COURSE CURRICULUM

AI Capability Programme

From Open-Source Models to Enterprise AI on Azure and Google Cloud

Course Objectives

By the end of the programme, participants will be able to:

- Find, compare and run open-source models with Hugging Face
- Build, assess and deploy ML models in Azure Machine Learning (no-code and code-first)
- Choose between prompting, RAG and fine-tuning, and fine-tune a model in Microsoft Foundry
- Use vibe coding (AI-assisted development) safely to build connected AI apps
- Build tool-enabled agents with MCP on Microsoft Foundry
- Evaluate AI systems for quality, grounding and safety before release
- Build the equivalent solutions on Google Cloud (Gemini Enterprise Agent Platform / Vertex AI)

Participant Requirements

Cloud subscriptions: To make the most of the hands-on labs, participants are kindly requested to bring their own active **Microsoft Azure** and **Google Cloud Platform (GCP)** subscriptions. This allows every participant to practise in their own environment and to keep access to their work after the programme. We would be grateful if the subscriptions could be set up and ready before the first session.

Duration

Module	Hours
1. Hugging Face Ecosystem and Responsible Model Selection	4
2. Transformers, Pipelines and Tokenisation	4
3. Azure Machine Learning Foundations	4
4. No-Code ML: Designer and Automated ML	4
5. Code-First Training, Responsible AI and Deployment	8
6. Model Customisation: Prompting vs RAG vs Fine-Tuning	8
7. Vibe Coding: AI-Assisted Development	8
8. AI Agents with MCP on Microsoft Foundry	8
9. AI Quality, Safety and Evaluation	8
10. Google Cloud AI: Gemini Enterprise Agent Platform (Vertex AI)	8
Total (16 days x 4 hours)	**64**

Modules

Module 1: Hugging Face Ecosystem and Responsible Model Selection (4 hrs)

- Navigate the Hub: models, datasets, Spaces
- Read model cards, dataset cards and licences
- Compare models by task fit, size, benchmarks, hardware and data sensitivity
- Document a go / pilot / reject decision

Lab: Shortlist and score models for resident-feedback analysis; explore models in Hugging Face Spaces

Module 2: Transformers, Pipelines and Tokenisation (4 hrs)

- Tasks, checkpoints, AutoModel and AutoTokenizer
- Pipelines: classification, NER, Q&A, summarisation, speech
- Subword tokens, special tokens, attention masks, context limits
- Why token counts drive cost, latency and quality

Lab: Run classification and NER on Council text; compare two tokenizers

Module 3: Azure Machine Learning Foundations (4 hrs)

- Workspace, compute, data assets, environments, jobs, models, endpoints
- Data quality checks, leakage, imbalance and sensitive attributes
- Versioning data, code and environments for repeatability

Lab: Create a versioned data asset and environment for a service-request dataset

Module 4: No-Code ML: Designer and Automated ML (4 hrs)

- Build visual pipelines with Designer
- AutoML for classification, regression and forecasting
- Primary metrics, featurisation, validation and exit criteria
- Read the leaderboard and pick the right model

Lab: Designer pipeline + AutoML classification and demand forecasting

Module 5: Code-First Training, Responsible AI and Deployment (8 hrs)

- Training jobs with SDK v2 / CLI v2 and MLflow tracking
- Register models with versions and tags
- Responsible AI dashboard: error analysis, fairness, interpretability
- Responsible AI scorecard (PDF) for non-technical stakeholders
- Deploy to a managed online endpoint, call it and monitor it

Lab: Train, register, assess and deploy a model; call it from a test client

Module 6: Model Customisation: Prompting vs RAG vs Fine-Tuning (8 hrs)

- Simple rule: start with prompting, add RAG when the model needs your documents, fine-tune only when behaviour or format must change
- Compare the three on cost, data needed, freshness, latency and effort
- Prepare a small training file (format, quality checks, train/validation split)
- Run supervised fine-tuning in Microsoft Foundry and compare with the baseline
- Awareness of preference-based and reinforcement fine-tuning
- Evaluate, deploy and test the fine-tuned model; cost and hosting considerations

Lab: Score three Council use cases, then run a fine-tuning job, deploy it and make a go / no-go call

Module 7: Vibe Coding: AI-Assisted Development (8 hrs)

- What vibe coding is and where it works well
- Prompting a coding assistant (GitHub Copilot) to build and refactor an app
- Reviewing AI-generated code: security, packages, licences, tests
- Connecting the app to a Foundry model: endpoints, SDK calls, keys vs managed identity
- Replace hard-coded keys with passwordless (Entra ID) sign-in
- Introduction to Model Context Protocol (MCP) and where it fits in an app

Lab: Vibe-code a small Council app, connect it to a deployed Foundry model and make it passwordless

Module 8: AI Agents with MCP on Microsoft Foundry (8 hrs)

- Foundry Agent Service: model, instructions, tools, runs
- Model Context Protocol: host, client and server roles
- Build a custom MCP server and connect it to an agent
- Register MCP and OpenAPI tools with an agent
- End-to-end walkthrough: user request, tool call, agent response
- Agent safety: least privilege, human approval, end-to-end tracing

Lab: Build an MCP server for a sample Council data source, add an OpenAPI tool, and trace one request

Module 9: AI Quality, Safety and Evaluation (4 hrs)

- Evaluation datasets, evaluators and pass thresholds
- RAG evaluation: relevance, groundedness, retrieval quality
- Safety evaluators and red-team testing
- Agent evaluation: intent resolution, tool-call accuracy, task adherence
- Guided portal activity: RAG evaluation in Microsoft Foundry
- Reading results and setting a repeatable release checklist

Lab: Run a RAG and safety evaluation in Foundry and present a release decision

Module 10: Google Cloud AI: Gemini Enterprise Agent Platform (Vertex AI) (12 hrs)

- Platform tour and Foundry-to-Google mapping
- Model Garden: Gemini, Gemma, open and partner models
- Vertex AI Studio: prompt design and grounding with Vertex AI Search (RAG)
- Supervised fine-tuning of Gemini models
- Agent Development Kit (ADK) and Agent Engine; MCP tool support
- Gen AI evaluation service (adaptive rubrics) and agent evaluation
- Security on GCP: IAM, service accounts, VPC Service Controls

Lab: Build a grounded Gemini agent with ADK, deploy it to Agent Engine, and evaluate it