INTERNATIONAL QUALIFICATIONS

AND ASSESSMENT CENTRE (IQAC)

«Approved»

2025 y. «__» _____

Programme	**Level 6 Diploma in Data Science**		
Unit Number/ Unit Title	UNIT 1 ADVANCED PROGRAMMING FOR DATA SCIENCE		
Cohort Code:	L06APDS-U1		
Unit Level	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 aims to equip learners with advanced programming skills required for solving complex data science problems. Emphasis is placed on writing efficient, maintainable, and scalable code using modern programming paradigms and tools. The module covers best practices in object-oriented and functional programming, version control, testing, and the use of APIs and automation frameworks to streamline data workflows.
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** • VanderPlas, J. (2016). Python Data Science Handbook. O'Reilly Media. • Grus, J. (2022). Data Science from Scratch: First Principles with Python. O'Reilly Media. • Lutz, M. (2013). Learning Python. O'Reilly Media. • Hunt, A. & Thomas, D. (1999). The Pragmatic Programmer. Addison-Wesley. • Wilson, G. (2014). Software Carpentry: Best Practices in Scientific Computing. PLOS Biology.

Learning Outcome	Assessment Criteria
LO1. **1.** Demonstrate proficiency in advanced Python programming for data manipulation and analysis.	**Practical Lab Work:** 1.1 Implement reusable code using object-oriented programming principles. 1.2 Use advanced data structures and libraries (e.g.,NumPy, Pandas, Dask).
LO2. **2.** Apply software development practices for scalable data science applications.	**Written Report:** 2.1 Develop modular code with appropriate documentation and testing. 2.2 Use version control systems like Git effectively in collaborative environments.
LO3. **3.** Integrate and manipulate data from multiple APIs and web sources.	**Project-Based Assignment:** 3.1 Build programs that connect to and process data from REST APIs 3.2 Automate data pipelines and workflows.
LO4. **4.** Utilise functional programming concepts for robust data processing.	**Portfolio Submission:** 4.1 Apply functional programming paradigms such as map-reduce and lambda functions. 4.2 Compare functional vs. imperative approaches to handling large-scale data.

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, magic methods (`__init__`, `__str__`)	LO1: Advanced Python for Data Science	
2.	**Advanced Data Structures**	**Advanced Data Structures** Namedtuples, defaultdict, custom iterators	LO1: Advanced Python for Data Science	
3.	**NumPy Mastery**	**NumPy Mastery** Vectorization, broadcasting, advanced indexing	LO1: Advanced Python for Data Science	
4.	**Pandas Optimization**	**Pandas Optimization** Efficient DataFrame operations, `eval()`/`query()`	LO1: Advanced Python for Data Science	
5.	**Parallel Computing with Dask**	**Parallel Computing with Dask** Handling out-of-core datasets, Dask DataFrames	LO1: Advanced Python for Data Science	
6.	**Modular Code Design**	**Modular Code Design** Functions, modules, packages, `__init__.py`	LO2: Software Development Practices	
7.	**Documentation Best Practices**	**Documentation Best Practices** Docstrings, Sphinx, MkDocs	LO2: Software Development Practices	
8.	Half-Term Exam	- Review of LO1 topics - Practice questions and mock assessment - **Half-term assessment** based on LO1 (theory)	LO1 LO2	
9.	**Unit Testing & Debugging**	**Unit Testing & Debugging** `pytest`, `unittest`, logging	LO2: Software Development Practices	

10.	**Git & Collaborative Workflows**	**Git & Collaborative Workflows** Feature branches, PRs, resolving merge conflicts	LO2: Software Development Practices	
11.	**CI/CD for Data Science**	**CI/CD for Data Science** GitHub Actions, model training pipelines	LO2: Software Development Practices	
12.	**Working with REST APIs**	**Working with REST APIs** `requests` library, authentication (OAuth, API keys)	LO3: API Integration & Data Pipelines	
13.	**API Rate Limiting & Caching**	**API Rate Limiting & Caching** Retry logic, `@lru_cache`	LO3: API Integration & Data Pipelines	
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.	**Web Scraping for Data**	**Web Scraping for Data** BeautifulSoup, Scrapy, ethical considerations	LO3: API Integration & Data Pipelines	
18.	**Automated Data Pipelines**	**Automated Data Pipelines** Cron jobs, Airflow DAGs	LO3: API Integration & Data Pipelines	
19.	**Data Quality Checks**	**Data Quality Checks** Great Expectations, schema validation	LO3: API Integration & Data Pipelines	
20.	**Functional Programming Concepts**	**Functional Programming Concepts** Pure functions, immutability, side effects	LO4: Functional Programming for Data	

21.	**Lambda Functions & Map-Reduce**	**Lambda Functions & Map-Reduce** `map()`, `filter()`, `reduce()`	LO4: Functional Programming for Data	
22.	**List/Dict Comprehensions**	**List/Dict Comprehensions** Nested comprehensions, memory efficiency	LO4: Functional Programming for Data	
23.	Half-Term Exam	**Project** End-to-end data application with all concepts		
24.	**Generators for Large Data**	**Generators for Large Data** `yield`, generator expressions	LO4: Functional Programming for Data	
25.	**Concurrency with** `asyncio`	**Concurrency with** `asyncio` Async API calls, event loops	LO4: Functional Programming for Data	
26.	**Performance Optimizatio**	**Performance Optimizatio** Profiling (`cProfile`), memory management	LO5: Integration & Capstone	
27.	**Error Handling Strategies**	**Error Handling Strategies** Custom exceptions, sentinel values	LO5: Integration & Capstone	
28.	**Cross-Paradigm Comparison** and **Real-World Data Pipeline**	**Cross-Paradigm Comparison** Benchmarking OOP vs. functional approaches **Real-World Data Pipeline** API → ETL → Analysis → Visualization	LO5: Integration & Capstone	
29.	Final Exam Preparation & Review	LO1, LO2, LO3, LO4	LO1, LO2, LO3, LO4	
30.	Final Exam		LO1, LO2, LO3, LO4	