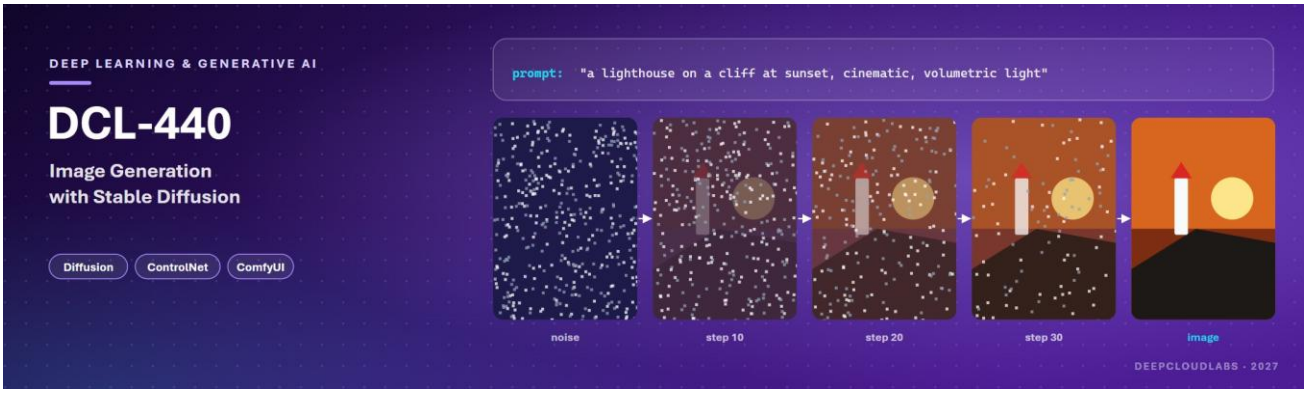
COURSE MODULES

PART 1 · INTRODUCTION			
01	Introduction to Generative AI	03	Generative AI Ethics
02	General Introduction to ChatGPT		
PART 2 · PRODUCTIVITY AND EFFICIENCY			
04	Prompt Engineering	07	Using ChatGPT for Problem Solving
05	Using ChatGPT as a Personal Assistant	08	Using ChatGPT for Creative Ideas
06	Using ChatGPT for Software Development	09	Using ChatGPT for Research
PART 3 · OPENAI API			
10	Introduction to the OpenAI API	11	Building Applications with ChatGPT using Your Own Documents

DCL-440 DEEP LEARNING & GENERATIVE AI

Introduction to Image Generation with Stable Diffusion

From prompt to picture — controlled, high-quality image generation with Stable Diffusion.



DURATION	LEVEL	DELIVERY	WHERE
2 days	Introductory	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A hands-on introduction to image generation with Stable Diffusion. Participants learn how latent diffusion models work, practice prompt engineering, and use web UI and node-based (ComfyUI) workflows for text-to-image and image-to-image generation, controlled generation, inpainting and outpainting, styling and text-to-video — within a clear framework for the ethical use of generative AI.

WHAT YOU WILL LEARN

- Explain how latent diffusion models generate images
- Write prompts that produce consistent, high-quality results
- Control composition and style with ControlNet and LoRA adapters
- Edit images with inpainting and outpainting
- Use generative imagery ethically and responsibly

WHO SHOULD ATTEND

Designers, marketers, creative professionals and developers.

PREREQUISITES

None. Basic computer skills.

TECHNOLOGIES

Stable Diffusion · SDXL · ComfyUI · ControlNet · LoRA

RELATED COURSES

DCL-434 Introduction to Generative AI with Python

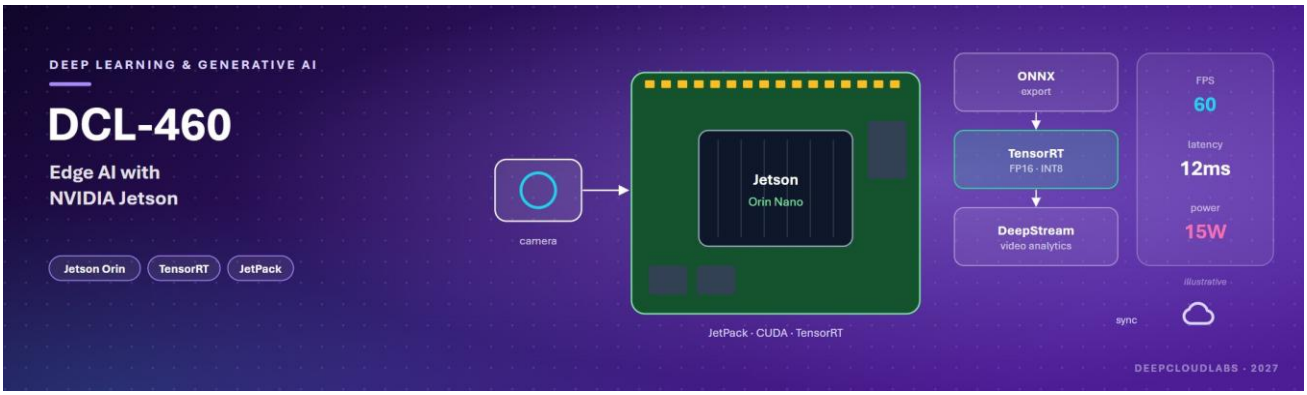
COURSE MODULES

01	Introduction to Generative AI	07	Image-to-Image Generation
02	General Introduction to Stable Diffusion	08	Controlling Image Generation (ControlNet)
03	Generative AI Ethics	09	Image Inpainting and Outpainting
04	Prompt Engineering	10	Styling Images (LoRA)
05	Introduction to Stable Diffusion WebUI and ComfyUI	11	Text-to-Video Generation
06	Text-to-Image Generation	12	Useful Extensions and Custom Nodes

DCL-460 DEEP LEARNING & GENERATIVE AI

Edge Computing: Deployment & Inference with NVIDIA Jetson

Optimize and deploy deep learning models for real-time inference at the edge.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This training introduces the NVIDIA Jetson platform — compact, power-efficient computers capable of running deep neural networks in parallel at the edge. After an introduction to deep learning with a well-known Python framework, participants explore the Jetson Orin Nano Developer Kit through system setup with JetPack, model optimization and deployment with TensorRT, real-time inference and edge computing workloads. Practical computer vision applications and future directions — including generative AI at the edge — prepare participants for real-world problems.

WHAT YOU WILL LEARN

- Set up and manage a Jetson device with JetPack
- Optimize models with TensorRT for low-latency inference
- Deploy computer vision pipelines on Jetson
- Balance accuracy, latency and power at the edge
- Plan edge–cloud architectures for AI workloads

WHO SHOULD ATTEND

Embedded, robotics and ML engineers deploying AI on devices.

PREREQUISITES

Python programming; deep learning basics recommended.

TECHNOLOGIES

NVIDIA Jetson Orin Nano · JetPack · CUDA · TensorRT · ONNX · DeepStream

RELATED COURSES

DCL-422 Deep Learning for Computer Vision

COURSE MODULES

01	Fundamentals of Machine Learning	06	Model Optimization and Deployment with TensorRT
02	Introduction to Deep Learning	07	Jetson Inference
03	Python Review for Deep Learning	08	Edge Computing with Jetson
04	Introduction to NVIDIA Jetson (Orin Nano Developer Kit)	09	Computer Vision Applications on Jetson
05	Jetson System Setup with JetPack	10	Generative AI at the Edge and Future Directions

TRAINING DOMAIN 08 · 9 COURSES · 30 TRAINING DAYS

08 AI-Native Software Engineering

Engineer software with AI — and engineer software built on AI — with verification, security and accountability built in.



Generative AI is changing both how software is built and what software can do. This domain addresses both from an engineering perspective. Track A, engineering with AI, teaches developers and teams to bring AI coding assistants and autonomous coding agents into the software development life cycle: context engineering, specification-driven development, AI-assisted testing and agentic workflows at team scale. Track B, engineering AI applications, teaches how to build, evaluate, operate and secure applications on large language models: LLM APIs and tool calling, enterprise retrieval-augmented generation, agent and multi-agent systems with the Model Context Protocol, LLMOps and AI application security. Every course is anchored in verification — tests, evaluations and reviews decide what ships — and labs are tool-agnostic, running with commercial or self-hosted models.

WHAT'S NEW IN 2027

- New domain in the 2027 catalog (version 3.1): nine courses and 30 training days, plus a one-day executive program (DCL-454).
- Two tracks: engineering with AI (DCL-600–606) and engineering AI applications (DCL-610–618).
- Tool-agnostic labs: Claude Code, OpenAI Codex, GitHub Copilot, Cursor or your approved assistant; commercial or self-hosted models.
- AI-native courses add a course specification page: learning outcomes, practical work, assessment and recommended reading.

PROGRAM OUTLINE

CODE	COURSE	LEVEL	DAYS	PAGE
DCL-600	AI-Assisted Software Engineering	Intermediate	3	72
DCL-602	Specification-Driven Development with AI Agents	Intermediate–Advanced	3	74
DCL-604	Agentic Software Engineering	Advanced	4	76
DCL-606	AI-Assisted Software Testing and Quality Engineering	Intermediate	3	78
DCL-610	Generative AI Application Development	Intermediate	4	80
DCL-612	Retrieval-Augmented Generation for Enterprise Applications	Advanced	3	82
DCL-614	AI Agent and Multi-Agent Systems Engineering	Advanced	4	84
DCL-616	LLMOps: Evaluating and Operating Production AI Systems	Advanced	3	86
DCL-618	Secure and Responsible AI Application Engineering	Advanced	3	88
Total	9 courses		**30**	

Two tracks, one engineering discipline

Track A teaches teams to engineer with AI across the software life cycle; Track B teaches them to engineer applications built on AI. Both tracks share the same rule: tests, evaluations and reviews — not the model — decide what ships.

TRACK	COURSE	LEVEL	DAYS	CORE QUESTION
A · Engineering with AI	DCL-600 AI-Assisted Engineering	Intermediate	3	How do we use AI tools on real code without lowering quality?
	DCL-602 Spec-Driven Development	Intermediate–Advanced	3	How do we decide what agents build — and prove it was built right?
	DCL-606 AI-Assisted Testing & QE	Intermediate	3	How do we get tests that are worth trusting?
	DCL-604 Agentic Engineering	Advanced	4	How do we delegate work to agents safely, at team scale?
B · Engineering AI applications	DCL-610 GenAI App Development	Intermediate	4	How do we build reliable LLM features into applications?
	DCL-612 Enterprise RAG	Advanced	3	How do we ground answers in enterprise knowledge?
	DCL-614 Agent & Multi-Agent Systems	Advanced	4	How do we build agents that act reliably?
	DCL-616 LLMOps	Advanced	3	How do we evaluate and operate AI systems in production?
	DCL-618 Secure & Responsible AI	Advanced	3	How do we keep AI systems secure, private and accountable?

How domain 08 relates to the Generative AI courses in domain 07

ASPECT	DOMAIN 07 · DCL-434, DCL-436, DCL-438	DOMAIN 08 · DCL-610 TO DCL-618
Perspective	Model-centric: how generative models work, are trained and fine-tuned	Engineering-centric: building, evaluating, operating and securing AI applications
Primary audience	Python developers, data scientists, ML engineers	Software engineers, architects, platform, SRE and security engineers
Lab languages	Python	Python or Java (Spring AI, LangChain4j)
RAG, agents, evaluation	Survey modules (DCL-434; DCL-436 Part 4)	Dedicated three- and four-day deep dives

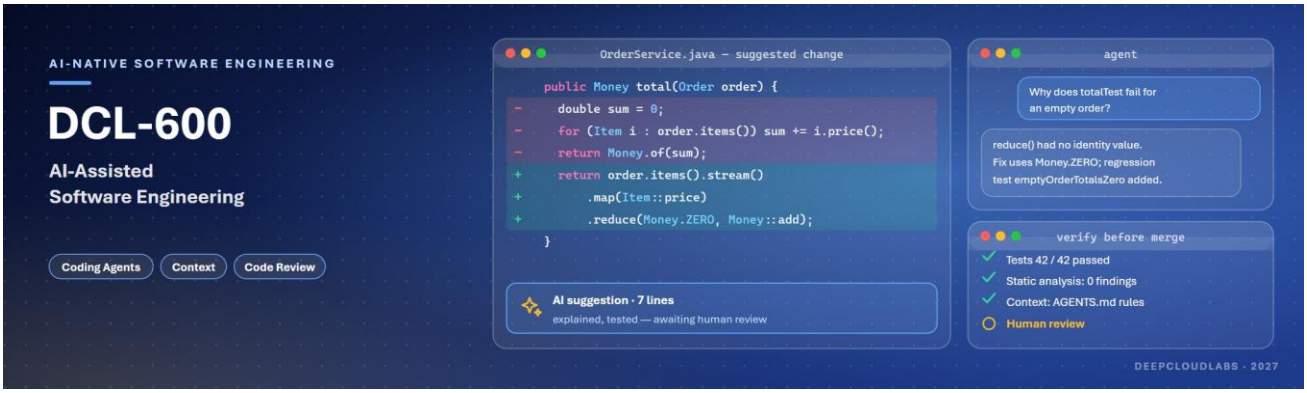
TOOL-AGNOSTIC BY DESIGN

- Labs run on Claude Code, OpenAI Codex, GitHub Copilot, Cursor or your organization's approved assistant — the practices transfer between tools.
- Commercial model APIs or self-hosted open-weight models (Ollama, vLLM) for environments where data must stay on your network.
- Tools and frameworks are used in their current versions at delivery; the curriculum is built on durable practices rather than product releases.

DCL-600 AI-NATIVE SOFTWARE ENGINEERING

AI-Assisted Software Engineering

Write, debug, refactor and review code with AI assistants and agents — without lowering the bar.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A practical course for professional developers on using AI coding assistants and agents in everyday engineering without compromising correctness, security or maintainability. Participants learn how language models generate code and where they fail, how to supply context through repository instructions, documentation and tests, and how to work in small, verifiable increments in the IDE and terminal. Labs on a realistic multi-module code base cover features, debugging, tests, legacy code, refactoring, code review and prototyping. AI output is treated as untrusted until tests, static analysis and review verify it.

WHAT YOU WILL LEARN

- Explain how AI coding tools generate code and where they fail
- Engineer context with repository instructions, docs and tests
- Develop, debug and refactor in small, verifiable increments
- Understand and safely modernize unfamiliar or legacy code
- Review AI-generated changes for correctness, security and licensing

WHO SHOULD ATTEND

Developers, technical leads and DevOps engineers adopting AI coding tools in professional code bases.

PREREQUISITES

One year of professional programming in Java, Python, TypeScript or C#; Git and unit testing. No AI experience required.

TECHNOLOGIES

Claude Code · OpenAI Codex · GitHub Copilot · Cursor · AGENTS.md · Git · JUnit · pytest · Vitest · Semgrep

RELATED COURSES

DCL-602 Specification-Driven Development with AI Agents

DCL-606 AI-Assisted Software Testing and Quality Engineering

DCL-604 Agentic Software Engineering

COURSE MODULES

01	How AI Coding Tools Work: Models, Context, Tools	07	Writing and Improving Unit Tests with AI
02	Completion, Chat and Agentic Editing	08	Understanding and Documenting Legacy Code
03	Context Engineering: AGENTS.md, Docs, Examples	09	Safe Refactoring behind Characterization Tests
04	Prompting Patterns for Code	10	AI-Assisted Code Review and Pull Requests
05	Feature Development in Verifiable Increments	11	Secure Use: Secrets, Data, Licensing, Packages
06	AI-Assisted Debugging: Reproduce, Isolate, Fix	12	Rapid Prototyping and Measuring Impact

DCL-600 COURSE SPECIFICATION · AI-NATIVE SOFTWARE ENGINEERING

AI-Assisted Software Engineering

Learning outcomes, practical components, assessment and recommended reading · 3 days · Intermediate

LEARNING OUTCOMES — ON COMPLETION, PARTICIPANTS ARE ABLE TO

LO1	Configure an AI coding tool for a repository — instruction file, permission profile and project conventions — and justify each choice.	**LO4**	Refactor a legacy module behind characterization tests with no change in observable behavior.
LO2	Implement a specified feature with AI assistance as a series of small commits, each passing the test suite and static analysis.	**LO5**	Find at least four of five seeded correctness, security and maintainability defects in an AI-generated pull request using a review checklist.
LO3	Diagnose a seeded defect by producing a minimal reproduction and a failing regression test before accepting an AI-proposed fix.	**LO6**	Draft a team working agreement for AI-assisted development covering data handling, licensing, review and impact metrics.

PRACTICAL COMPONENTS

#	LABORATORY ACTIVITY	DELIVERABLE AND EVIDENCE
1	Repository setup: instructions, permissions and conventions	AGENTS.md and a reviewed permission profile
2	Feature increment with AI pair programming	Small, tested and explained commits
3	Debugging seeded defects	Reproduction, regression test and fix
4	Legacy comprehension and safe refactoring	Characterization tests, refactored module, design notes
5	AI-assisted code review	Review report on an AI-generated pull request
6	Rapid prototype with a keep-or-discard decision	Working spike and written assessment

CAPSTONE PROJECT

Extend a multi-module reference application (REST API, persistence and web UI) with a new feature end to end — specification, AI-assisted implementation, tests and review notes — and present the verification evidence.

ASSESSMENT

COMPONENT	WEIGHT	EVIDENCE AND CRITERIA
Lab portfolio	40 %	Deliverables complete; tests and static analysis pass; commit history is small-step and explained
Capstone project	45 %	Acceptance tests pass; quality gates met; verification evidence and review notes are convincing
Scenario quiz	15 %	Failure modes, secure use and context-engineering decisions

Weights apply when a program requires graded assessment; otherwise labs and the capstone are reviewed formatively with individual feedback.

EXPECTED COMPETENCIES

- Context engineering for real code bases
- Test-first verification of generated code
- Critical, evidence-based code review
- Secure, policy-compliant use of AI tools
- Clear communication of AI-assisted changes

TOOLS AND TECHNOLOGIES

Assistants & agents Claude Code · OpenAI Codex · GitHub Copilot · Cursor — or your approved assistant

Engineering tooling Git · IntelliJ IDEA / VS Code · JUnit / pytest / Vitest · SonarQube · Semgrep

Lab languages Java (Spring Boot), Python or TypeScript — agreed per group

RECOMMENDED TEXTBOOKS

EMERGING Osmani, A. (2025). *Beyond Vibe Coding: From Coder to AI-Era Developer*. O'Reilly Media.

EMERGING Taulli, T. (2024). *AI-Assisted Programming: Better Planning, Coding, Testing, and Deployment*. O'Reilly Media.

ESTABLISHED Feathers, M. C. (2005). *Working Effectively with Legacy Code*. Prentice Hall.

ESTABLISHED Thomas, D. & Hunt, A. (2020). *The Pragmatic Programmer: Your Journey to Mastery* (20th anniversary ed.). Addison-Wesley.

Established: widely adopted reference works. Emerging: recent publications on fast-moving practice — check for newer editions.

DCL-602 AI-NATIVE SOFTWARE ENGINEERING

Specification-Driven Development with AI Agents

Make the specification the source of truth — and let agents implement against it, verified.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate–Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

When agents generate code in minutes, the bottleneck moves to deciding what to build and proving it was built correctly. Specification-driven development makes requirements, acceptance criteria, architecture constraints and plans versioned, reviewable artifacts that agents implement against and automated checks verify. Participants refine requirements with AI support, write Given–When–Then scenarios and OpenAPI contracts, decompose plans into agent-sized tasks and keep traceability from requirement to commit — placing the workflow in an AI-native life cycle with new roles, AI-assisted Scrum and Kanban, and outcome metrics.

WHAT YOU WILL LEARN

- Explain how AI changes the life cycle — and where judgment stays human
- Turn stakeholder needs into testable acceptance criteria
- Write executable specifications and contracts agents can follow
- Decompose plans into agent-sized tasks with verification steps
- Keep requirements traceable to tests, commits and releases

WHO SHOULD ATTEND

Developers, technical leads, architects, business analysts and product owners.

PREREQUISITES

DCL-600 or hands-on experience with AI coding tools; agile delivery and automated testing.

TECHNOLOGIES

GitHub Spec Kit · Kiro · Claude Code · OpenAI Codex · Cucumber · OpenAPI · JSON Schema · ADRs

RELATED COURSES

DCL-604 Agentic Software Engineering
DCL-606 AI-Assisted Software Testing and Quality Engineering
DCL-352 Domain-Driven Design Essentials

COURSE MODULES

01	The AI-Native Software Development Life Cycle	07	Task Decomposition for Coding Agents
02	From Vibe Coding to Specification-Driven Work	08	The Specify–Plan–Tasks–Implement Loop
03	Requirements Engineering with AI	09	Verification: Acceptance and Contract Tests
04	Executable Specifications (Given–When–Then)	10	Living Specifications and Traceability
05	API Contracts as Specifications (OpenAPI)	11	Human–AI Collaboration in Scrum and Kanban
06	Architecture Constraints and Decision Records	12	Measuring Flow, Quality and Rework

DCL-602 COURSE SPECIFICATION · AI-NATIVE SOFTWARE ENGINEERING

Specification-Driven Development with AI Agents

Learning outcomes, practical components, assessment and recommended reading · 3 days · Intermediate–Advanced

LEARNING OUTCOMES — ON COMPLETION, PARTICIPANTS ARE ABLE TO

LO1	Transform a stakeholder brief into a reviewed specification with testable acceptance criteria, explicit assumptions and open questions.	**LO4**	Direct a coding agent to implement a feature from the specification so that all acceptance and contract tests pass.
LO2	Write executable specifications (Given–When–Then) and an OpenAPI contract that cover every acceptance criterion of a feature.	**LO5**	Apply a change request to specification, tests and code with complete requirement-to-commit traceability.
LO3	Produce an implementation plan and agent-sized tasks, each with a definition of done and a verification command.	**LO6**	Recommend an AI-native delivery workflow and metric set for a given team context, with justification.

PRACTICAL COMPONENTS

#	LABORATORY ACTIVITY	DELIVERABLE AND EVIDENCE
1	AI-supported requirements workshop	Refined requirements and ambiguity log
2	Executable specifications	Gherkin scenarios and OpenAPI contract
3	Plan and task decomposition	Plan, task list and definitions of done
4	Agent implementation against the specification	Feature branch with green acceptance tests
5	Change request and traceability	Updated specification, tests and traceability matrix

CAPSTONE PROJECT

Deliver a small product increment from a stakeholder brief to a released, traced feature with a full specify–plan–tasks–implement–verify workflow; present the specification set, test evidence and a retrospective on human–AI collaboration.

ASSESSMENT

COMPONENT	WEIGHT	EVIDENCE AND CRITERIA
Specification set	30 %	Clarity, testability and completeness of requirements, scenarios and contract
Implementation and verification	40 %	Acceptance and contract tests pass; every task traceable to a requirement
Change-management exercise	15 %	Change propagated consistently to specification, tests and code
Workflow retrospective	15 %	Evidence-based assessment of the process and its metrics

Weights apply when a program requires graded assessment; otherwise labs and the capstone are reviewed formatively with individual feedback.

EXPECTED COMPETENCIES

- Requirements analysis and specification writing
- Planning work for human–agent teams
- Acceptance-test-driven verification
- Traceability and change control
- Facilitating human–AI collaboration in agile teams

TOOLS AND TECHNOLOGIES

Spec-driven tooling GitHub Spec Kit · Kiro specs · agent instruction files

Agents Claude Code · OpenAI Codex · GitHub Copilot cloud agent

Specification formats Markdown specifications · Gherkin (Cucumber) · OpenAPI · JSON Schema · ADRs

RECOMMENDED TEXTBOOKS

ESTABLISHED Adzic, G. (2011). *Specification by Example: How Successful Teams Deliver the Right Software*. Manning Publications.

ESTABLISHED Wiegers, K. & Beatty, J. (2013). *Software Requirements* (3rd ed.). Microsoft Press.

ESTABLISHED Smart, J. F. & Molak, J. (2023). *BDD in Action: Behavior-Driven Development for the Whole Software Lifecycle* (2nd ed.). Manning Publications.

EMERGING Kim, G. & Yegge, S. (2025). *Vibe Coding: Building Production-Grade Software With GenAI, Chat, Agents, and Beyond*. IT Revolution Press.

Established: widely adopted reference works. Emerging: recent publications on fast-moving practice — check for newer editions.

DCL-604 AI-NATIVE SOFTWARE ENGINEERING

Agentic Software Engineering

Delegate real engineering work to autonomous coding agents — safely, in parallel, at team scale.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

Coding agents now plan, edit, run tests and open pull requests on their own. This course teaches how to engineer with them at team scale: how agents work and fail, how to split work into verifiable tasks, and how to run agents in parallel in isolated workspaces. Participants configure agents as code — instructions, commands, skills, subagents, hooks and MCP servers — and run headless agents in CI/CD behind quality gates. Sandboxing, least privilege, prompt injection, cost and auditability run throughout; a team of agents delivers a full-stack capstone.

WHAT YOU WILL LEARN

- Explain agent architectures — planning, tools, memory — and their limits
- Decompose work into agent-sized tasks with verification criteria
- Configure agents as code: commands, skills, subagents, hooks, MCP
- Run parallel agents in isolated workspaces and in CI/CD
- Govern autonomy with sandboxing, least privilege and review gates

WHO SHOULD ATTEND

Senior developers, technical leads, architects and platform engineers.

PREREQUISITES

DCL-600 or equivalent; Git workflows, CI/CD and automated testing (DCL-353 recommended).

TECHNOLOGIES

Claude Code · Claude Agent SDK · OpenAI Codex · GitHub Copilot · MCP · Git worktrees · GitHub Actions · Docker

RELATED COURSES

DCL-614 AI Agent and Multi-Agent Systems Engineering

DCL-618 Secure and Responsible AI Application Engineering

DCL-353 DevOps Engineering

COURSE MODULES

01	From Assistants to Agents: Plan–Act–Observe	08	Headless Agents in CI/CD
02	Context Management: Instructions and Memory	09	Large-Scale Migrations and Refactorings
03	Task Decomposition and Delegation	10	Reviewing Agent Output at Scale
04	Commands, Skills, Subagents and Hooks	11	Security: Sandboxing, Permissions, Injection
05	Extending Agents with MCP Servers	12	Cost, Throughput and Observability
06	Parallel Agents: Worktrees and Sandboxes	13	Capstone: AI-Native Full-Stack Delivery
07	Multi-Agent Workflows: Plan, Build, Review		

DCL-604 COURSE SPECIFICATION · AI-NATIVE SOFTWARE ENGINEERING

Agentic Software Engineering

Learning outcomes, practical components, assessment and recommended reading · 4 days · Advanced

LEARNING OUTCOMES — ON COMPLETION, PARTICIPANTS ARE ABLE TO

LO1	Decompose an epic into agent-executable tasks with acceptance criteria, constraints and verification commands, and justify the decomposition.	**LO4**	Build a CI workflow in which a headless agent proposes a fix for a failing build that a human approves through a quality gate.
LO2	Configure a coding agent for a team repository — instructions, custom commands, a skill, a subagent, a hook and an MCP server — under version control.	**LO5**	Threat-model an agent setup and apply sandboxing, least-privilege and secret-handling controls that mitigate the identified risks.
LO3	Run at least three agents in parallel in isolated workspaces and integrate their results without regressions.	**LO6**	Report the cost, throughput and quality of an agentic workflow from collected telemetry.

PRACTICAL COMPONENTS

#	LABORATORY ACTIVITY	DELIVERABLE AND EVIDENCE
1	Agent anatomy and context budget	Annotated agent session with context analysis
2	Configuration as code	Versioned commands, skill, subagent, hook and MCP server
3	Parallel feature development	Merged branches from three isolated agent workspaces
4	Headless agent in CI	Pipeline that triages and proposes fixes behind an approval gate
5	Large-scale migration	Framework or API upgrade across a multi-module code base
6	Agent threat model and hardening	Threat model, sandbox and permission profile

CAPSTONE PROJECT

Build and deploy a full-stack application (web front end, REST back end, database, CI/CD) with a team of coordinated agents from a written specification; defend the delivery with test evidence, review records, cost data and a security assessment.

ASSESSMENT

COMPONENT	WEIGHT	EVIDENCE AND CRITERIA
Lab portfolio	35 %	Configurations versioned and reproducible; parallel work merged cleanly; CI agent behind gates
Capstone delivery	45 %	Working deployment; acceptance tests pass; review, cost and telemetry evidence
Security assessment	20 %	Threats identified and mitigated with appropriate controls

Weights apply when a program requires graded assessment; otherwise labs and the capstone are reviewed formatively with individual feedback.

EXPECTED COMPETENCIES

- Delegating to and supervising autonomous agents
- Designing agent-ready repositories and workflows
- Integrating agents into CI/CD safely
- Security and governance of AI autonomy
- Measuring and optimizing agentic delivery

TOOLS AND TECHNOLOGIES

Coding agents Claude Code · Claude Agent SDK · OpenAI Codex · GitHub Copilot cloud agent

Integration Model Context Protocol · GitHub Actions / Jenkins · Git worktrees · Docker and dev containers

Capstone stack React or Angular · Spring Boot or FastAPI · PostgreSQL · Kubernetes

RECOMMENDED TEXTBOOKS

EMERGING Kim, G. & Yegge, S. (2025). *Vibe Coding: Building Production-Grade Software With GenAI, Chat, Agents, and Beyond*. IT Revolution Press.

ESTABLISHED Winters, T., Manshreck, T. & Wright, H. (2020). *Software Engineering at Google: Lessons Learned from Programming Over Time*. O'Reilly Media.

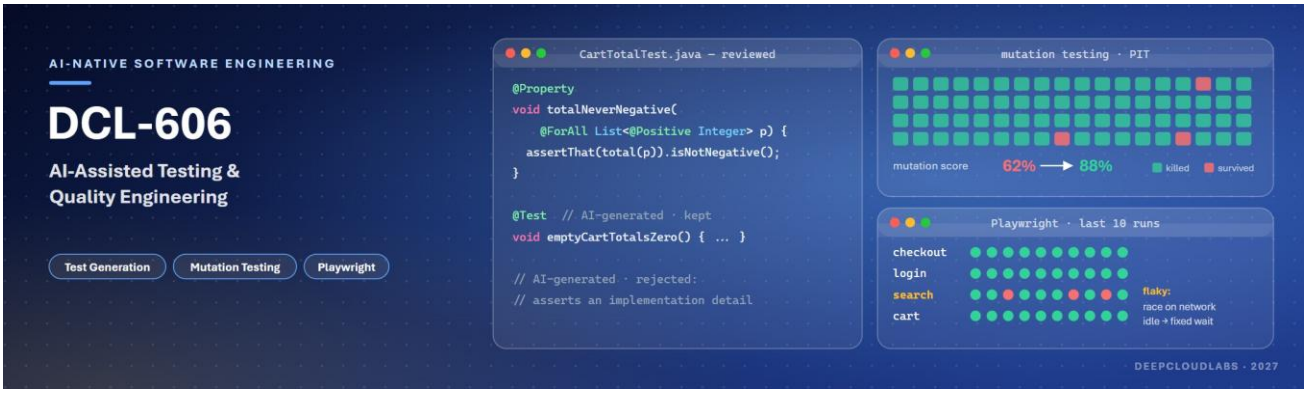
ESTABLISHED Humble, J. & Farley, D. (2011). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley.

Established: widely adopted reference works. Emerging: recent publications on fast-moving practice — check for newer editions.

DCL-606 AI-NATIVE SOFTWARE ENGINEERING

AI-Assisted Software Testing and Quality Engineering

Generate, strengthen and maintain tests with AI — and prove the tests are worth trusting.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

AI produces tests faster than teams can review them; the challenge is getting tests that find defects. This course teaches developers, test engineers and SDETs to use AI across the testing life cycle with a sound strategy in charge. Participants derive tests from requirements and risk, generate and refine unit, API and end-to-end tests, and measure quality with mutation testing rather than coverage alone. Property-based and fuzz testing, synthetic test data, Playwright UI automation, flaky-test diagnosis, security scanning of AI-generated code and testing LLM-powered features complete the course.

WHAT YOU WILL LEARN

- Design risk-based test strategies with a defined role for AI
- Generate and refine unit, API and end-to-end tests with AI
- Measure suite quality with mutation and property-based testing
- Automate UI tests with AI and eliminate flaky tests
- Test AI-generated code for security and LLM features for behavior

WHO SHOULD ATTEND

Developers, test automation engineers, SDETs and QA leads.

PREREQUISITES

Automated testing in Java, Python or TypeScript; DCL-600 recommended, DCL-215 or DCL-351 helpful.

TECHNOLOGIES

JUnit 6 · pytest · Playwright · PIT · Stryker · jqwik · Hypothesis · Testcontainers · Semgrep

RELATED COURSES

DCL-602 Specification-Driven Development with AI Agents

DCL-616 LLMOps: Evaluating and Operating Production AI Systems

DCL-351 Microservices Testing

COURSE MODULES

01	Testing with AI: Oracles, Risks and Strategy	07	Synthetic, Privacy-Safe Test Data
02	Test Design: Partitions, Boundaries, Scenarios	08	AI-Assisted UI Automation with Playwright
03	Generating Unit Tests: Review, Refine, Reject	09	Flaky Tests and Defect Localization
04	API and Integration Tests (Testcontainers)	10	Security Scans of AI-Generated Code
05	Mutation Testing: Quality beyond Coverage	11	Testing LLM-Powered Features
06	Property-Based and Fuzz Testing	12	Quality Gates and Metrics in CI

DCL-606 COURSE SPECIFICATION · AI-NATIVE SOFTWARE ENGINEERING

AI-Assisted Software Testing and Quality Engineering

Learning outcomes, practical components, assessment and recommended reading · 3 days · Intermediate

LEARNING OUTCOMES — ON COMPLETION, PARTICIPANTS ARE ABLE TO

LO1	Derive a risk-based test plan for a feature that states which tests are AI-generated, human-written or exploratory.	**LO4**	Build a maintainable Playwright end-to-end suite with AI assistance and remove the root causes of its flaky tests.
LO2	Raise the mutation score of a module by at least 15 percentage points through AI-assisted test generation and review.	**LO5**	Detect and remediate insecure patterns, hard-coded secrets and hallucinated dependencies in AI-generated code with automated scans.
LO3	Write property-based tests that reveal a seeded defect missed by the existing example-based tests.	**LO6**	Design an evaluation-based test with explicit pass criteria for an LLM-powered feature.

PRACTICAL COMPONENTS

#	LABORATORY ACTIVITY	DELIVERABLE AND EVIDENCE
1	Risk-based test design with AI	Test plan and generated test cases
2	Unit-test generation and review	Reviewed suite with a rejected-test log
3	Mutation testing	Mutation reports before and after improvement
4	Property-based testing	Properties and counterexamples found
5	UI automation and flakiness	Playwright suite and flaky-test analysis
6	Securing AI-generated code	SAST, secret and dependency scans with fixes

CAPSTONE PROJECT

Take over an under-tested service, define its test strategy and raise its verified quality — mutation score, end-to-end coverage of critical paths and security findings — presenting before-and-after evidence and a maintenance plan for the suite.

ASSESSMENT

COMPONENT	WEIGHT	EVIDENCE AND CRITERIA
Lab portfolio	40 %	Tests run green in CI, are reviewed and justified, and show measurable quality gains
Capstone	45 %	Mutation score, defect detection and security results against agreed targets
Test-strategy defense	15 %	Rationale for the human–AI split, risks and metrics

Weights apply when a program requires graded assessment; otherwise labs and the capstone are reviewed formatively with individual feedback.

EXPECTED COMPETENCIES

- Risk-based test strategy
- Critical review of generated tests
- Test-suite quality measurement
- Reliable UI and API automation
- Security-aware quality engineering

TOOLS AND TECHNOLOGIES

Test frameworks JUnit 6 · pytest · Vitest · Playwright · Testcontainers

Test quality PIT (Java) · Stryker (JavaScript, TypeScript, C#) · mutmut (Python) · jqwik · Hypothesis

Security & CI Semgrep · CodeQL · OWASP Dependency-Check · Gitleaks · GitHub Actions / Jenkins

RECOMMENDED TEXTBOOKS

EMERGING Winteringham, M. (2024). *Software Testing with Generative AI*. Manning Publications.

ESTABLISHED Aniche, M. (2022). *Effective Software Testing: A Developer's Guide*. Manning Publications.

ESTABLISHED Khorikov, V. (2020). *Unit Testing Principles, Practices, and Patterns*. Manning Publications.

Established: widely adopted reference works. Emerging: recent publications on fast-moving practice — check for newer editions.

DCL-610 AI-NATIVE SOFTWARE ENGINEERING

Generative AI Application Development

Build reliable LLM features into real applications — structured outputs, tools and evaluation.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

A hands-on course for software engineers who integrate large language models into products and enterprise systems. Participants build features with provider APIs: prompt templates, structured outputs validated against JSON Schema, tool calling, streaming, conversation state, multimodal input and embeddings. They connect applications to tools and data with the Model Context Protocol (MCP), compare direct APIs with LangChain, Spring AI and LangChain4j, and treat production concerns from day one — retries, latency and cost budgets, caching, evaluation of non-deterministic output and baseline security. Labs are offered in Python or Java.

WHAT YOU WILL LEARN

- Explain LLM capabilities, limits and cost drivers in engineering terms
- Design prompts that yield reliable, schema-validated output
- Implement tool calling, streaming and conversation state
- Integrate tools and data through MCP; choose APIs or frameworks
- Test and harden LLM features for cost, latency and security

WHO SHOULD ATTEND

Back-end, full-stack and integration developers adding generative AI features to applications.

PREREQUISITES

Python or Java programming, REST APIs and JSON. No machine learning background required.

TECHNOLOGIES

OpenAI API · Claude API · Gemini API · Ollama · MCP · LangChain · Spring AI 2 · LangChain4j · Pydantic

RELATED COURSES

DCL-612 Retrieval-Augmented Generation for Enterprise Applications

DCL-614 AI Agent and Multi-Agent Systems Engineering

DCL-616 LLMOps: Evaluating and Operating Production AI Systems

COURSE MODULES

01	LLMs for Engineers: Tokens, Context, Cost	08	Embeddings and Semantic Search
02	Provider APIs and Open-Weight Models	09	Connecting Tools and Data with MCP
03	Prompt Design, Templates and Versioning	10	Frameworks: LangChain, Spring AI, LangChain4j
04	Structured Outputs and Schema Validation	11	Reliability: Retries, Fallbacks, Caching
05	Function and Tool Calling	12	Testing and Evaluating LLM Features
06	Streaming, Conversation State and Memory	13	Security Basics for LLM Features
07	Multimodal Inputs: Documents, Images, Audio	14	Shipping a Feature: Configuration and Cost

DCL-610 COURSE SPECIFICATION · AI-NATIVE SOFTWARE ENGINEERING

Generative AI Application Development

Learning outcomes, practical components, assessment and recommended reading · 4 days · Intermediate

LEARNING OUTCOMES — ON COMPLETION, PARTICIPANTS ARE ABLE TO

LO1	Implement an LLM-backed endpoint whose responses are validated against a JSON Schema, with retry and graceful degradation for invalid output.	**LO4**	Create an evaluation set and automated tests that detect regressions when prompts or models change.
LO2	Build a tool-calling workflow that invokes at least two enterprise APIs with argument validation and authorization checks.	**LO5**	Reduce the per-request cost and latency of a feature through caching, model selection and prompt optimization, documenting the trade-offs.
LO3	Expose an internal capability as an MCP server and consume it from an LLM application.	**LO6**	Identify and mitigate prompt-injection and data-leakage risks in an LLM feature.

PRACTICAL COMPONENTS

#	LABORATORY ACTIVITY	DELIVERABLE AND EVIDENCE
1	First LLM service	Endpoint with prompt templates and externalized configuration
2	Structured extraction	Schema-validated extraction with retries
3	Tool-calling assistant	Assistant that calls enterprise APIs safely
4	MCP integration	MCP server and client integration
5	Evaluation and regression tests	Evaluation set and CI test suite
6	Cost and latency tuning	Measured before-and-after report

CAPSTONE PROJECT

Design and build an LLM-powered feature for a realistic enterprise scenario — for example a support-ticket triage and reply assistant — with structured outputs, tool calling, an evaluation suite, a cost budget and a security review.

ASSESSMENT

COMPONENT	WEIGHT	EVIDENCE AND CRITERIA
Lab portfolio	40 %	Working services with schema validation, error handling and tests
Capstone	45 %	Evaluation results meet thresholds; cost and latency within budget; security review done
Design review	15 %	Justified choice of model, framework, prompts and architecture

Weights apply when a program requires graded assessment; otherwise labs and the capstone are reviewed formatively with individual feedback.

EXPECTED COMPETENCIES

- Engineering LLM features into services
- Schema-first, testable prompt design
- API and tool integration
- Cost- and latency-aware design
- Baseline AI application security

TOOLS AND TECHNOLOGIES

Model APIs OpenAI API · Claude API · Gemini API · open-weight models through Ollama or vLLM

Frameworks LangChain · Spring AI 2 · LangChain4j · Pydantic · MCP SDKs

Lab stacks Python (FastAPI) or Java (Spring Boot 4) — agreed per group

RECOMMENDED TEXTBOOKS

EMERGING Huyen, C. (2025). *AI Engineering: Building Applications with Foundation Models*. O'Reilly Media.

EMERGING Berryman, J. & Ziegler, A. (2025). *Prompt Engineering for LLMs: The Art and Science of Building Large Language Model–Based Applications*. O'Reilly Media.

EMERGING Walls, C. (2025). *Spring AI in Action*. Manning Publications.

ESTABLISHED Nygard, M. T. (2018). *Release It! Design and Deploy Production-Ready Software* (2nd ed.). Pragmatic Bookshelf.

Established: widely adopted reference works. Emerging: recent publications on fast-moving practice — check for newer editions.

DCL-612 AI-NATIVE SOFTWARE ENGINEERING

Retrieval-Augmented Generation for Enterprise Applications

Ground LLM answers in enterprise knowledge — with measurable retrieval quality and access control.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

Retrieval-augmented generation (RAG) connects language models to an organization's own documents and data, and its quality depends more on retrieval engineering than on the model. The course covers the full pipeline: document parsing, chunking and metadata, embeddings, vector indexes, hybrid lexical–semantic search, re-ranking, query transformation and generation with citations. Participants add enterprise capabilities — permission-aware retrieval, incremental indexing, multilingual and structured data, knowledge graphs (GraphRAG) — then evaluate retrieval and answer quality, diagnose failures, and plan agentic retrieval and operating costs at scale.

WHAT YOU WILL LEARN

- Design end-to-end RAG architectures for enterprise sources
- Engineer ingestion, chunking and indexing for retrieval quality
- Combine lexical, semantic and graph retrieval with re-ranking
- Enforce document-level access control in retrieval
- Evaluate retrieval and answers and diagnose failure modes

WHO SHOULD ATTEND

AI engineers, back-end developers, search engineers and solution architects.

PREREQUISITES

DCL-610 or equivalent experience calling LLM APIs; Python or Java programming; basic SQL.

TECHNOLOGIES

pgvector · OpenSearch · Elasticsearch · Qdrant · Neo4j · LlamaIndex · LangChain · Spring AI 2 · Ragas

RELATED COURSES

DCL-614 AI Agent and Multi-Agent Systems Engineering

DCL-616 LLMOps: Evaluating and Operating Production AI Systems

DCL-618 Secure and Responsible AI Application Engineering

COURSE MODULES

01	RAG Architectures — and When Not to Use RAG	08	Grounded Answers: Citations and Abstention
02	Parsing PDF, Office, HTML and Tables	09	Permission-Aware Retrieval and Governance
03	Chunking Strategies and Metadata	10	Knowledge Graphs and GraphRAG
04	Embedding Models and Vector Representations	11	Agentic and Multi-Step Retrieval
05	Vector Indexes: HNSW, pgvector, OpenSearch	12	Evaluation: Recall@k, Faithfulness, Relevance
06	Hybrid Search: BM25, Dense and Rank Fusion	13	Operating RAG: Freshness, Scale and Cost
07	Re-Ranking and Query Transformation		

DCL-612 COURSE SPECIFICATION · AI-NATIVE SOFTWARE ENGINEERING

Retrieval-Augmented Generation for Enterprise Applications

Learning outcomes, practical components, assessment and recommended reading · 3 days · Advanced

LEARNING OUTCOMES — ON COMPLETION, PARTICIPANTS ARE ABLE TO

LO1	Design a RAG architecture for a given enterprise corpus and justify the chunking, embedding, index and retrieval choices.	**LO4**	Implement document-level access control so that users never retrieve content they are not authorized to see, verified by automated tests.
LO2	Build an ingestion pipeline that parses heterogeneous documents, preserves structure and metadata, and supports incremental updates.	**LO5**	Produce grounded answers with verifiable citations and correct abstention when evidence is missing.
LO3	Improve recall@10 on a labeled query set with hybrid search and re-ranking, demonstrating the gain over a dense-only baseline.	**LO6**	Evaluate a RAG system with retrieval and generation metrics and trace failing queries to their root cause.

PRACTICAL COMPONENTS

#	LABORATORY ACTIVITY	DELIVERABLE AND EVIDENCE
1	Ingestion and chunking	Pipeline with a chunk-quality report
2	Vector and hybrid indexes	Dense, lexical and hybrid indexes compared
3	Re-ranking and query transformation	Measured recall and precision gains
4	Permission-aware retrieval	Access-filtered retrieval with security tests
5	Grounded answers with citations	Answer service with automated citation checks
6	RAG evaluation and diagnosis	Evaluation report and failure analysis

CAPSTONE PROJECT

Build a knowledge assistant over a realistic, access-controlled document collection (policies, manuals, tickets) that meets agreed retrieval and faithfulness targets; present the evaluation evidence and an operating-cost estimate.

ASSESSMENT

COMPONENT	WEIGHT	EVIDENCE AND CRITERIA
Lab portfolio	35 %	Reproducible pipelines; metrics measured against baselines
Capstone	50 %	Retrieval and faithfulness targets met; access control verified; cost estimated
Architecture review	15 %	Justified design decisions and identified risks

Weights apply when a program requires graded assessment; otherwise labs and the capstone are reviewed formatively with individual feedback.

EXPECTED COMPETENCIES

- Information-retrieval engineering
- Pipeline design for unstructured content
- Security by design for knowledge access
- Metric-driven quality improvement
- Cost-aware search architecture

TOOLS AND TECHNOLOGIES

Search & vector stores pgvector · OpenSearch · Elasticsearch · Qdrant · Neo4j

Frameworks & parsing LlamaIndex · LangChain · Spring AI 2 · Apache Tika · Docling

Evaluation Ragas · labeled query sets · model-graded metrics calibrated by human review

RECOMMENDED TEXTBOOKS

ESTABLISHED Manning, C. D., Raghavan, P. & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

ESTABLISHED Turnbull, D. & Berryman, J. (2016). *Relevant Search: With Applications for Solr and Elasticsearch*. Manning Publications.

EMERGING Grainger, T., Turnbull, D. & Irwin, M. (2024). *AI-Powered Search*. Manning Publications.

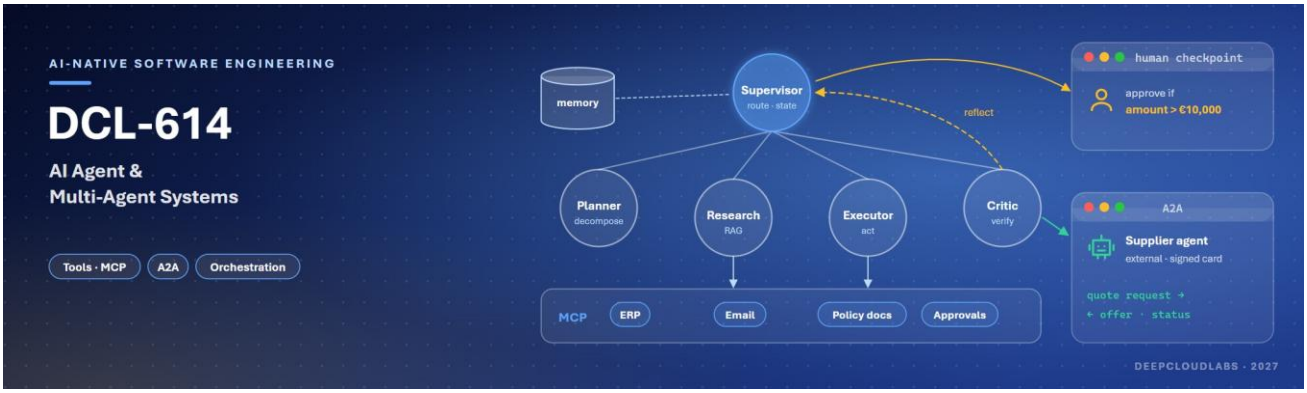
EMERGING Kimothi, A. (2025). *A Simple Guide to Retrieval Augmented Generation*. Manning Publications.

Established: widely adopted reference works. Emerging: recent publications on fast-moving practice — check for newer editions.

DCL-614 AI-NATIVE SOFTWARE ENGINEERING

AI Agent and Multi-Agent Systems Engineering

Design, build and operate tool-using AI agents and multi-agent systems that hold up in production.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

AI agents combine a language model with tools, memory and a control loop to pursue goals over many steps. This course teaches how to engineer them as dependable software: single tool-using agents, planner–executor, supervisor and graph-based workflows, with well-designed tools and the Model Context Protocol (MCP) for integration. Participants build memory, planning and reflection, human-in-the-loop checkpoints and inter-agent communication with the Agent2Agent (A2A) protocol, and study multi-agent coordination and failure modes — with determinism, error recovery, guardrails, trajectory evaluation, observability and cost addressed throughout.

WHAT YOU WILL LEARN

- Choose agent architectures — or a deterministic workflow instead
- Implement tool-using agents with validation and error recovery
- Integrate tools, data and other agents through MCP and A2A
- Engineer memory, planning, reflection and human oversight
- Evaluate, observe, secure and operate multi-agent systems

WHO SHOULD ATTEND

AI engineers, senior developers and architects building agentic applications.

PREREQUISITES

DCL-610 or equivalent; strong Python or Java skills. DCL-612 recommended.

TECHNOLOGIES

MCP · A2A · LangGraph · CrewAI · OpenAI Agents SDK · Claude Agent SDK · Google ADK · Microsoft Agent Framework · Spring AI 2

RELATED COURSES

DCL-616 LLMOps: Evaluating and Operating Production AI Systems

DCL-618 Secure and Responsible AI Application Engineering

DCL-604 Agentic Software Engineering

COURSE MODULES

01	Agents versus Workflows: Trade-Offs	08	Human-in-the-Loop Approvals and Escalation
02	The Agent Loop: Reason, Act, Observe	09	Multi-Agent Patterns: Supervisor, Swarm, Debate
03	Tool Design, Validation and Error Handling	10	Interoperability with the A2A Protocol
04	Building and Securing MCP Servers	11	Agent Frameworks and SDKs Compared
05	Stateful Graph Orchestration (LangGraph)	12	Evaluating Agents: Success and Trajectories
06	Memory: Working, Episodic, Semantic	13	Guardrails, Permissions and Safe Autonomy
07	Planning, Reflection and Self-Correction	14	Operating Agents: Durability, Tracing, Cost

DCL-614 COURSE SPECIFICATION · AI-NATIVE SOFTWARE ENGINEERING

AI Agent and Multi-Agent Systems Engineering

Learning outcomes, practical components, assessment and recommended reading · 4 days · Advanced

LEARNING OUTCOMES — ON COMPLETION, PARTICIPANTS ARE ABLE TO

LO1	Select an agentic or deterministic-workflow architecture for a business process and document the trade-offs.	**LO4**	Implement a multi-agent workflow with shared state, human approval checkpoints and inter-agent communication.
LO2	Implement an agent with at least three well-designed tools, input validation and recovery from tool failures.	**LO5**	Evaluate agent behavior on a task suite by success rate, trajectory quality and cost, and improve it measurably.
LO3	Build an MCP server that exposes an enterprise system to agents with authentication and least-privilege scopes.	**LO6**	Instrument an agent system with traces and guardrails suitable for production operation.

PRACTICAL COMPONENTS

#	LABORATORY ACTIVITY	DELIVERABLE AND EVIDENCE
1	Tool-using agent from first principles	Agent loop with validated tools
2	MCP server engineering	Secured MCP server with tests
3	Graph-based orchestration	Stateful workflow with checkpoints
4	Memory and human-in-the-loop	Agent with persistent memory and approvals
5	Multi-agent collaboration and A2A	Supervisor–worker system with an A2A endpoint
6	Agent evaluation and tracing	Task-suite results and trace analysis

CAPSTONE PROJECT

Engineer a multi-agent system that automates a realistic enterprise process — for example purchase-request handling across email, ERP and policy documents — with MCP-integrated tools, human approvals, evaluation results, guardrails and an operations runbook.

ASSESSMENT

COMPONENT	WEIGHT	EVIDENCE AND CRITERIA
Lab portfolio	35 %	Agents and servers work under test; failures handled; code reviewed
Capstone	50 %	Task-success and cost targets met; approvals, guardrails and observability in place
Architecture defense	15 %	Justified agent-versus-workflow decisions and risk analysis

Weights apply when a program requires graded assessment; otherwise labs and the capstone are reviewed formatively with individual feedback.

EXPECTED COMPETENCIES

- Agent architecture and workflow design
- Tool and protocol engineering (MCP, A2A)
- Designing for safe autonomy and human oversight
- Evaluating non-deterministic systems
- Operating agentic services in production

TOOLS AND TECHNOLOGIES

Protocols Model Context Protocol (MCP) · Agent2Agent (A2A)

Frameworks & SDKs LangGraph · CrewAI · OpenAI Agents SDK · Claude Agent SDK · Google ADK · Microsoft Agent Framework · Spring AI 2

Operations LangSmith or Langfuse tracing · OpenTelemetry · Docker · Kubernetes

RECOMMENDED TEXTBOOKS

ESTABLISHED Russell, S. & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

ESTABLISHED Wooldridge, M. (2009). *An Introduction to MultiAgent Systems* (2nd ed.). John Wiley & Sons.

EMERGING Lanham, M. (2026). *AI Agents in Action: Intelligent Workflows with LLMs, MCP, A2A, and More* (2nd ed.). Manning Publications.

EMERGING Albada, M. (2025). *Building Applications with AI Agents: Designing and Implementing Multiagent Systems*. O'Reilly Media.

Established: widely adopted reference works. Emerging: recent publications on fast-moving practice — check for newer editions.

DCL-616 AI-NATIVE SOFTWARE ENGINEERING

LLMOps: Evaluating and Operating Production AI Systems

Ship LLM applications with confidence — evaluation gates, observability and safe releases.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

Moving an LLM application from a promising demo to a dependable service takes engineering discipline across its life cycle. This course applies MLOps and site-reliability practice to systems built on language models. Participants design evaluations — curated datasets, reference-based and model-graded metrics, human review and online feedback — and run them as regression gates in CI/CD. They trace applications with OpenTelemetry, track quality, latency, tokens and cost, operate model gateways with routing, fallbacks and caching, and manage versioning, canary releases, self-hosted inference with vLLM, drift, incidents and governance records.

WHAT YOU WILL LEARN

- Design layered evaluation strategies for LLM applications and agents
- Automate evaluations as regression gates in CI/CD pipelines
- Observe LLM systems for quality, latency, cost and errors
- Operate model gateways, serving and safe release processes
- Manage incidents, drift, model upgrades and governance evidence

WHO SHOULD ATTEND

AI engineers, MLOps and platform engineers, SREs and technical leads running LLM applications.

PREREQUISITES

DCL-610 or equivalent; experience with CI/CD, containers and monitoring (DCL-353 recommended).

TECHNOLOGIES

OpenTelemetry · Langfuse · LangSmith · MLflow · promptfoo · LiteLLM · vLLM · Kubernetes · Prometheus

RELATED COURSES

DCL-618 Secure and Responsible AI Application Engineering

DCL-614 AI Agent and Multi-Agent Systems Engineering

DCL-353 DevOps Engineering

COURSE MODULES

01	From MLOps to LLMOps: Life Cycle and Roles	08	Model Gateways: Routing, Fallbacks, Caching
02	Evaluation Design: Datasets, Metrics, Rubrics	09	Self-Hosted Inference with vLLM
03	LLM-as-a-Judge and Human Review	10	Safe Releases: Shadow, Canary, A/B Tests
04	Evaluation Gates in CI/CD	11	Feedback Loops and Drift
05	Prompt, Model and Configuration Versioning	12	Incident Management for AI Systems
06	Tracing with OpenTelemetry GenAI Conventions	13	Governance Evidence and Audit Trails
07	Monitoring Quality, Latency, Tokens and Cost		

DCL-616 COURSE SPECIFICATION · AI-NATIVE SOFTWARE ENGINEERING

LLMOps: Evaluating and Operating Production AI Systems

Learning outcomes, practical components, assessment and recommended reading · 3 days · Advanced

LEARNING OUTCOMES — ON COMPLETION, PARTICIPANTS ARE ABLE TO

LO1	Build an evaluation dataset and metric suite for an LLM application and calibrate a model-graded metric against human judgments.	**LO4**	Configure a model gateway with routing, fallback, rate limiting and caching, and demonstrate its behavior under provider failure.
LO2	Implement a CI/CD pipeline that blocks a release when evaluation scores fall below defined thresholds.	**LO5**	Plan and execute a model upgrade with shadow evaluation, a canary release and a documented rollback criterion.
LO3	Instrument an LLM application end to end with traces, metrics and cost attribution, and build an operational dashboard.	**LO6**	Run an incident exercise for an AI-specific failure and write a blameless post-incident review.

PRACTICAL COMPONENTS

#	LABORATORY ACTIVITY	DELIVERABLE AND EVIDENCE
1	Evaluation suite and judge calibration	Dataset, metrics and agreement analysis
2	Evaluation gate in CI	Pipeline that blocks a regressed release
3	Tracing and dashboards	Instrumented service and operational dashboard
4	Gateway and resilience	Gateway configuration and failover test
5	Self-hosted serving	vLLM deployment with load-test results
6	Model upgrade and incident drill	Canary plan, rollback test and post-incident review

CAPSTONE PROJECT

Take a working LLM application to production readiness — evaluation gates, observability, gateway resilience, a release plan for a model change and an operations runbook — and demonstrate it in a production-readiness review.

ASSESSMENT

COMPONENT	WEIGHT	EVIDENCE AND CRITERIA
Lab portfolio	40 %	Pipelines, dashboards and gateway configurations work and are reproducible
Production-readiness review	45 %	Evaluation gates, observability, release plan and runbook meet the checklist
Incident exercise	15 %	Quality of diagnosis and of the post-incident review

Weights apply when a program requires graded assessment; otherwise labs and the capstone are reviewed formatively with individual feedback.

EXPECTED COMPETENCIES

- Evaluation-driven development
- Observability and SLOs for AI services
- Release engineering for models and prompts
- Cost and capacity management
- Operational governance and incident response

TOOLS AND TECHNOLOGIES

Evaluation promptfoo · DeepEval · Ragas · MLflow evaluation · LangSmith · human review workflows

Observability OpenTelemetry · Langfuse · Arize Phoenix · Prometheus · Grafana

Serving & gateways vLLM · LiteLLM · Kubernetes · managed model endpoints

RECOMMENDED TEXTBOOKS

ESTABLISHED Huyen, C. (2022). *Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications*. O'Reilly Media.

ESTABLISHED Beyer, B., Jones, C., Petoff, J. & Murphy, N. R. (Eds.) (2016). *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media.

EMERGING Huyen, C. (2025). *AI Engineering: Building Applications with Foundation Models*. O'Reilly Media.

EMERGING Aryan, A. (2025). *LLMOps: Managing Large Language Models in Production*. O'Reilly Media.

Established: widely adopted reference works. Emerging: recent publications on fast-moving practice — check for newer editions.

DCL-618 AI-NATIVE SOFTWARE ENGINEERING

Secure and Responsible AI Application Engineering

Defend LLM applications and agents against prompt injection, data leakage and misuse.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

LLM applications open attack surfaces that classic application security does not cover: instructions hidden in documents, over-privileged tools and agents, sensitive-data leakage, poisoned retrieval content and unsafe output handling. Participants threat-model AI systems against the OWASP Top 10 for LLM and agentic applications, build defense in depth against prompt injection — privilege separation, tool permissions, content isolation, output validation, human approval — and secure RAG pipelines, agents and MCP integrations. Labs include automated and manual red teaming. Responsible-AI engineering and governance with NIST AI RMF, ISO/IEC 42001 and the EU AI Act close the course.

WHAT YOU WILL LEARN

- Threat-model LLM applications, RAG pipelines and agents
- Defend against prompt injection and data exfiltration in depth
- Secure tools, agents and MCP servers with least privilege
- Red-team AI applications with automated and manual techniques
- Engineer privacy, transparency and oversight for governance

WHO SHOULD ATTEND

Application security and AI engineers, senior developers, architects and risk specialists.

PREREQUISITES

DCL-610 or equivalent; basic application-security knowledge (OWASP Top 10).

TECHNOLOGIES

OWASP LLM Top 10 · MITRE ATLAS · garak · PyRIT · promptfoo · NeMo Guardrails · Llama Guard · NIST AI RMF · ISO/IEC 42001

RELATED COURSES

DCL-616 LLMOps: Evaluating and Operating Production AI Systems

DCL-614 AI Agent and Multi-Agent Systems Engineering

DCL-604 Agentic Software Engineering

COURSE MODULES

01	AI Attack Surface: OWASP LLM and Agentic Top 10	08	AI Supply Chain: Models, Data, Packages
02	Threat Modeling AI (STRIDE, MITRE ATLAS)	09	Guardrails and Content-Safety Layers
03	Prompt Injection: Attacks and Defense in Depth	10	Red Teaming AI Applications
04	Sensitive Information Disclosure	11	Responsible AI: Privacy, Fairness, Oversight
05	Securing RAG: Poisoning, Access, Provenance	12	Governance: NIST AI RMF, ISO/IEC 42001, EU AI Act
06	Securing Agents, Tools and MCP Servers	13	Monitoring, Abuse Detection, Incident Response
07	Output Handling and Code-Execution Risks		

DCL-618 COURSE SPECIFICATION · AI-NATIVE SOFTWARE ENGINEERING

Secure and Responsible AI Application Engineering

Learning outcomes, practical components, assessment and recommended reading · 3 days · Advanced

LEARNING OUTCOMES — ON COMPLETION, PARTICIPANTS ARE ABLE TO

LO1	Produce a threat model for an LLM application with agents and RAG, prioritizing risks against the OWASP Top 10 for LLM and agentic applications.	**LO4**	Run an automated and manual red-team campaign and report findings with severity, reproduction steps and remediation.
LO2	Demonstrate an indirect prompt-injection attack on a vulnerable application and verify that layered mitigations prevent it.	**LO5**	Implement data-protection controls that keep personal data out of prompts, logs and responses.
LO3	Restrict an agent's tools and MCP servers to least privilege, with approval gates for high-impact actions verified by tests.	**LO6**	Map an AI system to the NIST AI RMF functions and identify the applicable EU AI Act obligations and evidence.

PRACTICAL COMPONENTS

#	LABORATORY ACTIVITY	DELIVERABLE AND EVIDENCE
1	AI threat modeling	Threat model with prioritized risks
2	Prompt-injection attack and defense	Working exploit and verified layered mitigations
3	Securing RAG and agents	Access-controlled retrieval and least-privilege tool profile
4	Guardrails and output handling	Input and output guardrails with tests
5	Red-team campaign	Scanner and manual findings report
6	Governance mapping	Risk register and control mapping

CAPSTONE PROJECT

Assess and harden a deliberately vulnerable AI assistant with RAG and tool access: threat model, attack demonstrations, implemented mitigations, red-team retest and a governance dossier suitable for an internal risk committee.

ASSESSMENT

COMPONENT	WEIGHT	EVIDENCE AND CRITERIA
Lab portfolio	35 %	Attacks reproduced; mitigations implemented and verified by tests
Capstone security assessment	50 %	Residual-risk reduction demonstrated; findings and dossier complete
Governance exercise	15 %	Correct mapping of obligations, controls and evidence

Weights apply when a program requires graded assessment; otherwise labs and the capstone are reviewed formatively with individual feedback.

EXPECTED COMPETENCIES

- AI threat modeling and risk assessment
- Secure design of agentic and RAG systems
- Adversarial testing of AI applications
- Privacy and responsible-AI engineering
- Governance and compliance evidence

TOOLS AND TECHNOLOGIES

Testing & red teaming garak · PyRIT · promptfoo red teaming · Burp Suite

Guardrails NeMo Guardrails · Llama Guard and Prompt Guard · provider moderation APIs

Threat modeling STRIDE · MITRE ATLAS · OWASP Threat Dragon

RECOMMENDED TEXTBOOKS

EMERGING Wilson, S. (2024). *The Developer's Playbook for Large Language Model Security: Building Secure AI Applications*. O'Reilly Media.

ESTABLISHED Shostack, A. (2014). *Threat Modeling: Designing for Security*. Wiley.

EMERGING Lu, Q., Zhu, L., Whittle, J. & Xu, X. (2024). *Responsible AI: Best Practices for Creating Trustworthy AI Systems*. Addison-Wesley.

EMERGING Sotiropoulos, J. (2024). *Adversarial AI Attacks, Mitigations, and Defense Strategies*. Packt Publishing.

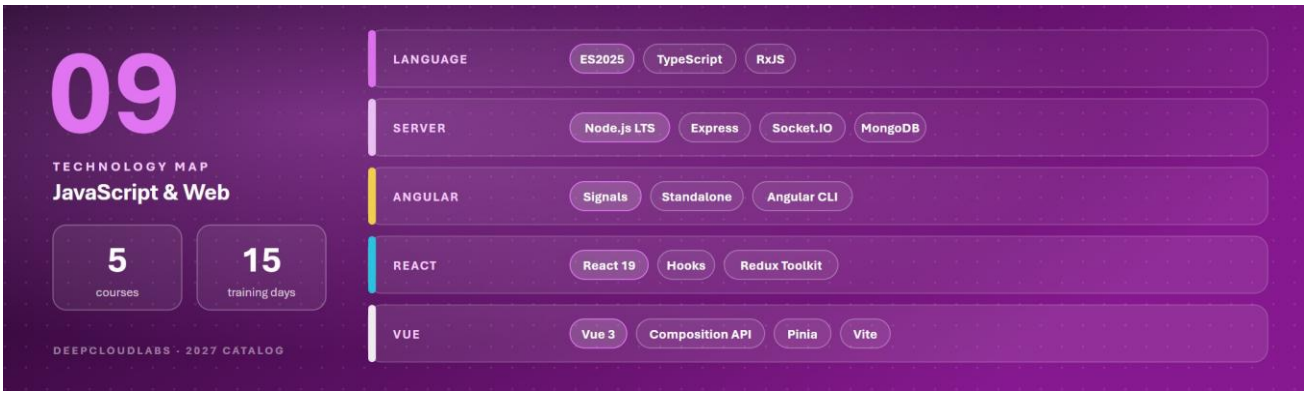
Key standards and frameworks: OWASP Top 10 for LLM Applications (2025 edition) · OWASP Top 10 for Agentic Applications · MITRE ATLAS · NIST AI RMF 1.0 and NIST AI 600-1 Generative AI Profile · ISO/IEC 42001:2023 · Regulation (EU) 2024/1689 (EU AI Act)

Established: widely adopted reference works. Emerging: recent publications on fast-moving practice — check for newer editions.

TRAINING DOMAIN 09 · 5 COURSES · 15 TRAINING DAYS

09 JavaScript & Modern Web

Modern JavaScript and TypeScript, Node.js, and the three leading front-end frameworks.



Whether you build front-end or back-end systems, a solid command of modern JavaScript is essential. This curriculum focuses on the job-ready skills in highest demand: modern ECMAScript and TypeScript, Node.js on the server, and the three leading front-end frameworks — Angular, React and Vue — with hands-on labs throughout.

WHAT'S NEW IN 2027

- ECMAScript coverage updated from "ES6–ES13" to ES2015–ES2025.
- Angular course modernized: standalone components, signals, built-in control flow, zoneless change detection.
- React course aligned with React 19 (Actions, Server Components) and Redux Toolkit.
- Vue course: Vue CLI (maintenance mode) replaced by Vite / create-vue.

PROGRAM OUTLINE

CODE	COURSE	LEVEL	DAYS	PAGE
DCL-302	Node.js Programming	Intermediate	3	91
DCL-304	Advanced JavaScript Programming	Advanced	3	92
DCL-305	Developing Angular Applications	Intermediate	3	93
DCL-306	Developing React Applications	Intermediate	3	94
DCL-318	Developing Vue 3 Applications	Intermediate	3	95
Total	5 courses		**15**	

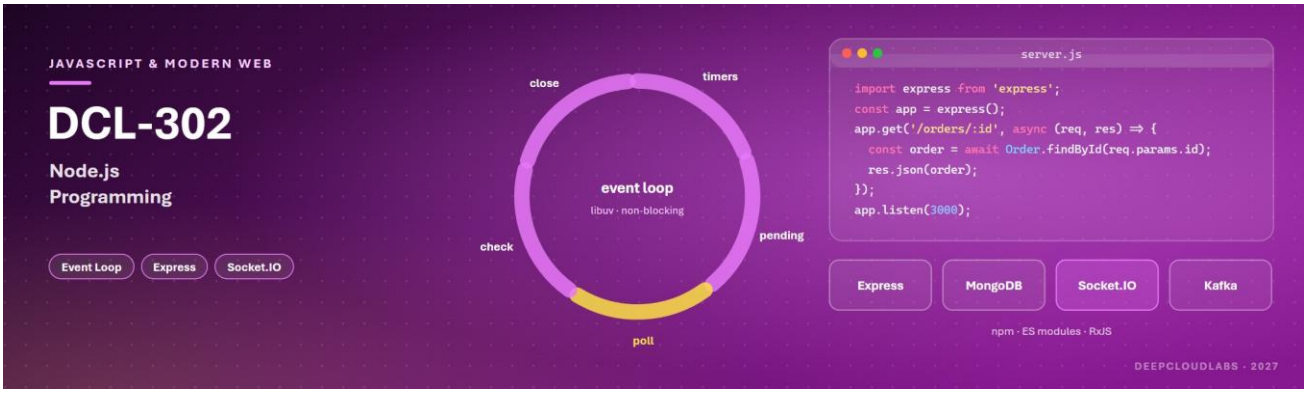
Choosing a front-end framework course

ASPECT	DCL-305 ANGULAR	DCL-306 REACT	DCL-318 VUE 3
Type	Full framework	UI library	Progressive framework
Language	TypeScript-first	JavaScript / TypeScript with JSX	JavaScript / TypeScript with SFC templates
Reactivity & state	Signals, RxJS, DI services	Hooks, Context, Redux Toolkit	Composition API, Pinia
Tooling	Angular CLI	Vite and React tooling	Vite (create-vue)

DCL-302 JAVASCRIPT & MODERN WEB

Node.js Programming

Build, test and launch scalable server-side applications with Node.js.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

In this training participants learn how to build, test and launch Node.js applications. The course covers creating REST APIs with Express, persistence with MongoDB and Mongoose, real-time web applications with Socket.IO, reactive programming with RxJS and event-driven integration with Apache Kafka — all written in modern JavaScript (ES2015–ES2025) on a current Node.js LTS release.

WHAT YOU WILL LEARN

- Explain Node.js architecture and the event loop
- Write and publish modules with npm
- Build REST APIs with Express and MongoDB
- Create real-time applications with Socket.IO
- Integrate services with RxJS and Apache Kafka

WHO SHOULD ATTEND

JavaScript developers moving to the server; full-stack developers.

PREREQUISITES

JavaScript fundamentals.

TECHNOLOGIES

Node.js LTS · Express · MongoDB · Mongoose · Socket.IO · RxJS · Apache Kafka

RELATED COURSES

DCL-304 Advanced JavaScript Programming

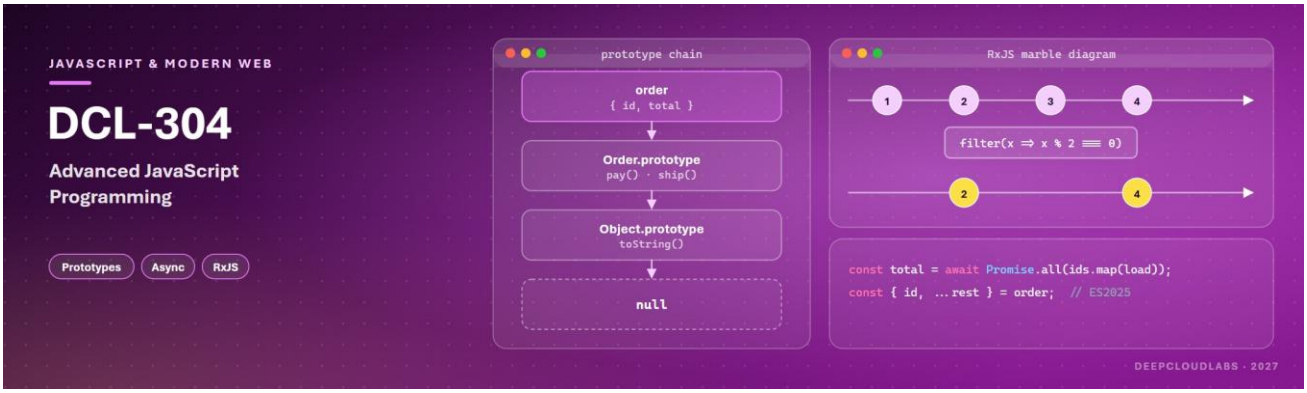
COURSE MODULES

01	Scalable Web Architectures	08	MongoDB
02	Server-Side JavaScript with Node.js	09	Node.js and MongoDB Integration
03	JavaScript	10	Express
04	Advanced JavaScript	11	Socket.IO
05	New Features in ES2015–ES2025	12	Reactive Programming using RxJS
06	Writing Node Modules	13	Event-Driven Programming using Node.js and Apache Kafka
07	Node Package Manager (npm)		

DCL-304 JAVASCRIPT & MODERN WEB

Advanced JavaScript Programming

Objects, functions, concurrency and reactive programming — mastered.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

In this training participants learn advanced JavaScript techniques, working with modern ECMAScript (ES2015–ES2025). It includes a thorough exploration of advanced objects, prototypes, arrays and functions, library and API design, concurrency with promises and async/await, functional programming, and reactive and event-driven programming using RxJS.

WHAT YOU WILL LEARN

- Reason precisely about scope, closures and prototypes
- Design clean, reusable library APIs
- Write correct asynchronous code with promises and async/await
- Apply functional and reactive programming with RxJS
- Use the latest ECMAScript features effectively

WHO SHOULD ATTEND

JavaScript developers seeking deep language mastery.

PREREQUISITES

Working JavaScript experience.

TECHNOLOGIES

ES2025 · TypeScript · RxJS · Promises · async/await

RELATED COURSES

DCL-305 Developing Angular Applications
DCL-306 Developing React Applications
DCL-318 Developing Vue 3 Applications

COURSE MODULES

01	Accustoming Yourself to JavaScript	07	Concurrency
02	Variable Scope	08	Functional Programming
03	Working with Functions	09	Reactive Programming using RxJS
04	Objects and Prototypes	10	Event-Driven Architecture and Event-Driven Programming
05	Arrays and Dictionaries	11	New Features in ES2015–ES2025
06	Library and API Design		

DCL-305 JAVASCRIPT & MODERN WEB

Developing Angular Applications

Enterprise-strength, signal-based Angular applications for desktop and mobile.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This training helps participants learn modern Angular and build responsive, enterprise-strength applications that run smoothly on desktop and mobile. Angular provides a robust framework for richly interactive applications across platforms. Participants gain experience with standalone components, signals for reactive state, the built-in template control flow, services and dependency injection, RxJS, HTTP, routing, forms and testing.

WHAT YOU WILL LEARN

- Build standalone components with the Angular CLI
- Manage reactive state with signals and RxJS
- Use templates, control flow, pipes and directives
- Implement routing, HTTP and forms with validation
- Test Angular applications

WHO SHOULD ATTEND

Front-end and full-stack developers building enterprise web applications.

PREREQUISITES

JavaScript experience; TypeScript basics helpful (DCL-304 recommended).

TECHNOLOGIES

Angular · TypeScript · Signals · RxJS · Angular CLI

RELATED COURSES

DCL-302 Node.js Programming

COURSE MODULES

01	Introduction to Modern Angular	08	RxJS and Observables
02	Writing Applications using Angular CLI	09	HTTP Client
03	TypeScript Essentials	10	Routing
04	Templates, Binding, Control Flow and Directives	11	Pipes
05	Standalone Components	12	Forms and Validation
06	Signals and Reactive State	13	Testing
07	Services and Dependency Injection		

DCL-306 JAVASCRIPT & MODERN WEB

Developing React Applications

Component-driven user interfaces with React 19, hooks and Redux Toolkit.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

React is a declarative, efficient and flexible JavaScript library for building user interfaces with a component-based approach that makes it easy to compose small components into large-scale applications. This training teaches the core knowledge needed to deeply understand and build React 19 components with hooks, manage state with Context, reducers and Redux Toolkit, tune performance, and use Actions and Server Components.

WHAT YOU WILL LEARN

- Build components with JSX, props and hooks
- Handle events, forms and routing
- Manage application state with Context, reducers and Redux Toolkit
- Optimize rendering performance
- Use React 19 Actions and Server Components

WHO SHOULD ATTEND

Front-end and full-stack developers.

PREREQUISITES

JavaScript experience (DCL-304 recommended).

TECHNOLOGIES

React 19 · Hooks · Redux Toolkit · React Router · TypeScript · Vite

RELATED COURSES

DCL-302 Node.js Programming

COURSE MODULES

01	Introduction to React	06	React Routing
02	HTML, CSS and JSX	07	Working with Data using Hooks, Context API and Reducers
03	Data Flow, Rendering and Effects	08	State Management with Redux Toolkit
04	Handling Events	09	Performance Tuning of React Applications
05	Working with Forms and Actions	10	Server Components in React 19

DCL-318 JAVASCRIPT & MODERN WEB

Developing Vue 3 Applications

Highly interactive, enterprise-scale front ends with Vue 3, Pinia and Vite.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Intermediate	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

Vue is a progressive JavaScript framework for building front-end applications that combines approachable templates with a powerful reactivity system. Participants learn the theory behind Vue and how to use it to build highly interactive, large enterprise-level web applications with the Composition API, Pinia state management, Vue Router, testing and server-side rendering — scaffolded with Vite.

WHAT YOU WILL LEARN

- Scaffold and structure Vue 3 applications with Vite
- Build components with templates, directives and the Composition API
- Manage state with Pinia and navigation with Vue Router
- Create advanced rendering, transitions and plugins
- Test components and render on the server

WHO SHOULD ATTEND

Front-end and full-stack developers.

PREREQUISITES

JavaScript experience (DCL-304 recommended).

TECHNOLOGIES

Vue 3 · Composition API · Pinia · Vue Router · Vite · Vitest

RELATED COURSES

DCL-302 Node.js Programming

COURSE MODULES

01	Introduction to Web Architectures and Vue	07	Composition API
02	Creating Vue Applications with Vite (create-vue)	08	Working with External Data
03	Writing a Component	09	Advanced Rendering, Dynamic Components and Plugins
04	Data Binding and Directives	10	Transitions and Animation
05	State Management with Pinia	11	Testing Vue Components
06	Vue Router	12	Server-Side Rendering in Vue

TRAINING DOMAIN 10 · 5 COURSES · 16 TRAINING DAYS

10 C/C++ & Linux Systems

Modern C and C++ (C++23 with a C++26 preview) and Linux system programming.



C and C++ remain the foundation of operating systems, embedded devices, high-performance computing, finance and games. Developers fluent in modern C++ produce efficient, portable software for embedded systems, mobile devices, games and much more. The curriculum is based on C++23 and previews the major C++26 additions — static reflection, contracts and std::execution — whose technical work was completed by the ISO C++ committee in 2026.

WHAT'S NEW IN 2027

- C++ courses retitled "Modern C++ (C++23/26)" with C++26 preview content.
- Multi-threaded C++ adds std::execution (senders/receivers), hazard pointers and RCU.
- Functional C++ adds monadic operations with std::optional and std::expected.

PROGRAM OUTLINE

CODE	COURSE	LEVEL	DAYS	PAGE
DCL-100	C Programming	Introductory	3	97
DCL-112	Object-Oriented Programming in Modern C++ (C++23/26)	Intermediate	4	98
DCL-113	Functional Programming in Modern C++ (C++23/26)	Advanced	2	99
DCL-115	Multi-Threaded Programming in Modern C++ (C++23/26)	Advanced	3	100
DCL-140	Linux System Programming	Advanced	4	101
Total	5 courses		**16**	

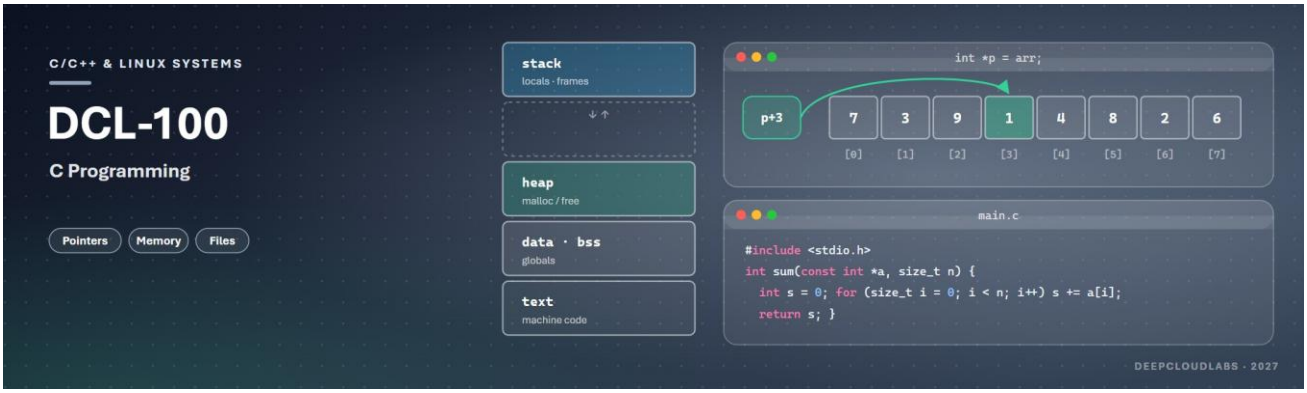
C++ standards in this curriculum

STANDARD	STATUS (OCTOBER 2026)	HIGHLIGHTS TAUGHT
C++20	Published ISO standard	Concepts, ranges, coroutines, std::jthread
C++23	Published ISO standard	std::expected, std::print, monadic std::optional, ranges improvements
C++26	Technically complete (March 2026); ISO publication pending	Static reflection, contracts, std::execution, hazard pointers, RCU (preview)

DCL-100 C/C++ & LINUX SYSTEMS

C Programming

Solid C fundamentals — data, control flow, functions, pointers and files.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Introductory	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This course introduces the fundamentals of programming in C. Participants learn how to work with data, control program flow and use functions; how to create data structures with arrays, strings, structures and pointers; and how to build, organize and run complex C programs using files, the preprocessor and recursion.

WHAT YOU WILL LEARN

- Write structured C programs with functions
- Use arrays, strings and derived data types
- Master pointers and memory
- Read and write files
- Use the preprocessor and recursion effectively

WHO SHOULD ATTEND

New programmers, embedded developers and engineers needing C.

PREREQUISITES

None. Basic computer literacy.

TECHNOLOGIES

C17 / C23 · GCC · Clang · GDB

RELATED COURSES

DCL-112 Object-Oriented Programming in Modern C++ (C++23/26)

DCL-140 Linux System Programming

COURSE MODULES

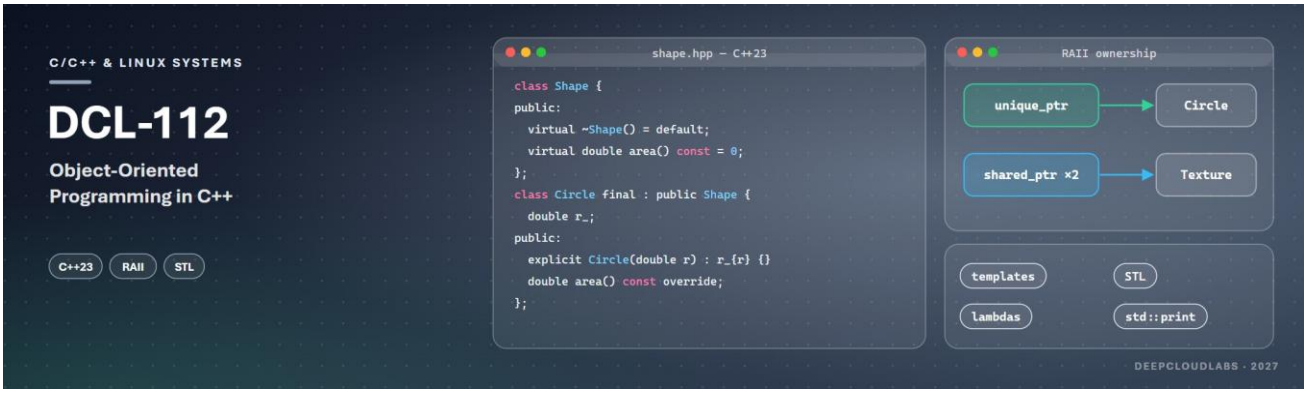
01	Introduction to Computing	07	Multidimensional Arrays
02	Basic C Constructs	08	Functions
03	Selection	09	Pointers
04	Repetition	10	File Operations
05	Derived Data Types	11	Preprocessor
06	Arrays and Strings	12	Recursion

DCL-112 C/C++ & LINUX SYSTEMS

Object-Oriented Programming in Modern C++ (C++23/26)

Updated for 2027 · formerly "Object-Oriented Programming using C++23"

Object-oriented, generic and functional programming in modern C++.



DURATION	LEVEL	DELIVERY	WHERE
4 days	**Intermediate**	**Instructor-led, hands-on**	**Our sites · on-site · your venue**

OVERVIEW

This course introduces several programming paradigms — object-oriented, generic and functional programming — and shows how to apply them with modern C++ to build good programs. It is based on C++23, uses RAII and smart pointers throughout, and previews relevant C++26 additions.

WHAT YOU WILL LEARN

- Design classes with correct construction, copying and destruction
- Manage resources safely with RAII and smart pointers
- Use inheritance and polymorphism appropriately
- Write generic code with templates and the STL
- Apply lambdas, exceptions and modern I/O

WHO SHOULD ATTEND

C and C++ developers; developers from other languages moving to C++.

PREREQUISITES

C programming experience (DCL-100 or equivalent).

TECHNOLOGIES

C++23 · C++26 preview · STL · GCC · Clang · CMake

RELATED COURSES

DCL-113 Functional Programming in Modern C++ (C++23/26)

DCL-115 Multi-Threaded Programming in Modern C++ (C++23/26)

COURSE MODULES

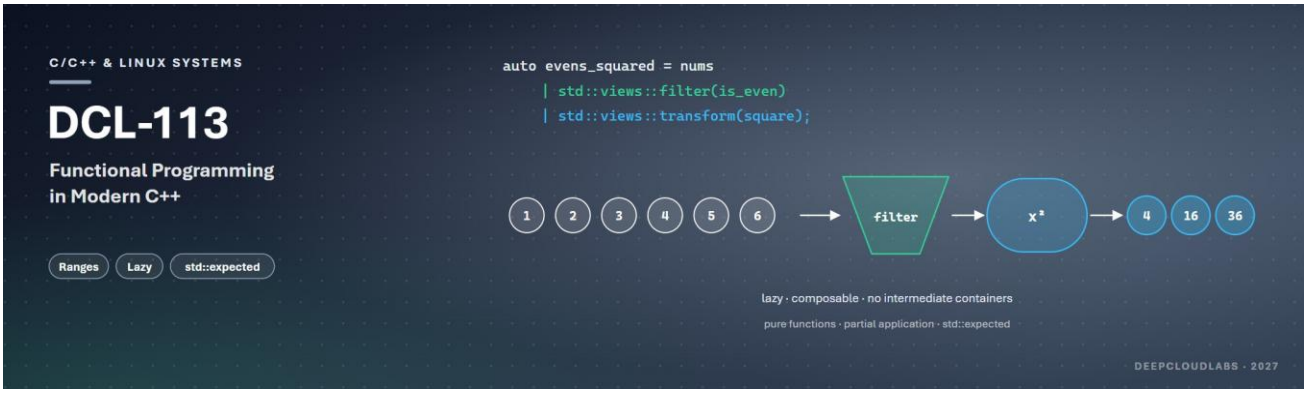
01	Introduction to Object-Oriented Programming	**08**	Polymorphism
02	C++: A Better C	**09**	Lambda Expressions and Closures
03	Classes and Objects	**10**	Exceptions
04	Constructors and Destructors	**11**	Templates
05	Operator Overloading	**12**	The Standard Template Library (STL)
06	Inheritance	**13**	Multithreading
07	Pointers, References and Smart Pointers	**14**	Advanced I/O: Streams, std::format and std::print

DCL-113 C/C++ & LINUX SYSTEMS

Functional Programming in Modern C++ (C++23/26)

Updated for 2027 · formerly "Functional Programming in C++23"

Think functionally in C++ — pure functions, laziness, ranges and composition.



DURATION	LEVEL	DELIVERY	WHERE
2 days	**Advanced**	**Instructor-led, hands-on**	**Our sites · on-site · your venue**

OVERVIEW

This training is not just about the C++ language itself — it is about functional programming and how it fits in with C++. Functional programming offers a different way to think about software design compared to the imperative, object-oriented styles commonly used with C++. The first part covers functional idioms and how to apply them in C++; the second deals with more advanced concepts, mostly pertaining to functional software design, including ranges and monadic error handling.

WHAT YOU WILL LEARN

- Apply functional idioms in C++
- Write pure functions and use lazy evaluation
- Compose algorithms with ranges and views
- Handle errors functionally with std::optional and std::expected
- Design software with functional principles

WHO SHOULD ATTEND

Experienced C++ developers.

PREREQUISITES

Solid C++ experience (DCL-112 or equivalent).

TECHNOLOGIES

C++23 · Ranges · std::expected · Lambdas

RELATED COURSES

DCL-115 Multi-Threaded Programming in Modern C++ (C++23/26)

COURSE MODULES

01	Introduction to Functional Programming	**05**	Pure Functions and Lazy Evaluation
02	Getting Started with Functional Programming	**06**	Ranges
03	Function Objects	**07**	Monadic Operations: std::optional and std::expected
04	Partial Function Application		

DCL-115 C/C++ & LINUX SYSTEMS

Multi-Threaded Programming in Modern C++ (C++23/26)

Updated for 2027 · formerly "Multi-Threaded Programming in C++23"

Correct, scalable concurrency in C++ — from threads and atomics to std::execution.



DURATION	LEVEL	DELIVERY	WHERE
3 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

Multithreaded applications execute multiple threads concurrently to make full use of modern multi-core processors. This training teaches the finer points of multithreading and concurrency and how to apply them efficiently in modern C++. It begins with the fundamentals of concurrency, examines how these concepts work at the hardware level and how operating systems and frameworks use them, and covers the native thread support available since C++11, synchronization, the memory model and atomics, lock-based and lock-free data structures, parallel algorithms — and a preview of C++26 std::execution, hazard pointers and RCU.

WHAT YOU WILL LEARN

- Manage threads and share data safely
- Synchronize with mutexes, condition variables and futures
- Reason about the C++ memory model and atomics
- Design lock-based and lock-free data structures
- Use parallel algorithms and preview C++26 std::execution

WHO SHOULD ATTEND

Experienced C++ developers building concurrent and high-performance systems.

PREREQUISITES

Solid C++ experience (DCL-112 or equivalent).

TECHNOLOGIES

C++23 · std::jthread · Atomics · Parallel STL · std::execution

RELATED COURSES

DCL-140 Linux System Programming

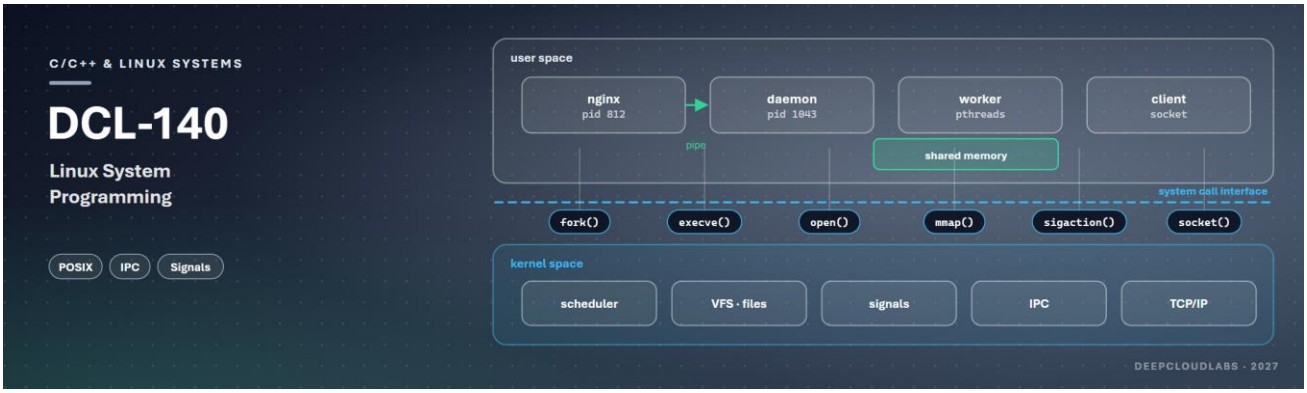
COURSE MODULES

01	Introduction to Concurrency in Modern C++	07	Designing Lock-Free Concurrent Data Structures
02	Managing Threads	08	Designing Concurrent Code
03	Sharing Data between Threads	09	Advanced Thread Management
04	Synchronizing Concurrent Operations	10	Parallel Algorithms: Parallel STL and Ranges
05	The C++ Memory Model and Operations on Atomic Types	11	C++26 Preview: std::execution, Hazard Pointers and RCU
06	Designing Lock-Based Concurrent Data Structures		

DCL-140 C/C++ & LINUX SYSTEMS

Linux System Programming

Processes, files, signals, threads, IPC and sockets — straight from the system-call interface.



DURATION	LEVEL	DELIVERY	WHERE
4 days	Advanced	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This training brings C developers up to speed with the tools and capabilities of Linux, including development and debugging tools as well as system and library functions. Participants learn to use the GNU tools for compiling and debugging and an integrated development environment; to write POSIX-threaded applications; and to use system calls for inter-process communication, file-system interaction, signals, time, daemon creation and scheduling.

WHAT YOU WILL LEARN

- Build and debug programs with the GNU toolchain
- Create and manage processes and daemons
- Work with files, directories and signals
- Write POSIX-threaded applications
- Communicate between processes with IPC and sockets

WHO SHOULD ATTEND

C developers working on Linux, embedded and infrastructure software.

PREREQUISITES

C programming experience (DCL-100 or equivalent).

TECHNOLOGIES

Linux · POSIX · GCC · GDB · pthreads · System calls

RELATED COURSES

DCL-115 Multi-Threaded Programming in Modern C++ (C++23/26)

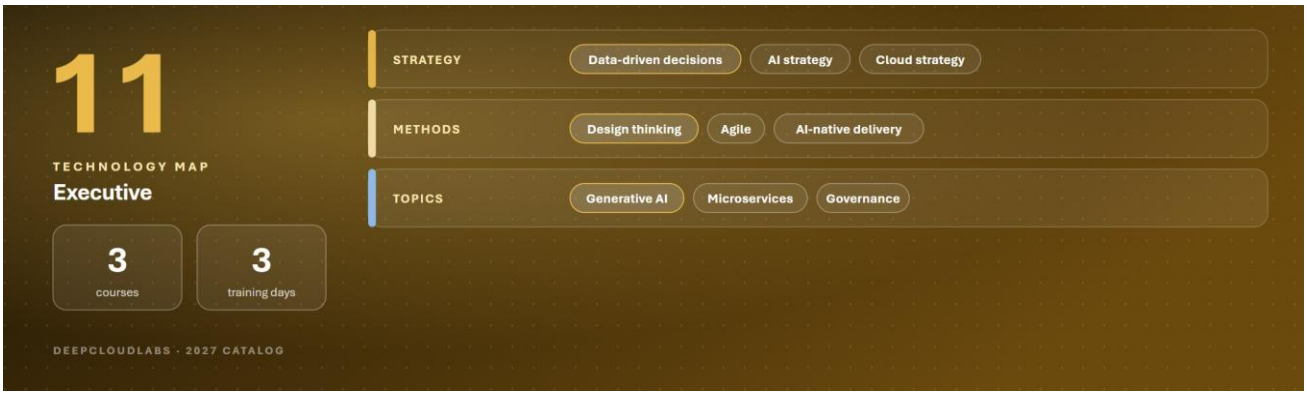
COURSE MODULES

01	Introduction to Linux Programming	07	Threads
02	Spawning New Tasks	08	Overview of IPC
03	System and Process Information	09	Short Messages
04	Files	10	Shared Memory
05	Directories	11	Synchronization
06	Signals	12	Sockets

TRAINING DOMAIN 11 · 3 COURSES · 3 TRAINING DAYS

11 Executive Programs

One-day programs that give leaders the literacy to set technology strategy with confidence.



Technology decisions are business decisions. These one-day programs give executives, directors and managers the data, AI, cloud and software-delivery literacy they need to set strategy, ask the right questions and lead transformation initiatives — without requiring a technical background.

WHAT'S NEW IN 2027

- Big Data and AI for Executives now includes generative AI opportunities, risks and governance.
- New: AI-Native Software Delivery for Technology Leaders (DCL-454) — operating model, metrics and governance for AI-assisted engineering.

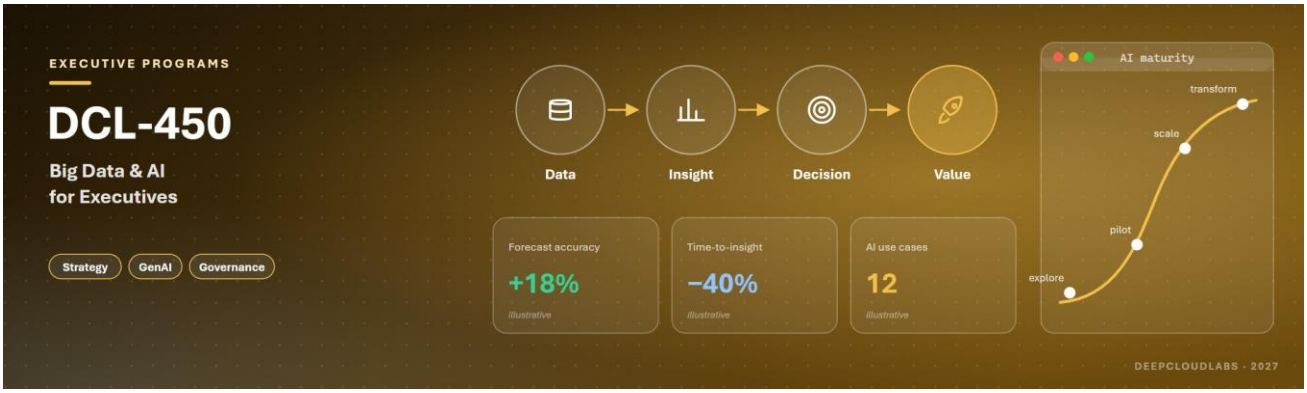
PROGRAM OUTLINE

CODE	COURSE	LEVEL	DAYS	PAGE
DCL-450	Big Data and Artificial Intelligence for Executives	Executive	1	103
DCL-452	Cloud Computing and Microservice Architecture for Executives	Executive	1	104
DCL-454	AI-Native Software Delivery for Technology Leaders	Executive	1	105
Total	3 courses		**3**	

DCL-450 EXECUTIVE PROGRAMS

Big Data and Artificial Intelligence for Executives

Data and AI literacy for leaders — including generative AI opportunities and risks.



DURATION	LEVEL	DELIVERY	WHERE
1 day	Executive	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

Governments and organizations recognize big data and AI — now including generative AI — as a system change, transforming what organizations do and how they do it. This creates challenging tasks for executives and managers. This one-day program builds essential data and AI literacy for senior leaders and lets them learn and practice creative, data-driven strategies that enhance decision making across every facet of the organization.

WHAT YOU WILL LEARN

- Uncover hidden or unexpected connections, correlations, patterns and trends to drive better decisions
- Apply design thinking and agile methodologies to develop big data solutions that are usable and deliver value
- Understand the opportunities, risks and governance needs of generative AI
- Explore the future of big data, machine learning and artificial intelligence

WHO SHOULD ATTEND

C-level and D-level executives.

PREREQUISITES

None.

TECHNOLOGIES

Big data · Machine learning · Generative AI · Design thinking · Agile

RELATED COURSES

DCL-452 Cloud Computing and Microservice Architecture for Executives

DCL-454 AI-Native Software Delivery for Technology Leaders

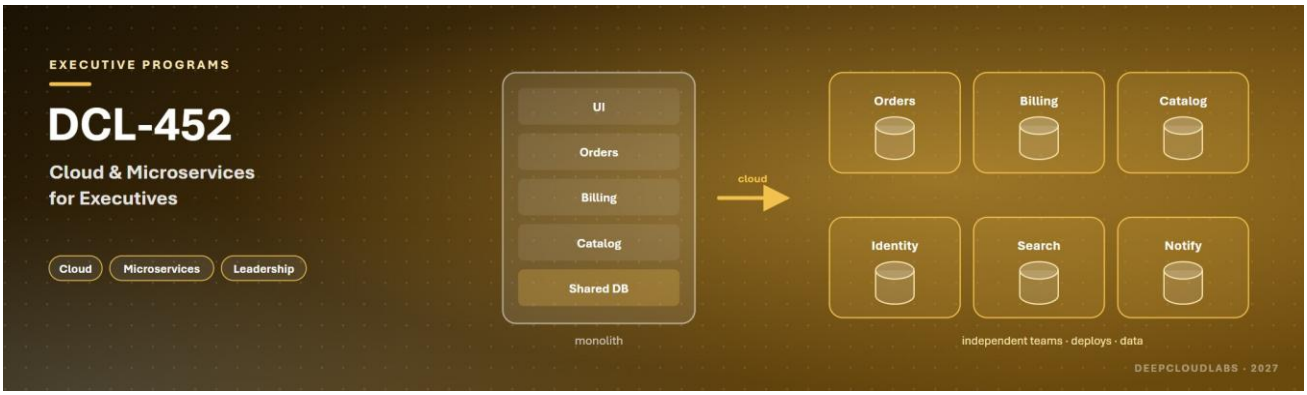
COURSE MODULES

01	Data and AI as a Strategic Capability	04	Generative AI: Opportunities, Risks and Governance
02	Data-Driven Decision Making: Patterns, Correlations and Trends	05	The Future of Big Data, Machine Learning and AI
03	Design Thinking and Agile for Data Solutions That Deliver Value		

DCL-452 EXECUTIVE PROGRAMS

Cloud Computing and Microservice Architecture for Executives

What cloud and microservices mean for your business — and how to lead the change.



DURATION	LEVEL	DELIVERY	WHERE
1 day	Executive	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

This course explains what cloud computing is, and why and how to adopt microservices architecture for your own applications. It answers the questions leaders ask most often — from the challenges of architecting software for the cloud and the problems microservices solve, to the tools and technologies required and how to lead a microservices project successfully.

WHAT YOU WILL LEARN

- Explain cloud computing and its business implications
- Understand what microservices are and which problems they solve
- Recognize the challenges and costs of microservices
- Identify the tools and technologies required
- Lead a microservices initiative successfully

WHO SHOULD ATTEND

IT directors, technology department directors, technology project managers and project managers.

PREREQUISITES

None.

TECHNOLOGIES

Cloud computing · Microservices · DevOps · Kubernetes

RELATED COURSES

DCL-450 Big Data and Artificial Intelligence for Executives

DCL-454 AI-Native Software Delivery for Technology Leaders

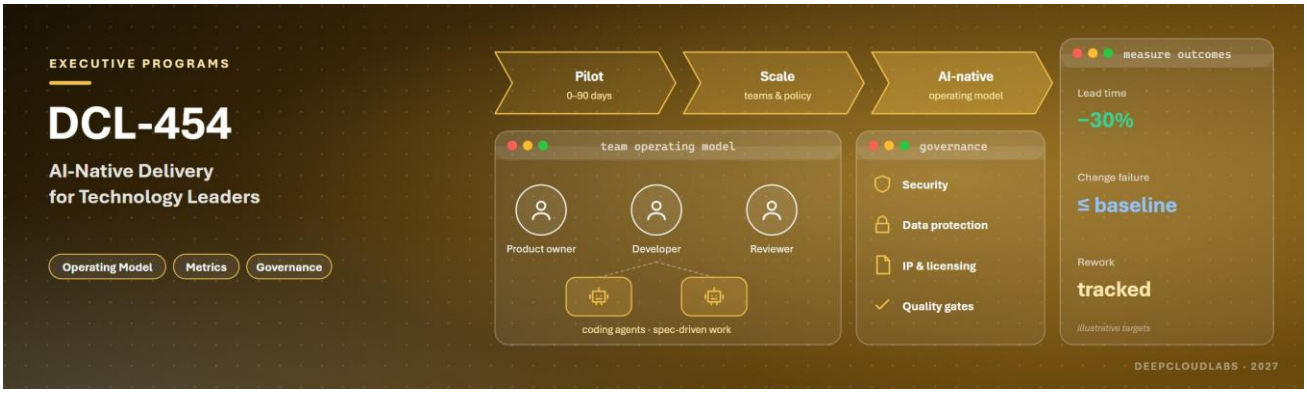
COURSE MODULES

01	What Is Cloud Computing?	06	What Does a Typical Microservices Architecture Look Like?
02	What Are the Challenges of Architecting Software for the Cloud?	07	What Are Some Challenges of Microservices?
03	What Are Microservices?	08	What Tools and Technologies Do I Need?
04	What Problems Do Microservices Solve?	09	How Can I Successfully Lead a Microservices Project?
05	How Can Microservices Help?		

DCL-454 EXECUTIVE PROGRAMS

AI-Native Software Delivery for Technology Leaders

Lead the shift to AI-native software delivery — operating model, evidence of impact, risk and governance.



DURATION	LEVEL	DELIVERY	WHERE
1 day	Executive	Instructor-led, hands-on	Our sites · on-site · your venue

OVERVIEW

AI coding assistants and agents are changing how software is specified, built, tested and operated — and how engineering organizations should be structured, measured and governed. This one-day program gives technology leaders an evidence-based view of AI-native software delivery: what AI tools and agents can reliably do today, where productivity gains are real and where they are not, and how roles, team structures, agile practices and quality gates change. Leaders learn to set policy for tool adoption, data protection, intellectual property and security; to measure impact with delivery and quality outcomes rather than activity counts; and to build a staged adoption roadmap with the skills plan to support it.

WHAT YOU WILL LEARN

- Understand the capabilities, limits and economics of AI coding assistants and agents
- Assess how roles, team structures and agile practices change in AI-native delivery
- Set policy for tool adoption, data protection, intellectual property and security
- Measure the impact of AI on delivery speed, quality and developer experience
- Plan a staged adoption roadmap with the skills to support it

WHO SHOULD ATTEND

CTOs, engineering directors and managers, heads of software delivery, and project and portfolio managers.

PREREQUISITES

None. No programming background required.

TECHNOLOGIES

AI coding agents · Spec-driven development · DORA metrics · AI governance · Change management

RELATED COURSES

DCL-600 AI-Assisted Software Engineering
DCL-602 Specification-Driven Development with AI Agents
DCL-450 Big Data and Artificial Intelligence for Executives

COURSE MODULES

01	The State of AI-Native Software Delivery	04	Risk, Security, Intellectual Property and Governance
02	Operating Model: Roles, Team Structures and Human–AI Collaboration	05	Measuring Impact and Building the Adoption Roadmap
03	Specification-Driven and Agentic Workflows: What Changes in Agile Delivery		

DCL-454 COURSE SPECIFICATION · EXECUTIVE PROGRAMS

AI-Native Software Delivery for Technology Leaders

Learning outcomes, practical components, assessment and recommended reading · 1 day · Executive

LEARNING OUTCOMES — ON COMPLETION, PARTICIPANTS ARE ABLE TO

LO1	Evaluate vendor and internal claims about AI productivity against delivery and quality evidence.	**LO4**	Select a balanced metric set — flow, stability, quality and developer experience — and a plan to baseline it.
LO2	Define a target operating model for one team, including roles, review responsibilities and quality gates.	**LO5**	Produce a 90-day adoption roadmap with pilots, success criteria, risks and a skills plan.
LO3	Draft an AI-assisted development policy covering approved tools, data classification, intellectual property and security.		

PRACTICAL COMPONENTS

#	LABORATORY ACTIVITY	DELIVERABLE AND EVIDENCE
1	Live demonstration and debrief	Capability-and-limitation map
2	Operating-model workshop	Target roles and quality gates for one team
3	Policy drafting	Draft AI-assisted development policy
4	Metrics and roadmap	Metric set and 90-day roadmap

CAPSTONE PROJECT

Each participant leaves with a one-page AI-native delivery plan for their organization, reviewed by peers and the instructor.

ASSESSMENT

COMPONENT	WEIGHT	EVIDENCE AND CRITERIA
Workshop outputs	40 %	Quality and realism of the operating-model and policy drafts
Adoption plan	60 %	Feasible pilots, measurable success criteria and risks addressed

Weights apply when a program requires graded assessment; otherwise labs and the capstone are reviewed formatively with individual feedback.

EXPECTED COMPETENCIES

- Evidence-based technology strategy
- Operating-model design
- AI policy and risk governance
- Outcome-oriented engineering metrics
- Leading organizational change

TOOLS AND TECHNOLOGIES

Frameworks DORA metrics · SPACE framework · Team Topologies · NIST AI RMF

Demonstrated tools Claude Code · OpenAI Codex · GitHub Copilot — compared vendor-neutrally

RECOMMENDED TEXTBOOKS

ESTABLISHED Forsgren, N., Humble, J. & Kim, G. (2018). *Accelerate: The Science of Lean Software and DevOps*. IT Revolution Press.

ESTABLISHED Skelton, M. & Pais, M. (2025). *Team Topologies: Organizing Business and Technology for Fast Flow of Value* (2nd ed.). IT Revolution Press.

EMERGING Kim, G. & Yegge, S. (2025). *Vibe Coding: Building Production-Grade Software With GenAI, Chat, Agents, and Beyond*. IT Revolution Press.

EMERGING Mollick, E. (2024). *Co-Intelligence: Living and Working with AI*. Portfolio.

Established: widely adopted reference works. Emerging: recent publications on fast-moving practice — check for newer editions.

PART III · CONSULTANCY SERVICES

Application Development Consultancy

Engineering services that turn research and expertise into working products — from AI and big data to trading systems and blockchain platforms.

AI & Data

01 **Custom AI & Machine Learning Engineering: From Discovery to Scale**
End-to-end engineering of custom AI and machine learning solutions — from discovery and prototyping to production at scale.

02 **Big Data Solution and Application Development**
Design and development of big data platforms and analytics applications.

03 **Advanced Computer Vision Solution and Application Development**
Computer vision solutions and applications built on deep learning, including image and video analytics.

04 **Advanced Image Processing Solution and Application Development**
Image processing solutions and applications, from enhancement to analysis.

Cloud & Web

01 **Scalable Web Application Development**
Design and development of web applications engineered for scale and maintainability.

02 **Cloud Native Application Development**
Development of cloud-native, microservices-based applications.

Quantitative Finance & Trading

01 **Quantitative Finance & Automated Trading Systems Development for Stock Markets**
Quantitative strategies and automated trading systems for stock markets.

02 **Quantitative Finance & Automated Trading Systems Development for Cryptocurrency Exchange Markets**
Quantitative strategies and automated trading systems for cryptocurrency exchange markets.

03 **Ultra-Low Latency & High-Frequency Trading Application Development**
Ultra-low-latency and high-frequency trading applications.

Blockchain & Digital Assets

01 **Blockchain Application Development**
Design and development of blockchain-based applications.

02 **Wallet Management Application Development for Cryptocurrencies**
Development of wallet management applications for cryptocurrencies.

03 **Cryptocurrency Exchange Platform Development**
Development of cryptocurrency exchange platforms.

© 2027 DEEPCLOUDLABS

PART III · CONSULTANCY SERVICES

Project Management Consultancy

Process, measurement and delivery expertise for software organizations.

Application Lifecycle Management Consultancy Service

Consultancy on application lifecycle management processes and practices.

Managing Enterprise Transition to Agile Methodologies

Guidance for managing an enterprise-wide transition to agile methodologies.

Key Performance Indicator (KPI) Development and Measurement

Definition, development and measurement of key performance indicators.

Proof of Concept Development and Project Benefits and Risks Analysis

Proof-of-concept development combined with analysis of project benefits and risks.

Scrum-Based Project Management and Software Development

Scrum-based project management and software development.

A TYPICAL ENGAGEMENT PATH

- Discovery and proof of concept — clarify goals, validate feasibility, analyze benefits and risks
- Agile implementation — iterative delivery with our engineering team
- Scale — move to production, with knowledge transfer and training for your teams

Consultancy engagements are scoped individually. Combined training-and-consultancy programs — for example an agile transition supported by role-based training — can be designed on request.

PART IV · ENGAGE WITH US

Private Group Classes

Flexible, customizable training solutions that fit your organization’s unique needs.

DEEPCLOUDLABS offers private group classes that provide flexible, customizable training solutions to fit your organization's unique needs. Private training allows an organization to train an entire team or department with one unified learning experience, ensuring that everyone obtains the same knowledge and skills.

Courses can be delivered off the shelf, slightly modified, or completely customized to meet your organization's learning initiatives. Private training can be delivered at any of our locations, on-site at your offices, or at a location of your choice.

Contact us to learn more about our private group class options by phone or e-mail:

PHONE
Head Office: 0 850 259 2 444

E-MAIL
info@deepcloudlabs.com

WEB
www.deepcloudlabs.com

HEAD OFFICE
Biruni Teknopark, Kazlıçeşme Mah. 245. Sk. No:5, Zeytinburnu, İstanbul

WHEN YOU CONTACT US, IT HELPS TO INCLUDE

- Course codes (e.g. DCL-204) or the learning path you are interested in
- Number of participants and their current experience level
- Preferred dates, delivery location and any customization needs

PART IV · LEGAL ENTITY

Company Information

DEEPCLOUDLABS BİLİŞİM TEKNOLOJİLERİ
EĞİTİM VE DANIŞMANLIK HİZMETLERİ TİCARET LİMİTED ŞİRKETİ

MERSİS NO	0272069934700001
VERGİ NO	ZEYTİNBURNU V.D. 2720699347
TİCARET SİCİL NO	116810-5
ADRES (MERKEZ OFİS)	BİRUNİ TEKNOPARK, Kazlıçeşme Mah. 245. Sk. No:5 Zeytinburnu İstanbul
TEL	0 850 259 2 444
E-POSTA	info@deepcloudlabs.com
KEP	deepcloudlabs@hs03.kep.tr
WEB	https://www.deepcloudlabs.com

DEEPCLOUDLABS

Bridging the talent gap with
high-impact technical mastery.

0 850 259 2 444 · info@deepcloudlabs.com · www.deepcloudlabs.com
Biruni Teknopark, Kazlıçeşme Mah. 245. Sk. No:5, Zeytinburnu, İstanbul, Türkiye

© 2027 DEEPCLOUDLABS. All rights reserved. · DCL-CRS-01 · Version 3.1 · 18.10.2026
Course content, versions and durations are subject to change. Product names are trademarks of their respective owners.