INTERNATIONAL QUALIFICATIONS

AND ASSESSMENT CENTRE (IQAC)

«Approved»

2025 y. «____» _____

Programme	**Level 6 Diploma in Artificial Intelligence**		
Unit Number/ Unit Title	UNIT 1. ADVANCED PROGRAMMING FOR DATA SCIENCE		
Cohort Code:	L06APDS-U1		
Unit Level	6		
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	This unit focuses on developing advanced programming capabilities in Python for AI and data science contexts. Learners will master data structures, object-oriented programming, performance optimization, and integration of libraries such as NumPy, Pandas, and Scikit-learn. Emphasis is also placed on writing maintainable, modular, and testable code for real-world AI applications.
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** • Lutz, M. (2013). Learning Python. O'Reilly Media. • McKinney, W. (2017). Python for Data Analysis. O'Reilly Media. • Grus, J. (2019). Data Science from Scratch. O'Reilly. • Beazley, D. (2016). Python Cookbook. O'Reilly Media.

Learning Outcome	Assessment Criteria
LO1. **1. Master advanced programming concepts in Python.**	1.1 Demonstrate use of classes, decorators, and generators. 1.2 Implement exception handling and unit testing.
LO2. **2. Apply data structures and algorithms in Python.**	2.1 Implement stacks, queues, trees, and graphs. 2.2 Analyse time and space complexity of algorithms.
LO3. **3. Utilize Python libraries for AI/data science.**	3.1 Perform data wrangling using Pandas. 3.2 Apply NumPy for vectorized operations and matrix computation.
LO4. **4. Optimize and refactor code for scalability.**	4.1 Profile and benchmark Python code. 4.2 Refactor code for maintainability and modularity.
LO5. **5. Integrate APIs and external data sources.**	5.1 Connect Python programs with RESTful APIs. 5.2 Parse and process JSON/XML data for use in AI workflows.

No	Learning Outcome / Topic	Learning and Teaching Activities	Which assessment criteria does the session relate to?	Day/month/year/ signature
1.	**Object-Oriented Programming (OOP) in Python**	**Object-Oriented Programming (OOP) in Python** Classes, inheritance, polymorphism, magic methods (__init__, __str__)	LO1: Advanced Python Programming	
2.	**Decorators & Context Managers**	**Decorators & Context Managers** @property, @staticmethod, @contextmanager	LO1: Advanced Python Programming	
3.	**Generators & Iterators**	**Generators & Iterators** yield, lazy evaluation, memory efficiency	LO1: Advanced Python Programming	
4.	**Exception Handling & Logging**	**Exception Handling & Logging** Custom exceptions, try-except-else-finally, logging module	LO1: Advanced Python Programming	
5.	**Unit Testing & Debugging**	**Unit Testing & Debugging** pytest, unittest, debugging with pdb	LO1: Advanced Python Programming	
6.	**Stacks & Queues**	**Stacks & Queues** LIFO vs. FIFO, collections.deque	LO2: Data Structures & Algorithms	
7.	**Linked Lists & Trees**	**Linked Lists & Trees** Node-based structures, binary search trees (BSTs)	LO2: Data Structures & Algorithms	
8.	Half-Term Exam	- Review of LO1 topics - Practice questions and mock assessment - **Half-term assessment** based on LO1 (theory)	LO1 LO2	
9.	**Graphs & Graph Algorithms**	**Graphs & Graph Algorithms** Adjacency lists, BFS/DFS, Dijkstra's algorithm	LO2: Data Structures & Algorithms	
10.	**Hash Tables & Dictionaries**	**Hash Tables & Dictionaries** Collision handling, dict internals	LO2: Data Structures & Algorithms	
11.	**Algorithm Complexity Analysis**	**Algorithm Complexity Analysis** Big-O notation, time/space tradeoffs	LO2: Data Structures & Algorithms	

12.	**Pandas for Data Wrangling**	**Pandas for Data Wrangling** DataFrame operations, groupby, merge, handling missing data	LO3: Python Libraries for AI/Data Science	
13.	**NumPy for Numerical Computing**	**NumPy for Numerical Computing** Vectorization, broadcasting, matrix operations	LO3: Python Libraries for AI/Data Science	
14.	Final Exam Preparation & Review	- Comprehensive review of all learning outcomes - Practice questions and revision of key topics		
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.	**Advanced Pandas Techniques**	**Advanced Pandas Techniques** Multi-indexing, apply, lambda functions	LO3: Python Libraries for AI/Data Science	
18.	**Optimizing Pandas Performance**	**Optimizing Pandas Performance** dtype optimization, eval()/query()	LO3: Python Libraries for AI/Data Science	
19.	**Working with Dates & Times**	**Working with Dates & Times** datetime, pandas.Timestamp, timezone handling	LO3: Python Libraries for AI/Data Science	
20.	**Profiling Python Code**	**Profiling Python Code** cProfile, timeit, line profilers	LO4: Code Optimization & Refactoring	
21.	**Memory Management**	**Memory Management** sys.getsizeof(), __slots__, garbage collection	LO4: Code Optimization & Refactoring	
22.	**Parallel & Concurrent Programming**	**Parallel & Concurrent Programming** multiprocessing, threading, asyncio	LO4: Code Optimization & Refactoring	
23.	Half-Term Exam	**Capstone Project** End-to-end data pipeline: API → ETL → Analysis		

24.	**Code Refactoring Best Practices**	**Code Refactoring Best Practices** DRY principles, modular design, PEP 8 compliance	LO4: Code Optimization & Refactoring	
25.	**Design Patterns for Data Science**	**Design Patterns for Data Science** Singleton, factory, observer patterns	LO4: Code Optimization & Refactoring	
26.	**Working with RESTful APIs**	**Working with RESTful APIs** requests library, authentication (OAuth, API keys)	LO5: API Integration & Data Pipelines	
27.	**Parsing JSON/XML Data**	**Parsing JSON/XML Data** json module, xml.etree.ElementTree	LO5: API Integration & Data Pipelines	
28.	**Building Data Pipelines**	**Building Data Pipelines** ETL workflows with pandas, Airflow basics **Database Integration** SQLite, SQLAlchemy, psycopg2 (PostgreSQL)	LO5: API Integration & Data Pipelines	
29.	Final Exam Preparation & Review	LO1, LO2, LO3, LO4	LO1, LO2, LO3, LO4	
30.	Final Exam		LO1, LO2, LO3, LO4	