**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

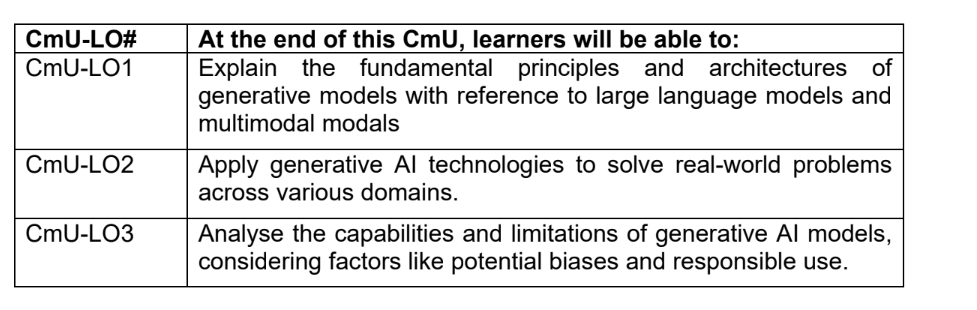
possibilities for new applications and enabling faster development of AI solutions. In

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** |  |
| **Lesson 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 |  |
| **Topic 2: Transformers & LLMs** |  |
| **Lesson 2: Transformer Architecture & Large Language Models**   * Overview of Transformer architecture * Embedding Application: **semantic search/clustering, classification/anomaly detection, and recommendation/multimodal retrieval** * Prompting strategies: zero-, few-, chain-of-thought | Can improve from the ITI108 materials  --Overview architecture of different embedding model  \*embedding will be use for RAG search  https://skillsbuild.org/college-students/artificial-intelligence |
| **Lesson 3: Prompt Engineering**   * LangChain concepts: chains, tools, memory * Prompt Chaining * LLM Routing * LLM Parallelization * Lab: Q&A system using LangChain/llamaindex * Fine-tuning with Llamaindex API * Evaluation of LLM outputs * Lab: Fine-tune a small LLM on a custom dataset | Can improve from the ITI108 materials  To decide framework to use  **LangChain/llamaindex**  Llamaindex API  <https://docs.llamaindex.ai/en/stable/examples/finetuning/mistralai_fine_tuning/> |

|  |  |
| --- | --- |
| **Topic 3: LLM Application, Tool & Deployment** |  |
| **Lesson 4: 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) -\*overview,detail will be in agent * Monitoring and analytics (token usage, logs) * Debug * UI design (Gradio/Streamlit) * **Lab:** Build a complete LLM-powered application | Can improve from the ITI108 materials  Botpress  Langgraph  Llamaindex  <https://zapier.com/>  Debugging |
| **Lesson 5: 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  Event drive worflow  <https://medium.com/the-ai-forum/implementing-advanced-rag-using-llamaindex-workflow-and-groq-bd6047299fa5>  DAG – graph methods  https://www.llamaindex.ai/blog/introducing-query-pipelines-025dc2bb0537 |
| **Lesson 6: 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 |  |
| **Lesson 7: Ethics and Social Implications**   * Responsible AI principles, focus on generative AI * Ethical issues: hallucination, bias, misinformation * Social impact * Case Studies | Material from SCS BoK  SCS BoK ↔ MGF‑GenAI (9 Dimensions)  -case studies: find case studies for the learner to present |
| Lesson 8:  Test – Lesson 1 to 7 - 40% |  |
| **Lesson 9: Ethics and Social Implications(Guardrail)**   * Types of Guardrails & Their Roles * How Guardrails Work: * Benefits & Deployment Strategy of Guardrails * Practical: project moonshot | <https://www.abs.org.sg/docs/library/handbook-on-generative-ai-guardrails-in-banking.pdf>  <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-are-ai-guardrails>  https://aiverifyfoundation.sg/project-moonshot/ |

|  |  |
| --- | --- |
| **Topic 3: Generative AI Applications** |  |
| **Lesson 10: Text Generation**   * Applications: chatbots, summarization, translation, code generation * Tools: OpenAI API, Hugging Face pipelines * Lab: Build a custom chatbot or summarizer | flowise |
| **Lesson 11: Image Generation**   * Stable Diffusion, DALL·E, Style Transfer * Prompt engineering for visuals * Lab: Image generation from prompts | Comfyui  <https://comfyai.run/>  <https://platform.openai.com/docs/guides/image-generation?image-generation-model=gpt-image-1> |
| Lesson 12  Assignment 1- use case presentation – 15% |  |
| **Lesson 13: 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  <https://docs.llamaindex.ai/en/stable/understanding/>  <https://billtcheng2013.medium.com/agentic-ai-frameworks-ee3ca711815b> |
| **Lesson 14: 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 | Will cover A2A |

|  |  |
| --- | --- |
| **Lesson 15: Audio & Speech(AI Audio Agent)**   * Understand the role of audio agents in AI systems * Define what an audio agent is and how it fits within a multi-modal or interactive AI architecture. * Explore real-world applications (e.g., virtual assistants, call center automation, voice-controlled devices). | Openai audio agent  TTS  <https://cookbook.openai.com/examples/agents_sdk/app_assistant_voice_agents>  https://github.com/openai/openai-realtime-agents |

|  |  |
| --- | --- |
| **Lesson 16: Multi-Agent Systems & Advanced Agentic Patterns**   * Understand MCP’s client-server architecture for connecting LLMs to external tools and data. * Learn to build and run MCP servers that expose resources and prompts. * Integrate MCP with platforms like Claude Desktop for real-time access. | https://huggingface.co/learn/mcp-course/en/unit0/introduction |
| Lesson 17  Assignment 2- Programming GenAI application -45% |  |
| Lesson 18: 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 1- use case to apply the GenAI to solve a problem
3. Assignment 2 - To decide what programming scenario