INTERNATIONAL QUALIFICATIONS

AND ASSESSMENT CENTRE (IQAC)

Approved»

2025 y. «___» _____

Programme	**LEVEL 4 EXTENDED DIPLOMA IN ARTIFICIAL INTELLIGENCE**		
Unit Number/ Unit Title	UNIT 2 PROGRAMMING WITH PYTHON		
Cohort Code:	L04PRT-U2		
Unit Level	4		
Total GLH	Total qualification time 200/ Total Guided learning hours 90/ Self-guided learning hours 110		
Credits	20 CATS/ 10 ECTS		
Lecturer			
Start Date		**End Date**	

Unit Aims	The unit aims to equip students with the programming skills necessary to implement AI algorithms, using Python as the primary programming language. Students will gain hands-on experience in designing, implementing, and testing AI-based solutions, including the application of machine learning techniques and the use of popular AI development tools and libraries. The ultimate goal is to enable learners to become proficient in translating AI concepts and theories into practical, working AI applications that can address real-world problems.
Differentiation Strategies *(e.g. planned activities or support for individual learners according to their needs)*	The total number of students to be in the lesson is approximately 20. This is a multicultural group of students predominantly between the ages of 24 – 45, with numerous ethnic, gender, and creed background. These are UK academic level 5 students; hence it is assumed that they have practical, theoretical, or technological knowledge and understanding of a subject or field of work to find ways forward in broadly defined, complex contexts. These students must be able to generate information, evaluate, synthesise the use information from a variety of sources. Various approaches to addressing the various identified students needs will be adopted throughout the lesson. Such will include:-

	1. Progressive tasks 2. Digital resources 3. Verbal support 4. Variable outcomes 5. Collaborative learning 6. Ongoing assessment 7. Flexible-pace learning
Equality & Diversity	Variety of teaching techniques will be employed to ensure that the needs of each individual learner are met.
Safeguarding & Prevent	Safeguarding policies and the Prevent duty are strictly observed to ensure the safety, well-being, and inclusivity of all students and staff.
Health & Safety	SIRM H&S policies will be maintained.
Learning Resources	**Teaching and Learning Materials** • "Python Crash Course" by Eric Matthes • "Automate the Boring Stuff with Python" by Al Sweigart • "Python for Data Analysis" by Wes McKinney.

Learning Outcome	Assessment Criteria
LO1. Develop proficiency in Python programming.	1.1 Write and execute basic Python programs. 1.2 Utilize Python data structures and control flow mechanisms.
LO2. Utilize Python libraries for data manipulation and analysis	2.1: Perform data manipulation using Pandas. 2.2: Conduct numerical computations using NumPy.
LO3. Implement AI and machine learning algorithms in Python	3.1: Apply machine learning algorithms using Scikit-Learn. 3.2: Create and train simple neural networks with TensorFlow or PyTorch.
LO4. Debug and optimize Python code.	4.1: Identify and correct errors in Python programs. 4.2: Optimize Python code for efficiency and performance.

No	Learning Outcome / Topic	Learning and Teaching Activities	Which assessment criteria does the session relate to?	Day/month/year/ signature
1.	**Introduction to Python**	**Introduction to Python**: Syntax, variables, and basic I/O operations.	LO1: Develop Proficiency in Python Programming	
2.	**Control Flow in Python**	**Control Flow in Python**: Conditionals (if-elif-else), loops (for, while).	LO1: Develop Proficiency in Python Programming	
3.	**Functions and Modules**	**Functions and Modules**: Defining functions, parameters, return values, and imports.	LO1: Develop Proficiency in Python Programming	
4.	**Python Data Structures**	**Python Data Structures**: Lists, tuples, dictionaries, and sets.	LO1: Develop Proficiency in Python Programming	
5.	**String Manipulation**	**String Manipulation**: String methods, formatting, and regular expressions.	LO1: Develop Proficiency in Python Programming	
6.	**File Handling**	**File Handling**: Reading/writing files (txt, csv, json).	LO1: Develop Proficiency in Python Programming	
7.	**Error Handling and OOP**	**Error Handling**: try-except blocks and exception handling. **Object-Oriented Programming (OOP)**: Classes, objects, inheritance.	LO1: Develop Proficiency in Python Programming	
8.	**Half-Term Exam**	- Review of LO1 topics - Practice questions and mock assessment - **Half-term assessment** based on LO1 (theory)		
9.	**Introduction to NumPy**	**Introduction to NumPy**: Arrays, slicing, and basic operations. **NumPy for Numerical Computing**: Matrix operations, broadcasting.	*LO2: Utilize Python Libraries for Data Manipulation & Analysis*	

10.	**Introduction to Pandas**	**Introduction to Pandas**: Series and DataFrames.	LO2: Utilize Python Libraries for Data Manipulation & Analysis	
11.	**Data Cleaning with Pandas and Data Filtering & Aggregation**	**Data Cleaning with Pandas**: Handling missing data, duplicates. **Data Filtering & Aggregation**: groupby, pivot_table, and merge.	LO2: Utilize Python Libraries for Data Manipulation & Analysis	
12.	**Matplotlib**	**Data Visualization with Matplotlib**: Line plots, bar charts, histograms.	LO2: Utilize Python Libraries for Data Manipulation & Analysis	
13.	**Seaborn**	**Advanced Visualization with Seaborn**: Heatmaps, pair plots, and custom styling.	LO2: Utilize Python Libraries for Data Manipulation & Analysis	
14.	**Final Exam Preparation & Review**	- Comprehensive review of all learning outcomes - Practice questions and revision of key topics	LO1, LO2	
15.	**Final Exam**	- **Final-term assessment** covering all learning outcomes (theory and practical elements)		
16.	**Feedback & Reflection**	- Review of final exam - Individual feedback on performance - Reflective discussion on key learning points		
17.	**Introduction to Scikit-Learn**	**Introduction to Scikit-Learn**: ML workflow (train-test split, pipelines).	*LO3: Implement AI & Machine Learning Algorithms in Python*	
18.	**Supervised Learning and Unsupervised Learning**	**Supervised Learning**: Linear regression, decision trees, and SVM. **Unsupervised Learning**: Clustering (K-Means), dimensionality reduction (PCA).	LO3: Implement AI & Machine Learning Algorithms in Python	
19.	**Model Evaluation**	**Model Evaluation**: Metrics (accuracy, precision, recall, ROC curves).	LO3: Implement AI & Machine Learning Algorithms in Python	

20.	**Neural Networks with TensorFlow/Keras**	**Neural Networks with TensorFlow/Keras**: Perceptrons, activation functions. **Building a Simple Neural Network**: MNIST classification example.	LO3: Implement AI & Machine Learning Algorithms in Python	
21.	**Deep Learning with PyTorch**	**Deep Learning with PyTorch**: Tensors, autograd, and basic NN implementation.	LO3: Implement AI & Machine Learning Algorithms in Python	
22.	**Natural Language Processing (NLP)**	**Natural Language Processing (NLP)**: Text preprocessing with NLTK/spaCy.	LO3: Implement AI & Machine Learning Algorithms in Python	
23.	**Half-Term Exam**	**Best Practices**: PEP 8, docstrings, and modular coding.	LO4: Debug & Optimize Python Code	
24.	**Debugging Techniques and Profiling Python Code**	**Debugging Techniques**: Using print(), logging, and pdb. **Profiling Python Code**: timeit, cProfile, and bottlenecks.	LO4: Debug & Optimize Python Code	
25.	**Code Optimization**	**Code Optimization**: Vectorization, avoiding loops with NumPy.	LO4: Debug & Optimize Python Code	
26.	**Memory Management**	**Memory Management**: Generators, garbage collection.	LO4: Debug & Optimize Python Code	
27.	**Parallel Computing**	**Parallel Computing**: Multithreading vs. multiprocessing.	LO4: Debug & Optimize Python Code	
28.	**Final Project**	**Final Project: End-to-end AI/Data analysis project (e.g., predictive model or data pipeline).**	LO4: Debug & Optimize Python Code	
29.	**Final Exam Preparation & Review**	LO1, LO2, LO3, LO4	LO1, LO2, LO3, LO4	
30.	**Final Exam**		LO1, LO2, LO3, LO4	