**Generative Artificial Intelligence**

**Special Diploma module- Generative AI**

**Generative AI [60 hours]**

Description

Advances in generative models have allowed us to create increasingly realistic text

and images. These models, including large language models and multimodal models,

are showing promising signs of reasoning abilities, thus opening up exciting

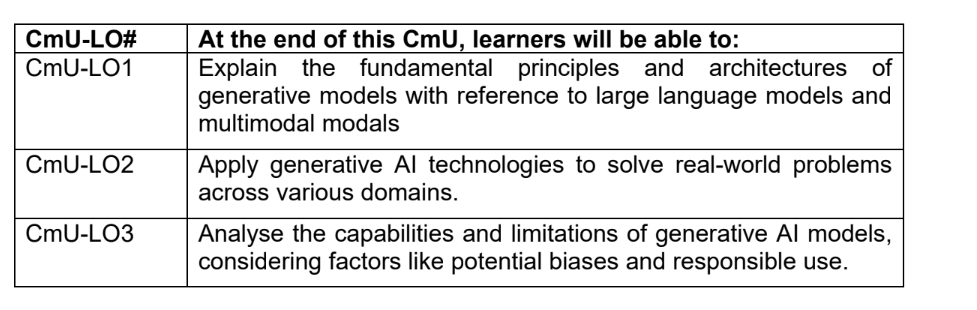
possibilities for new applications and enabling faster development of AI solutions. In

this unit, learners will gain an in-depth knowledge of the capability and limitations of

Generative AI technologies and get hands-on experience applying them to solve real-

world problems.

Learning Outcomes



**Course Structure and Weekly Topics**

| Week | Materials |
| --- | --- |
| **Topic 1: Introduction to Generative AI** |  |
| **Week 1: Introduction & Landscape**   * What is Generative AI? Definitions and overview * Real-world applications: text, image, audio, video * Evolution: Traditional ML → Deep Learning → Generative AI * Overview of model types: VAE, GAN, Diffusion, Transformers |  |
| **Week 2: Ethics and Social Implications**   * Ethical issues: hallucination, bias, misinformation * Social impact: automation, trust, deepfakes * Responsible AI principles * Open-source vs proprietary model | Material from SCS BoK |
| **Topic 2: Core Generative Models** |  |
| **Week 3: VAE and GANs**   * Variational Autoencoders (VAEs): theory and coding * GANs: architecture, training challenges * Lab: Build a VAE and a DCGAN | Consider to combine week3 & 4 |
| **Week 4: Advanced GANs & Diffusion**   * StyleGAN, CycleGAN * Diffusion Models: DDPM, Stable Diffusion overview * Lab: Text-to-image generation with pretrained diffusion models |  |
| **Topic 3: Transformers & LLMs** |  |
| **Week 5: Transformer Architecture & Large Language Models**   * Attention mechanisms * Encoder, decoder, and encoder-decoder models * Introduction to Transformer architecture * BERT, GPT, T5 – Architecture and applications * Pretraining and fine-tuning workflows | Can improve from the ITI108 materials |
| **Week 6: Prompt Engineering & LangChain/llamaindex**   * Prompting strategies: zero-, few-, chain-of-thought * LangChain concepts: chains, tools, memory * Prompt Chaining * LLM Routing * LLM Parallelization * Lab: Q&A system using LangChain/llamaindex | Can improve from the ITI108 materials |
| **Week 7: Fine-Tuning LLMs**   * Dataset preparation * Fine-tuning with Hugging Face Trainer * Evaluation of LLM outputs * Lab: Fine-tune a small LLM on a custom dataset | Kheng Keng mentioned he had some materials |
| **Module 4: Generative Applications** |  |
| **Week 8: Text Generation**   * Applications: chatbots, summarization, translation, code generation * Tools: OpenAI API, Hugging Face pipelines * Lab: Build a custom chatbot or summarizer | Can improve from the ITI108 materials  If we have good use cases for each, the individual topics will be meaningful  Week 8 to 10  -The coverage will be focus on the end user and the developer  Eg. End user - use prompt application to automate task  Developer - use prompt program to integrate to different application |
| **Week 9: Image Generation**   * Stable Diffusion, DALL·E, Style Transfer * Prompt engineering for visuals * Lab: Image generation from prompts |  |
| **Week 10: Audio & Speech**   * STT (Whisper) * Music and audio synthesis models * Lab: Build a basic STT |  |
| **Topic5: Tools, Application & Deployment** |  |
| **Week 12: LLM Applications**   * What is an LLM Application? * Input processing (preprocessing, chunking, embeddings) * LLM model interaction (API/local inference, token management) * Memory management (context windows, vector DBs) * Decision logic and agent-style flows * Output generation and formatting * Tool integration (search, APIs, file tools) * Monitoring and analytics (token usage, logs) * UI design (Gradio/Streamlit) * **Lab:** Build a complete LLM-powered tool | Can improve from the ITI108 materials  Botpress  Langgraph  Llamaindex  https://zapier.com/ |
| **Week 13: Retrieval-Augmented Generation (RAG)**   * Retriever-generator architecture * Vector search (FAISS, Chroma) * Semantic search, query rewriting   + What are embeddings? From discrete tokens to dense vectors   + Generating embeddings with LLMs (e.g., sentence-transformers, OpenAI embeddings)   + Use in semantic search, clustering, retrieval   + Cosine similarity, Euclidean distance in vector space * Intro to vector databases (FAISS, Chroma, Weaviate)   + Lab: RAG-based document Q&A system | Can improve from the ITI108 materials  <https://notebooklm.google/>  To cover for 2 groups of learners  -End user  -Developer |
| **Week 8: Evaluating Relevance of Search & Generated Results**  * Precision, Recall, F1 for search systems * Relevance scoring: cosine similarity thresholds * Human evaluation: faithfulness, coherence, hallucination * Tools for evaluation: LangChain eval, RAGAS, custom metrics |  |
| **Week 14: Introduction to Agentic AI (Single-Agent Systems)**  * What is an AI agent? * Core components   + LLM + memory + tools + planning * Type of agents * Tool usage, action loops, prompt chaining * Frameworks:   + LangChain Agents   + Auto-GPT   + CrewAI (single agent setup) * Safety, limitations, and guardrails | Can improve from the ITI108 materials |
| **Week 15: Multi-Agent Systems & Advanced Agentic Patterns**   * **Multi-Agent Collaboration Concepts:** Why use multiple agents? Role specialization Multi-agent planning and task decomposition Message passing and communication protocols * **Frameworks & Tools:** CrewAI (teams of agents) LangGraph (agent workflows as state machines) AutoGen (Microsoft), SWARM pattern * **Control & Coordination:**   Agent orchestration  Managing shared memory or knowledge  Stopping conditions and failure recovery   * **Safety in multi-agent system**   Prompt injection between agent  Runaway loops |  |
| **Week 16: Multi-Agent Systems & Model Context Protocol**   * **Understanding MCP fundamentals:** This involves grasping the client-server architecture, the purpose of MCP, and how it enables LLMs to access and utilize external context. * **Building and running MCP servers:** You'll learn how to create MCP servers that expose resources, tools, and prompts to LLMs. * **Integrating MCP with various AI tools:** The course will cover how to connect MCP to different platforms like Claude Desktop, allowing you to build applications with real-time information access and tool usage.   Benefits of learning MCP:   * **Enhanced AI capabilities:** MCP enables AI models to access external data, tools, and APIs, leading to more intelligent and capable AI applications. * **Simplified integration:** MCP standardizes how LLMs interact with external sources, making it easier to integrate them into AI applications. * **Improved AI agent workflows:** MCP helps automate tasks and create more complex agent behaviors by leveraging contextual data. * **Increased versatility:** MCP allows developers to build AI applications that can adapt to different contexts and tasks by connecting to various resources and tools. | https://huggingface.co/learn/mcp-course/en/unit0/introduction |
| **Week 16: Model Deployment**   * Full Stack Generative AI * Deployment tools: Gradio, Streamlit, FastAPI * Hosting on Hugging Face Spaces or cloud * Sharing AI tools with users * Lab: Deploy a working GenAI application |  |
| Week 17: LLM Application Security   * LLM Threat Models: prompt injection, data leakage, overreliance * Prompt Injection Attacks: direct & indirect, real-world examples * Mitigation Techniques: input/output filtering, guardrails, role limits * Safety Frameworks: OpenAI Moderation API, Guardrails AI, LangChain * Adversarial Testing: red teaming, fuzzing, stress tests * Logging & Monitoring: traceability, anomaly detection, kill switches |  |
|  |  |
|  |  |

Assessment:

1. Test - Concept of the topics
2. Assignment - use case to apply the GenAI to solve a problem
3. ? - To decide