

# Artificial Intelligence - Curriculum

## 1. AI Literacy

|                                                                 |                                                                                                                                                                                                         |
|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1.1 What Artificial Intelligence is (and what it is not)</b> | Difference between traditional software, rule-based systems and AI systems. Not all technology is AI and not all robots are AI. How AI learns from data and how that differs from the way humans learn. |
| <b>1.2 Everyday AI tools</b>                                    | AI that is already embedded in the tools we use daily - search, maps and travel time, recommendations, autocorrect, camera effects and voice assistants.                                                |
| <b>1.3 Using AI tools</b>                                       | Working with AI tools to produce text, images, video, speech, sound effects and music from a prompt.                                                                                                    |
| <b>1.4 Prompting and prompt effectiveness</b>                   | How the wording and detail of a prompt shape the output. Earlier prompts carry context into later ones. The same prompt can produce different results each time.                                        |
| <b>1.5 Reliability and limitations of AI systems</b>            | AI predicts, it does not verify. A confident answer can still be wrong. Why AI output has to be checked against a real source.                                                                          |
| <b>1.6 Bias, safety and ethics of AI usage</b>                  | Bias in training data and how it surfaces in output. Hallucination, inconsistency, privacy and trust. Responsible use and fact checking.                                                                |
| <b>1.7 Generation versus retrieval</b>                          | The difference between creating something new and looking up something that already exists, and why a search engine and a generative model answer the same prompt differently.                          |

## 2. AI and ML Concepts - A Simplified Overview

|                                            |                                                                                                                                                                                   |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>2.1 Types of AI</b>                     | Generative models, classification models and detection models, and where each type is used.                                                                                       |
| <b>2.2 Generative AI fundamentals</b>      | Generation as prediction of the next unit from learned patterns rather than understanding of meaning, emotion or intent. The same principle across text, images, audio and video. |
| <b>2.3 Pattern recognition</b>             | How a model identifies something from a small set of learned visual or acoustic features instead of from meaning.                                                                 |
| <b>2.4 Digital representation of media</b> | Images as pixels, video as a sequence of frames, sound as a waveform. How media is turned into data a model can process.                                                          |
| <b>2.5 Training data and labels</b>        | The role of examples and labels in what a model learns. Dataset size, range and balance, and the effect of incorrect or insufficient data.                                        |
| <b>2.6 Supervised learning</b>             | Learning from labelled examples, and the label - train - test - improve cycle.                                                                                                    |
| <b>2.7 Model accuracy and confidence</b>   | Confidence scores, thresholds, and why a high confidence score is not proof that a prediction is correct.                                                                         |
| <b>2.8 Tokens</b>                          | Text broken into tokens for processing. Images, audio and video also converted into tokens. Token windows and context limits.                                                     |

|                                       |                                                                                                                                               |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>2.9 Embeddings</b>                 | Meaning represented as numbers, so that similar meanings sit closer together and different meanings farther apart.                            |
| <b>2.10 Neural networks</b>           | Patterns held as weights across a network. Hidden layers between input and output. Models learn by adjusting weights in response to feedback. |
| <b>2.11 Attention mechanism</b>       | Importance weighting across the input, so a model focuses on the parts that matter most to the response.                                      |
| <b>2.12 Limitations of models</b>     | Limits of the training data, behaviour on unseen or unusual input, and the fact that many models do not reason to produce their output.       |
| <b>2.13 Why AI models hallucinate</b> | Why a model produces confident but false output, how to recognise it, and how to verify an answer.                                            |

### 3. Programming Foundations for AI

|                                           |                                                                                                                                                                               |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>3.1 Canvas and coordinate system</b>   | Creating a drawing surface and positioning elements using coordinates.                                                                                                        |
| <b>3.2 Variables and the draw loop</b>    | Storing values in variables and updating them inside a repeating draw loop to produce motion.                                                                                 |
| <b>3.3 Conditionals and program state</b> | Making a program behave differently depending on its current state or on what a model has returned.                                                                           |
| <b>3.4 Arrays and iteration</b>           | Recap of array and loop concepts from the previous course, applied to the collections a model returns - detected points, faces or objects - and worked through one at a time. |
| <b>3.5 Media assets in code</b>           | Loading and drawing images, and positioning them relative to values produced at runtime.                                                                                      |
| <b>3.6 User input and interaction</b>     | Responding to input from the keyboard, mouse, camera or microphone while the program is running.                                                                              |
| <b>3.7 Spatial and collision logic</b>    | Comparing positions to decide whether two things overlap, and acting on the result.                                                                                           |

### 4. Working with Pre-trained AI Models

|                                                       |                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>4.1 Pre-trained models and inference</b>           | What it means to use a model somebody else has already trained, and how a program sends input to it and receives a prediction back. The pre-trained models covered in this course are the hand model, the face detection model, the facial landmark model, the object detection model, the audio classification model and the speech synthesis model. |
| <b>4.2 Real-time input from camera and microphone</b> | Feeding a live webcam or microphone stream into a model frame by frame, and the constraints that come with working in real time.                                                                                                                                                                                                                      |
| <b>4.3 Hand model and keypoints</b>                   | Detecting a hand and reading the set of finger and knuckle keypoints it returns rather than a single label.                                                                                                                                                                                                                                           |
| <b>4.4 Face detection and facial landmark points</b>  | Locating a face in an input, then reading the full set of landmark points that describe its features.                                                                                                                                                                                                                                                 |

|                                                 |                                                                                                                                                                |
|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>4.5 Object detection and localisation</b>    | Identifying what is present in an input and where it is, using bounding boxes.                                                                                 |
| <b>4.6 Audio classification</b>                 | Identifying what a sound is from a live or recorded audio input.                                                                                               |
| <b>4.7 Multiple-instance detection</b>          | Detecting and tracking more than one subject at the same time, and telling them apart - multiple hands, multiple faces, multiple objects in one frame.         |
| <b>4.8 Mapping model output to an interface</b> | Turning the coordinates and labels a model returns into something visible or audible - bounding boxes, tracked overlays, accessories aligned to a moving face. |
| <b>4.9 Detection failure and edge cases</b>     | Missed detections, misclassification, low confidence, and subjects that are partially hidden or unusual.                                                       |
| <b>4.10 Speech synthesis</b>                    | Turning text produced by a program into spoken output.                                                                                                         |

## 5. Training Custom AI Models

|                                             |                                                                                                                                                                                                      |
|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>5.1 Defining classes for a model</b>     | Deciding what categories a model should distinguish between, and how that decision shapes everything after it.                                                                                       |
| <b>5.2 Collecting and labelling samples</b> | Gathering image or sound examples for each class, including samples the students generated themselves, and keeping the set balanced. How an unbalanced or narrow set biases what the model predicts. |
| <b>5.3 Training a model</b>                 | Running the training process, and what the model is actually doing while it trains.                                                                                                                  |
| <b>5.4 Evaluating model performance</b>     | Testing a trained model, finding where it fails, and reading the results honestly.                                                                                                                   |
| <b>5.5 Improving a model through data</b>   | Raising accuracy by changing the dataset rather than the code.                                                                                                                                       |
| <b>5.6 Exporting and deploying a model</b>  | Taking a trained model out of the training tool and running it inside an application.                                                                                                                |

## 6. Working with Language Models

|                                             |                                                                                                                         |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>6.1 Calling a language model</b>         | Sending a prompt to a large language model from within an application, receiving the response, and handling it in code. |
| <b>6.2 System prompts and roles</b>         | Pre-defining instructions that set the model's persona, scope and tone before any user input arrives.                   |
| <b>6.3 Context and conversation history</b> | Passing previous turns back to the model so a conversation stays coherent, and what happens at the context limit.       |
| <b>6.4 Model parameters</b>                 | Settings such as temperature that shift output between predictable and creative.                                        |

## 7. Designing AI-enabled Systems

|                                                   |                                                                                                                    |
|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| <b>7.1 AI as part of a system</b>                 | Seeing a model as one component among many, rather than as the whole solution.                                     |
| <b>7.2 Problem definition and model selection</b> | Stating a real problem clearly and choosing the kind of model that actually fits it.                               |
| <b>7.3 Integrating models into applications</b>   | Driving the features of a working application from model output.                                                   |
| <b>7.4 Multimodal and multi-model workflows</b>   | Combining vision, audio and language models so that one system handles more than one kind of input.                |
| <b>7.5 Model pipelining</b>                       | Chaining models so that the output of one becomes the input to the next, and deciding the order in which they run. |
| <b>7.6 Testing and failure handling</b>           | Testing an application, identifying edge cases, and deciding what it should do when the model is wrong or unsure.  |