Level 5

Data Engineer

Higher Apprenticeship

Programming and Scripting Essentials Module Handbook
2024

MODULE OVERVIEW

Programming and Scripting Essentials

Module	Phase	Weekly Topics	Description
Programming and Scripting Essentials	?	• **Linux for data engineers:** Learn what the Linux system is and why you should use it, how to navigate the Linux filesystem, become familiar with key utilities for data engineers, learn how to complete job scheduling and automation, and advanced text processing and log management. • **Version control:** Learn about DevOps principles and their use in data engineering for improved collaboration, automation, and monitoring. We will also unpack version control, Continuous Integration/Continuous Deployment (CI/CD) processes for efficient data product delivery, and finally, how to use GitHub for code reviews to maintain quality and teamwork. • **Python for data engineers:** Gain a solid understanding of Python and its application in data engineering. You will be equipped with the skills to write efficient Python code, develop data products, and understand the importance of test-driven development. • **Data structures and OOP:** Learn about data serialisation for easy data storage and transmission, coding conventions for clean and maintainable code, and Object-Oriented Programming (OOP) for structured	This module is designed to make Apprentices comfortable with programming, scripting and the software development practices in the modern data-oriented enterprise. Learners will script, code and conduct practical code reviews. There is an emphasis on practicing skills throughout this module and you will be applying the skills acquired to a practical work-based scenario, to develop a portfolio of evidence for the professional discussion element of the endpoint assessment. Focusing on Big Data programming, you will learn Linux, job scheduling, version control, Git, Python, data manipulation with pandas, parallel programming with Spark, algorithmic thinking, data visualisation, Docker, regular expressions, unit testing, the essentials of data pipelines, documenting, error handling and logging. This module contains one full-day Hackathon (on one of the regular teaching days).

		programming. This topic will also introduce you to design patterns as solutions to common software design problems. • **Algorithmic thinking:** Gain a solid foundation in algorithmic thinking and its applications, break down complex algorithmic problems, design efficient algorithms, and bring algorithmic solutions to life with code. • **Parallel programming:** This topic will cover the application of concurrency, parallelism, and distributed computing in Python. The topic also explores the benefits of using Spark for parallelism and evaluates other comparable platforms, enhancing your practical capabilities for handling large-scale data processing tasks. • **Spark for data engineers:** The aim of this topic is to enable to evaluate the use of Spark clusters for data processing, understand the essential features of data pipelines, and construct your own using SparkSQL and Spark Streaming. • **Practical programming:** The focus of this module is the real-world applications and problem-solving with programming. It will cover the building robust data pipelines, ensuring their security, utilising data visualisation for maximum impact, and writing effective test suites.	

Outcomes

After finishing this module, you will be able to meet the following outcomes and KSBs:

- **Employ software development tools and techniques for designing, deploying and maintaining secure data products and pipelines, including debugging, version control and testing.**
- **K3, K6, K17, K20, K26**
- **S9, S12, S13, S14, S17, S20, S25, S27**
- **B1, B5, B6**
- **Construct algorithms that correctly and efficiently handle data at scale whilst mitigating risks.**
- **K2, K5, K24, K25, K27**
- **S1, S2, S3, S5, S16, S17, S24, S26**
- **B2, B4**
- **Demonstrate the knowledge of the steps needed to prepare the code for production.**
- **K6, K8, K28**
- **S2, S4, S8, S14, S17, S20, S26**
- **B1, B3**

Introduction to the Module

This module, "Programming and Scripting Essentials," spanning months three to four, is designed to equip you with foundational and advanced skills necessary for a data engineering role, focusing particularly on the software development practices pivotal in a data-centric business environment.

Over the course of this module, you will delve into various aspects of data engineering, including the utilisation of Linux for Big Data applications, mastery of version control with Git, and software development techniques in Python. Key topics such as data structures, object-oriented programming, data manipulation via APIs and the Pandas library, as well as algorithmic thinking with an emphasis on searching, sorting, and machine learning algorithms, are comprehensively covered.

Moreover, you will explore parallel programming concepts using Databricks and PySpark, and will gain practical experience in Spark for constructing efficient data pipelines. The curriculum also includes training on Docker for environment management, data visualisation techniques, regular expressions, and rigorous testing methods to ensure code reliability and security.

A unique feature of this module is a full-day Hackathon, which integrates regular teaching with hands-on coding challenges, thereby reinforcing the learning through practical application. This format is especially designed to prepare you for real-world data engineering tasks and culminates in the development of a professional portfolio.

Upon completion, you will possess the ability to design, deploy, and maintain secure data products and pipelines, construct robust algorithms suitable for handling large-scale data, and prepare code for production with a keen awareness of operational requirements. This will prepare you not only to contribute effectively to your current role but also to meet the stringent requirements of professional competency assessments in their future careers.

Mode of Assessment

END POINT ASSESSMENT (EPA)

It is **important** to read all the assessment guide documents contained in the Programme Handbook, as they contain important details.

Reminder: Refer to the programme handbook for further guidance.

Core Reading:

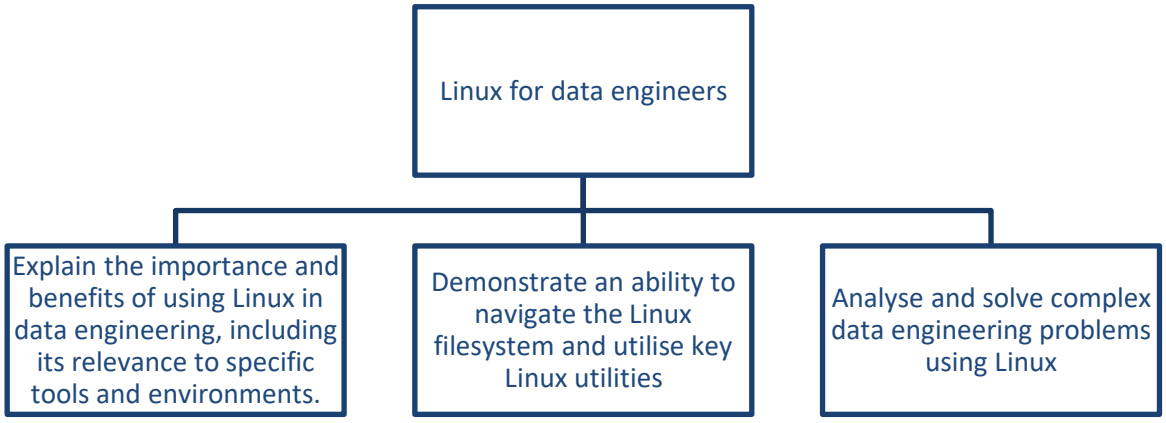
- Programme Handbook: Students will be provided with a single handbook by BPP which will summarise the key expectations for each topic in the course. The handbook is reviewed annually to include the latest and most appropriate academic resources.
- Haskell, C. (2024). Driving Data Projects: A comprehensive guide. British Computer Society [20 hours]
- King, T., Schwarzenbach, J., (2020). Managing Data Quality: A practical guide. British Computer Society [15 hours]
- Reis, J., Housley, M. (2022), Fundamentals of Data Engineering: Plan and Build Robust Data Systems, O'Reilly [25 hours]

TOPIC BREAKDOWN

Topic 1 – Linux for data engineers

Topic Learning Outcomes

As a step towards build your skills towards the final module assessment, the learning objectives for this topic are:



Introduction

In the rapidly evolving field of data engineering, one constant remains: the ubiquitous presence of Linux. As data engineers, we often find ourselves interacting with Linux-based systems, whether it's managing databases, orchestrating data pipelines, or deploying machine learning models. However, a common challenge that many data engineers face, especially those at level 5, is the lack of comprehensive understanding and proficiency in Linux. This gap in knowledge can lead to inefficiencies and roadblocks in their workflow, hindering their ability to fully leverage the power of their data infrastructure.

Understanding the relevance of Linux for data engineers is crucial. Linux offers a robust and flexible environment that is ideal for handling large datasets and complex computations. It provides a wide range of tools and utilities that can be used to manipulate and process data efficiently. Moreover, Linux's open-source nature allows for high customisability, making it adaptable to various data engineering needs. For level 5 data engineering students, mastering Linux can significantly enhance their skill set, making them more versatile and effective in their roles.

Consider the case of a real-world data engineering project at a leading tech company. The data engineering team was tasked with setting up a reliable and scalable data pipeline to handle the company's growing data needs. The team chose to use a Linux-based system due to its stability, performance, and the wide array of data processing tools available. However, the team's limited familiarity with Linux led to bottlenecks in the project, with tasks taking longer than necessary due to the learning curve associated with the Linux environment. This case underscores the importance of having a solid understanding of Linux in data engineering roles.

By the end of this topic, you will have gained a deep understanding of Linux from a data engineering perspective. You will learn how to navigate the Linux filesystem, use key command-line tools, write shell scripts to automate tasks, manage processes and resources, and secure a Linux system. These skills will not only make you a more efficient data engineer but also open up new opportunities for you. With a strong foundation in Linux, you will be able to design and manage more robust data infrastructures, contribute to open-source data projects, and stay ahead in the ever-evolving field of data engineering.

Structure

Topics for this programme follow a Prepare-Collaborate-Apply structure:

Prepare

This is the stage where you build the knowledge to underpin your learning. This might involve completing interactive e-learning packages, watching videos, or working through reading materials.

It is essential that you make the most of the learning materials provided before attending webinars, as this will allow you to test your knowledge and stretch you understanding further.

Collaborate

This is where you will receive guidance from our expert tutors and coaches to shape and refine your understanding through in-depth explanation, discussion, testing and carrying out more advanced practical and realistic tasks. This also helps to develop valuable team-working skills.

Apply

You now apply the knowledge you have developed to real-world tasks.

This stage is all about ensuring you truly grasp and retain what you've learned. Through completion of off-the-job (OTJ) revision tasks and tests, you'll get plenty of practice applying your knowledge. Plan to dedicate 6-8 hours each week to guided study and portfolio work, with sessions typically on the same day each week.

Task 1 brief: Setting up and getting started with NDG Linux Unhatched

NDG Linux Unhatched allows students to wade into the shallow end of Linux, the back-end operating system used by global titans such as Facebook, Google, Microsoft, NASA, Tesla, Amazon and more.

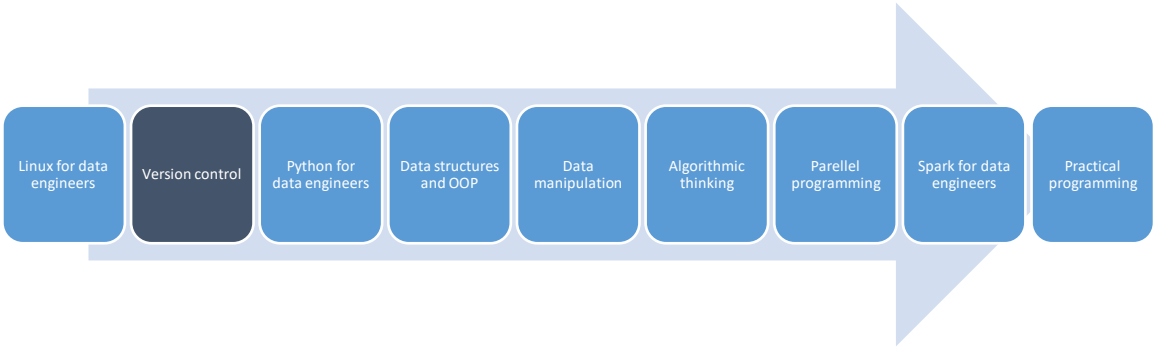
The aim of this task is to not only enable you to learn new concepts but also apply them practically, setting a strong foundation for your future learning.

To ensure you don't get stuck, you will be guided step-by-step through a series of hands-on virtual machine activities.

Further guidance can be found here: L5DE 3.1 Apply (OTJ)

Link

This topic is one of 9 topics for this Module.



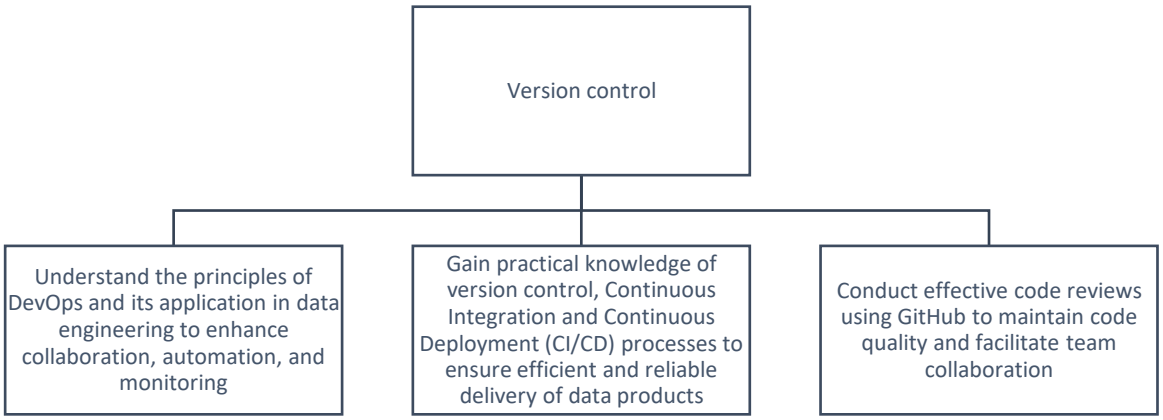
The sequence of topics in this module is carefully designed so that your knowledge and skills will develop as you progress.

The next topic is **Version control**.

Topic 2 – Version control

Topic Learning Outcomes

As a step towards build your skills towards the final module assessment, the learning objectives for this topic are:



Introduction

In the rapidly evolving world of technology, the ability to manage and track changes in code is a critical skill. This is where the concept of 'Version Control' comes into play. Version control systems are a category of software tools that help a software team manage changes to source code over time. They allow you to keep track of what was changed, when it was changed, and who changed it.

For Level 5 data engineering students, understanding and implementing version control is particularly relevant. As data engineers, you will be working with large volumes of data and complex algorithms. Changes and improvements to these algorithms are a constant part of the job. Without a robust version control system in place, managing these changes can become a daunting task.

Consider the real-world case of a multinational corporation with a global team of data engineers working on the same project. Without version control, coordinating tasks, tracking changes, and maintaining the integrity of the project would be nearly impossible. However, with a version control system like Git, changes from multiple sources can be seamlessly integrated, tracked, and even rolled back if necessary.

By the end of this course, you will have gained a deep understanding of version control systems, particularly Git. You will learn how to track changes, resolve conflicts, and manage branches in a collaborative coding environment. These skills are not just limited to your academic projects; they are highly sought after in the professional world. Whether you're working in a small startup or a large corporation, version control is an essential part of the software development lifecycle. Mastering it will give you a significant edge in your future career as a data engineer.

Structure

Topics for this programme follow a Prepare-Collaborate-Apply structure:

Prepare

This is the stage where you build the knowledge to underpin your learning. This might involve completing interactive e-learning packages, watching videos, or working through reading materials.

It is essential that you make the most of the learning materials provided before attending webinars, as this will allow you to test your knowledge and stretch you understanding further.

Collaborate

This is where you interact with our expert tutors and coaches to shape and refine your understanding through discussion, testing and carrying out more advanced practical and realistic tasks. This also helps to develop valuable team-working skills.

Apply

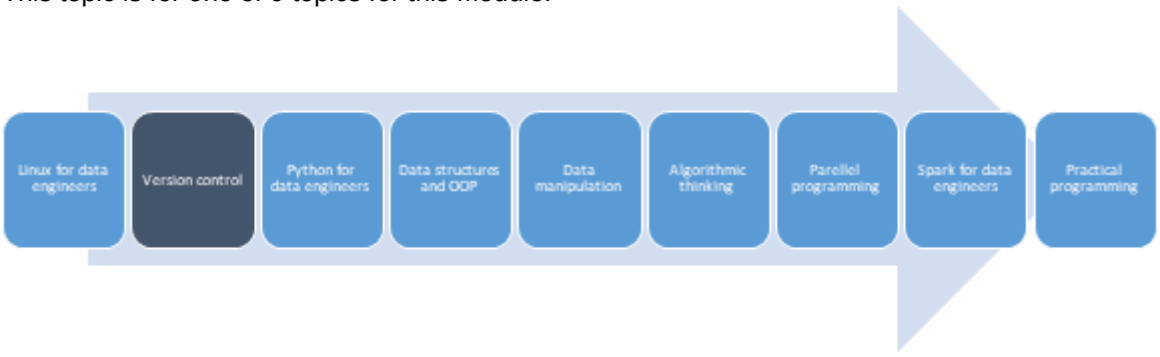
You now apply the knowledge you have developed to real-world tasks, including off-the-job (OTJ) and on-the-job (OJT).

Off-the-job learning tasks

This stage is all about ensuring you truly grasp and retain what you've learned. Through completion of off-the-job (OTJ) revision tasks and tests, you'll get plenty of practice applying your knowledge. Plan to dedicate 6-8 hours each week to guided study and portfolio work, with sessions typically on the same day each week.

Link

This topic is for one of 9 topics for this Module.



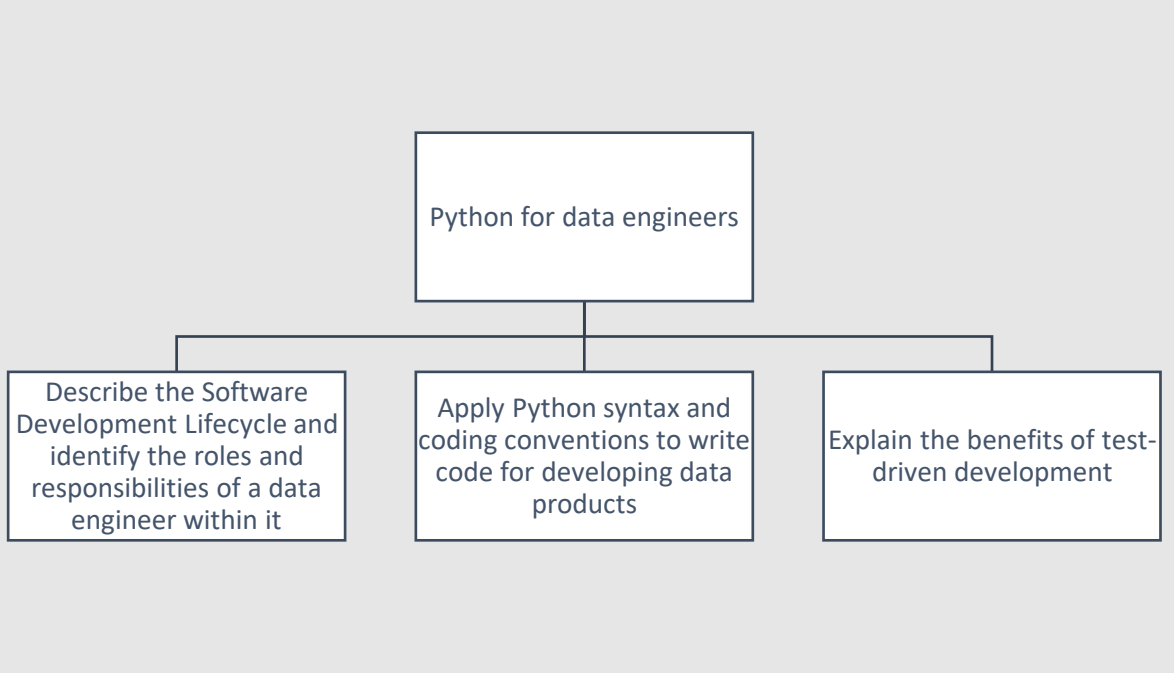
The sequence of topics in this module is carefully designed so that your knowledge and skills will develop as you progress.

The next topic is **Python for data engineers**.

Topic 3 – Python for data engineers

Topic Learning Outcomes

As a step towards build your skills towards the final module assessment, the learning objectives for this topic are:



Introduction

This topic is designed to equip you with the advanced knowledge and practical skills essential for thriving in today's data-driven landscape.

As a data engineer, your role is pivotal in shaping the data infrastructure that powers modern businesses. Python, being a versatile and powerful programming language, stands at the forefront of data engineering tools. This topic will not only enhance your Python proficiency but also contextualise its application within the broader framework of data engineering practices.

Throughout the topic, you'll work on projects that mirror actual scenarios encountered in the field. This application-oriented learning will bridge the gap between theory and practice, preparing you to solve real-world problems effectively.

Upon completion of this topic, you'll have:

- A deep understanding of the SDLC and your role within it
- Enhanced Python programming skills tailored for data engineering tasks
- Proficiency in test-driven development methodologies
- The ability to develop reliable and efficient data products
- Strategic insights to elevate your contributions within any organisation

This topic is more than just technical training; it's a comprehensive preparation for the challenges and opportunities in the evolving field of data engineering. Whether you're looking to advance your career or increase your value within your current role, this topic will empower you to become a more effective, efficient, and innovative data engineer.

Structure

Topics for this programme follow a Prepare-Collaborate-Apply structure:

Prepare

This is the stage where you build the knowledge to underpin your learning. This might involve completing interactive e-learning packages, watching videos, or working through reading materials.

It is essential that you make the most of the learning materials provided before attending webinars, as this will allow you to test your knowledge and stretch you understanding further.

Collaborate

This is where you interact with our expert tutors and coaches to shape and refine your understanding through discussion, testing and carrying out more advanced practical and realistic tasks. This also helps to develop valuable team-working skills.

Apply

You now apply the knowledge you have developed to real-world tasks.

Off-the-job learning tasks

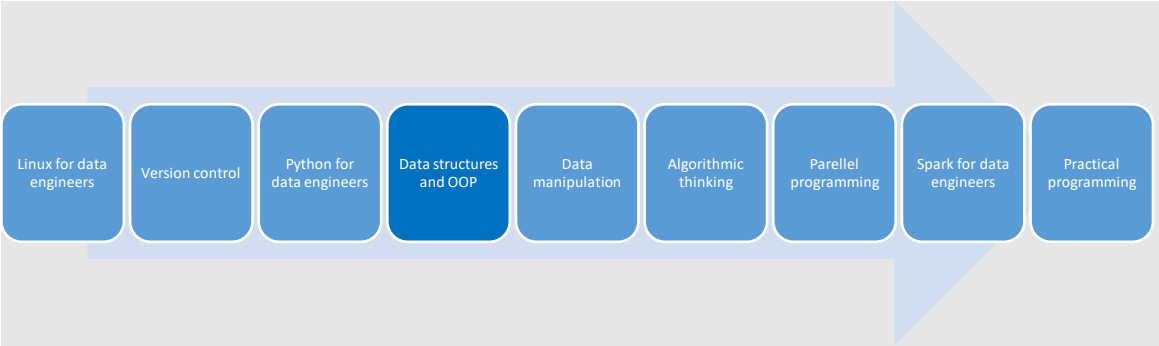
This stage is all about ensuring you truly grasp and retain what you've learned. Through completion of off-the-job (OTJ) revision tasks and tests, you'll get plenty of practice applying your knowledge. Plan to dedicate 6-8 hours each week to guided study and portfolio work, with sessions typically on the same day each week.

Task 1 brief:

Your tutor will provide you with a brief on how to complete the Python notebook for this topic.

Link

This handbook is for one of 9 topics for this Module.



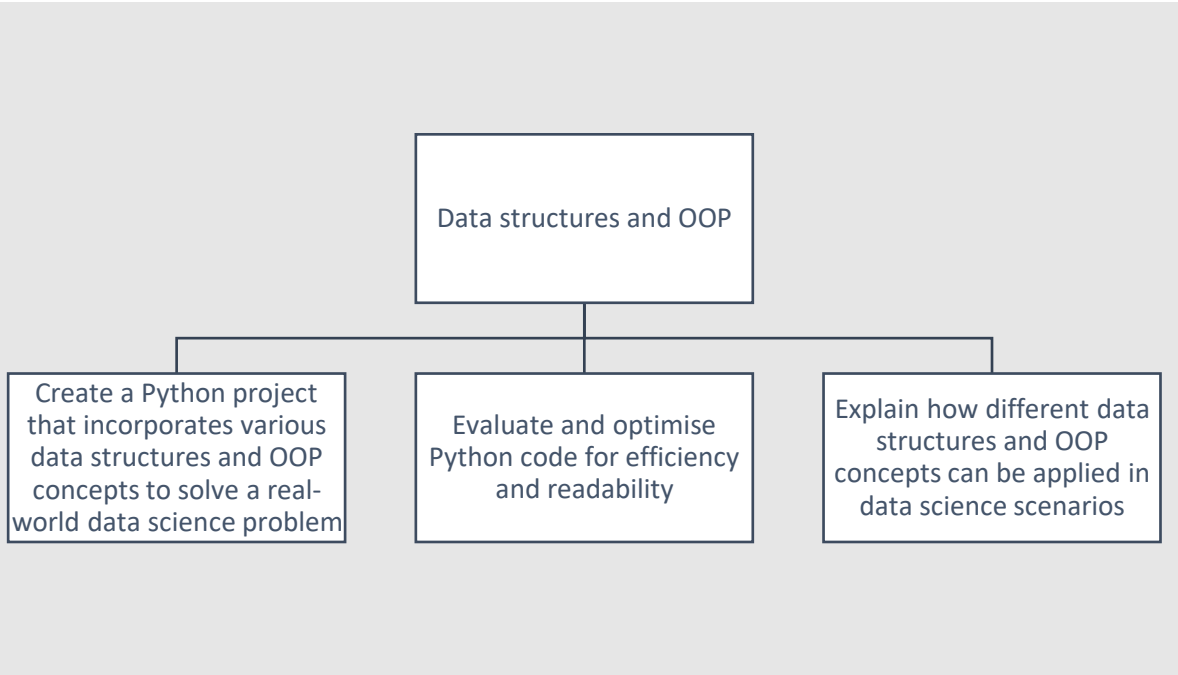
The sequence of topics in this module is carefully designed so that your knowledge and skills will develop as you progress.

The next topic is **Data structures and OOP**.

Topic 4 – Data structures and OOP

Topic Learning Outcomes

As a step towards build your skills towards the final module assessment, the learning objectives for this topic are:



Introduction

In today's data-driven world, organisations grapple with ever-increasing volumes of data, diverse data types, and the need for real-time processing. As a data engineer, you're tasked with creating robust data pipelines and systems to handle these demands efficiently. Without a deep understanding of data structures and OOP principles, it's easy to create solutions that are difficult to scale, maintain, or adapt to changing business needs.

Consider a global e-commerce platform processing millions of transactions daily. The company needs a system to ingest and process transaction data in real-time, provide instant product recommendations, analyse sales trends across regions and categories, and ensure data integrity and security throughout the pipeline. This scenario exemplifies why advanced knowledge of data structures and OOP is crucial for data engineers.

By the end of this topic, you will have gained a deep understanding of Python data structures, OOP concepts, and their application in data engineering. You'll comprehend best practices for code organisation and data serialisation, and understand how to design scalable, maintainable data systems. Your skills will include selecting and implementing appropriate data structures for various tasks, designing object-oriented solutions for complex data problems, writing clean and efficient Python code, and applying OOP principles to create flexible, reusable data pipeline components.

Through a blend of theoretical learning and hands-on practice, this topic will enable you to develop the expertise to tackle real-world data engineering challenges. You'll learn to create data systems that are not just functional, but also efficient, scalable, and adaptable to evolving business needs. This knowledge will equip you to design sophisticated data architectures, optimise processing workflows, and contribute more effectively to your organisation's data strategy. Whether you're working on big data processing, building ETL pipelines, or developing data APIs, the insights gained from this topic will be invaluable in your role as a data engineer.

Structure

Topics for this programme follow a Prepare-Collaborate-Apply structure:

Prepare

This is the stage where you build the knowledge to underpin your learning. This might involve completing interactive e-learning packages, watching videos, or working through reading materials.

It is essential that you make the most of the learning materials provided before attending webinars, as this will allow you to test your knowledge and stretch you understanding further.

This e-learning for this topic covers a comprehensive range of topics including visual modelling and diagramming techniques (UML, ERD, custom diagrams), key architectural patterns and principles (layered architecture, microservices), designing data products aligned with business objectives, data governance, and architectural frameworks like TOGAF.

Collaborate

This is where you interact with our expert tutors and coaches to shape and refine your understanding through discussion, testing and carrying out more advanced practical and realistic tasks. This also helps to develop valuable team-working skills.

Apply

You now apply the knowledge you have developed to real-world tasks.

Off-the-job learning tasks

This stage is all about ensuring you truly grasp and retain what you've learned. Through completion of off-the-job (OTJ) revision tasks and tests, you'll get plenty of practice applying your knowledge. Plan to dedicate 6-8 hours each week to guided study and portfolio work, with sessions typically on the same day each week.

Task 1 brief: Introduction to debugging

In this task, you will explore the critical skill of debugging in software development, including:

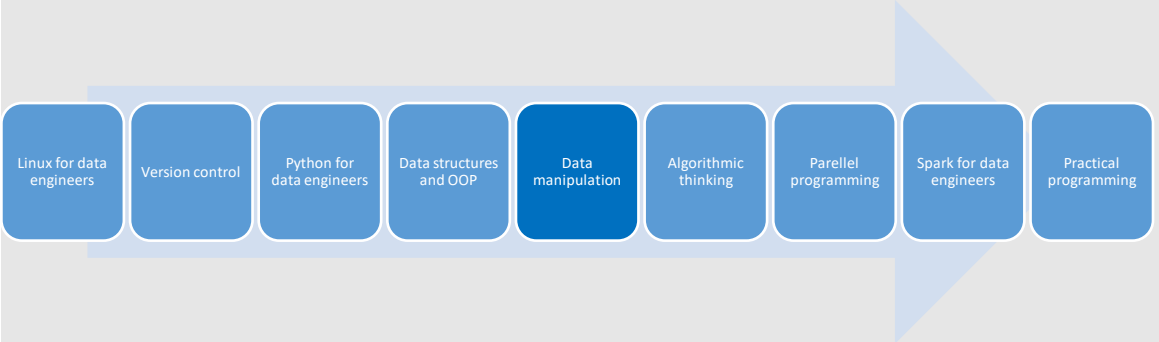
1. Importance of Debugging: You'll learn why debugging matters. It's about identifying and fixing errors in your code. Think of it as detective work for programmers!
2. Python Debugging Fundamentals:
    - Print Statements: You'll use print statements to track what's happening in your code. It's like leaving breadcrumbs to follow your program's journey.
    - Python Debugger (pdb): This powerful tool lets you pause your program, inspect variables, and figure out what's going wrong step by step.
    - Exception Handling: You'll catch and handle errors gracefully, preventing your program from crashing.
    - Logging: Imagine keeping a journal of your program's adventures. Logging helps you record messages about what's going on.

3. Activity: You'll tackle a practical challenge. There's a function called stripped_reversed_lowercase with a bug. Your mission? Debug it! Use the debugger, print variables, and solve the mystery.

Further guidance can be found here: Apply e-learning: Introduction to debugging

Link

This topic is for one of 9 topics for this Module.



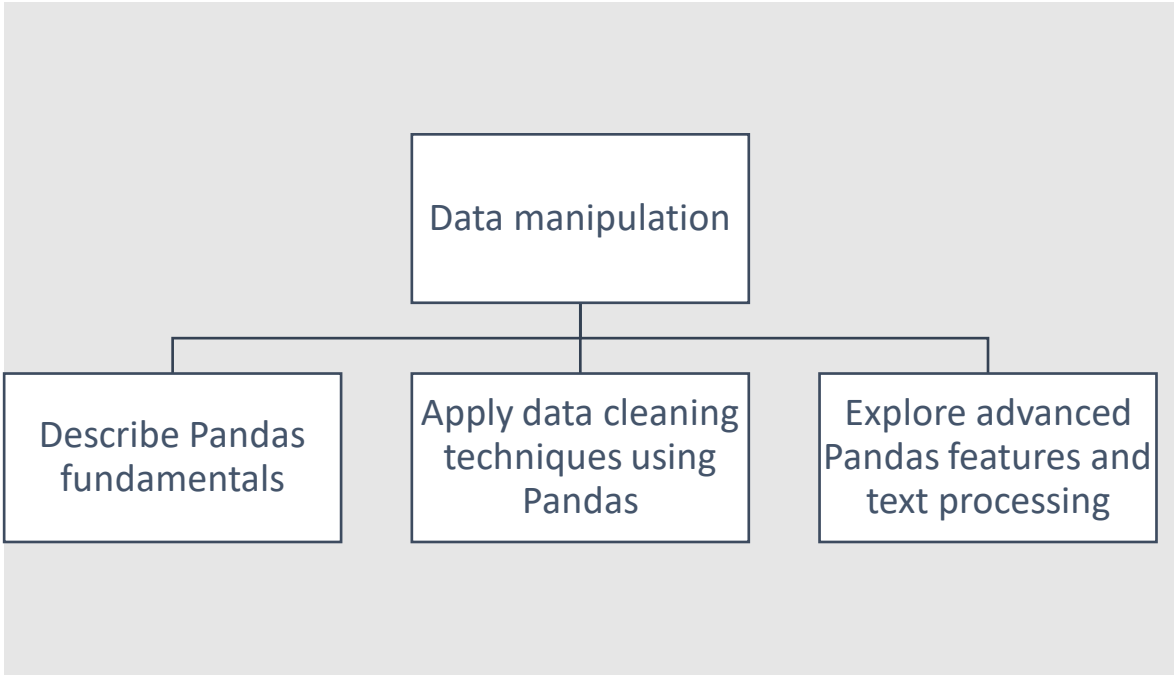
The sequence of topics in this module is carefully designed so that your knowledge and skills will develop as you progress.

The next topic is **Data manipulation**.

Topic 5 – Data manipulation

Topic Learning Outcomes

As a step towards build your skills towards the final module assessment, the learning objectives for this topic are:



Introduction

In the modern era, we are surrounded by an overwhelming amount of information. However, the true value of this information does not lie in its sheer volume, but in how we utilise it. Consider a vast library filled with countless books. Without the ability to organise, search, or extract wisdom from these books, the library's potential remains untapped. This is where data manipulation comes into play, serving as a powerful tool that allows us to transform raw data into actionable insights.

As individuals aspiring to become data engineers, we are faced with the challenge of dealing with a wide variety of data. This data, arriving in diverse formats such as spreadsheets, databases, logs, APIs, and more, can often seem like an unruly beast that needs to be tamed.

For instance, imagine a large retail company inundated with sales records. These records, amounting to millions of rows, are scattered across various files. The challenge lies in extracting meaningful trends and identifying top-selling products from this chaotic sea of information.

Data manipulation is crucial in these scenarios as it helps unlock hidden patterns within raw data. By cleaning, transforming, and aggregating data, we can unearth valuable insights that were previously hidden. These insights fuel decision-making processes within organisations, whether it's optimising supply chains, predicting stock prices, or tailoring marketing campaigns. Moreover, mastering data manipulation can accelerate one's career, as employers are always on the lookout for data engineers who can skilfully handle data. This mastery can open doors to exciting roles across various sectors such as tech, finance, healthcare, and more.

In terms of skills, this topic will focus on Python, SQL, and Pandas. We will use Python to craft scripts that can reshape data, filter rows, and merge tables seamlessly. SQL, or Structured Query Language, will be our tool for querying databases, joining tables, and extracting insights. Pandas, a versatile tool, will be used to slice, dice, and transform data effortlessly.

In conclusion, data manipulation is not just about writing code. It's about unravelling the stories hidden within the intricate tapestry of data. So, let's grab our keyboards, put on our data capes, and dive into the fascinating world of data manipulation.

Structure

Topics for this programme follow a Prepare-Collaborate-Apply structure:

Prepare

This is the stage where you build the knowledge to underpin your learning. This might involve completing interactive e-learning packages, watching videos, or working through reading materials.

It is essential that you make the most of the learning materials provided before attending webinars, as this will allow you to test your knowledge and stretch you understanding further.

Collaborate

This is where you will receive guidance from our expert tutors and coaches to shape and refine your understanding through in-depth explanation, discussion, testing and carrying out more advanced practical and realistic tasks. This also helps to develop valuable team-working skills.

Apply

You now apply the knowledge you have developed to real-world tasks.

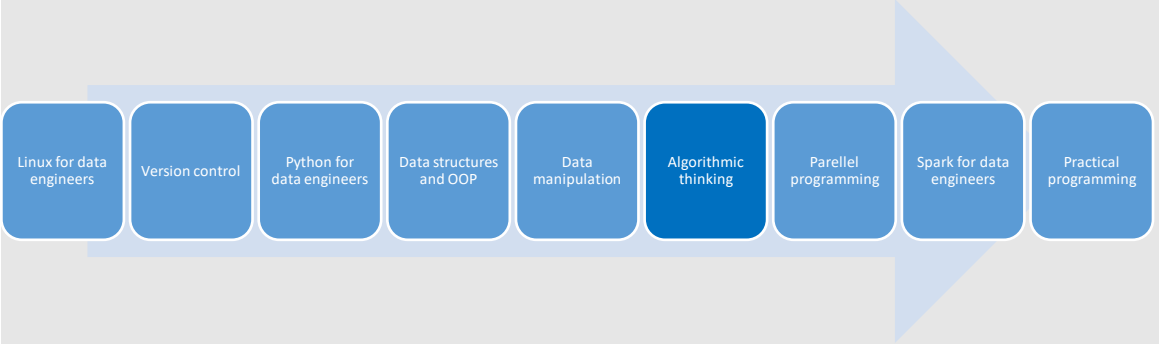
Off-the-job learning tasks

This stage is all about ensuring you truly grasp and retain what you've learned. Through completion of off-the-job (OTJ) revision tasks and tests, you'll get plenty of practice applying your knowledge. Plan to dedicate 6-8 hours each week to guided study and portfolio work, with sessions typically on the same day each week.

Task 1 brief: Your tutor will provide you with a brief on how to complete the Python notebook for this topic.

Link

This handbook is for one of 9 topics for this Module.



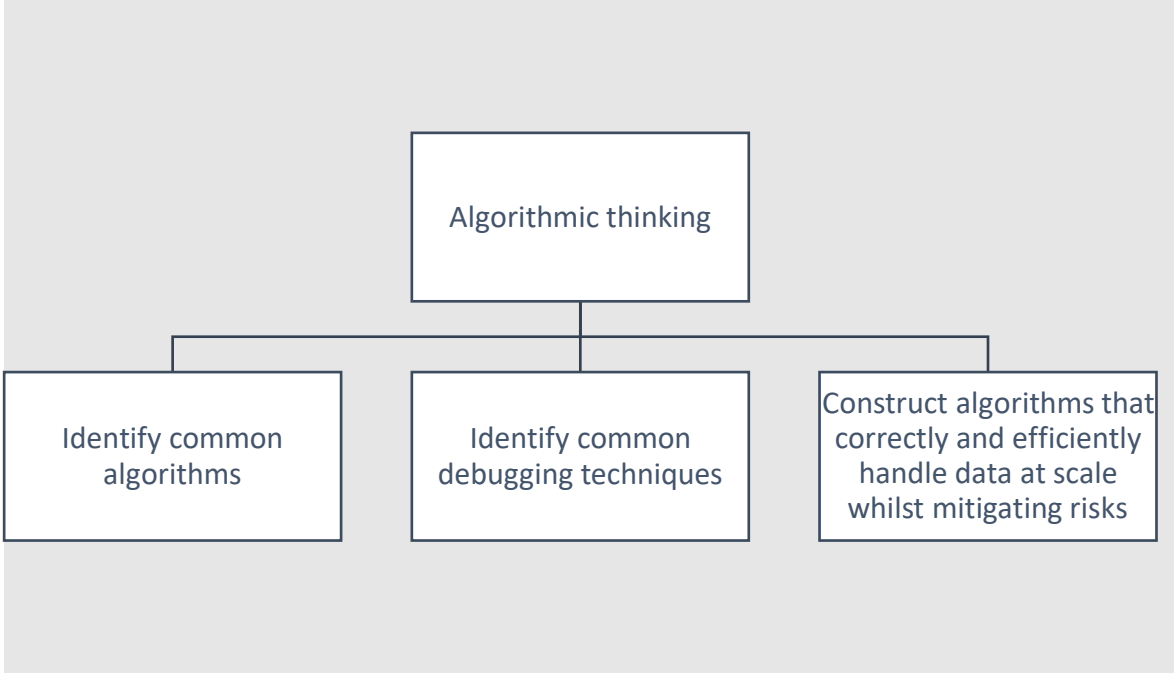
The sequence of topics in this module is carefully designed so that your knowledge and skills will develop as you progress.

The next topic is **Parallel programming.**

Topic 6 – Parallel programming

Topic Learning Outcomes

As a step towards build your skills towards the final module assessment, the learning objectives for this topic are:



Introduction

Algorithmic thinking is a fundamental skill that will empower you to design efficient, scalable, and robust solutions for complex data processing challenges. This critical ability will set you apart in the

field of data engineering, where the ability to conceptualise and implement effective algorithms is paramount.

In the realm of big data, traditional approaches to data processing often fall short when faced with massive datasets, real-time processing requirements, and complex data relationships. As a data engineer, you'll be tasked with creating data pipelines and systems that can handle these demands efficiently and accurately. Without strong algorithmic thinking skills, you may struggle to optimise performance, manage resource utilisation, and ensure data integrity across large-scale systems. The challenges you'll face require not just technical knowledge, but also the ability to think critically about how to structure and process data in the most effective way possible.

To illustrate the real-world application of algorithmic thinking, consider a leading streaming service like Netflix. They process enormous amounts of user data to provide personalised content recommendations, manage video streaming quality, and predict viewing trends. Their recommendation algorithm alone analyses billions of records, considering factors like viewing history, ratings, time of day, and device type. This system must work in near real-time, scale to millions of users, and adapt to constantly changing content and user behaviours. The complexity of this task underscores the importance of sophisticated algorithmic thinking in modern data engineering.

By the end of this topic, you will have developed a deep understanding of algorithmic complexity and efficiency, a comprehensive grasp of common algorithmic patterns and their applications in data engineering, insight into optimisation techniques for data-intensive algorithms, and an understanding of how to analyse and improve algorithm performance. These knowledge areas form the foundation of effective algorithmic thinking in data engineering.

Structure

Topics for this programme follow a Prepare-Collaborate-Apply structure:

Prepare

This is the stage where you build the knowledge to underpin your learning. This might involve completing interactive e-learning packages, watching videos, or working through reading materials.

It is essential that you make the most of the learning materials provided before attending webinars, as this will allow you to test your knowledge and stretch you understanding further.

The e-learning for this topic covers building a data-driven culture, fundamentals of data types and sizes, standards and best practices in data engineering, working with different data sources and types through the data management lifecycle, and utilisation of visualisation tools.

Collaborate

This is where you will receive guidance from our expert tutors and coaches to shape and refine your understanding through in-depth explanation, discussion, testing and carrying out more advanced practical and realistic tasks. This also helps to develop valuable team-working skills.

The webinar for this topic will focus on building a data-driven culture, data fundamentals like types/sizes, key standards and best practices in data engineering, a practical lab, and summarised the core learning concepts.

Apply

You now apply the knowledge you have developed to real-world tasks.

Off-the-job learning tasks

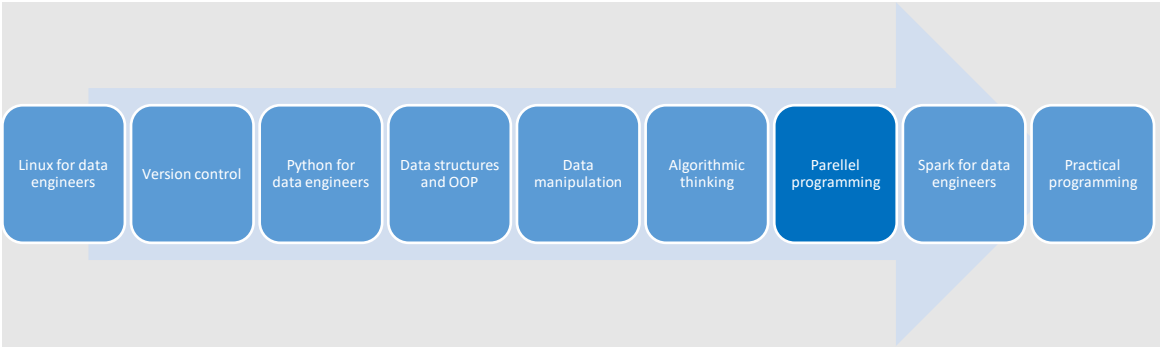
This stage is all about ensuring you truly grasp and retain what you've learned. Through completion of off-the-job (OTJ) revision tasks and tests, you'll get plenty of practice applying your knowledge. Plan to dedicate 6-8 hours each week to guided study and portfolio work, with sessions typically on the same day each week.

Task 1 brief: Your tutor will provide you with a brief on how to complete the Python notebook for this topic.

Further guidance can be found here:

Link

This handbook is for one of 9 topics for this Module.



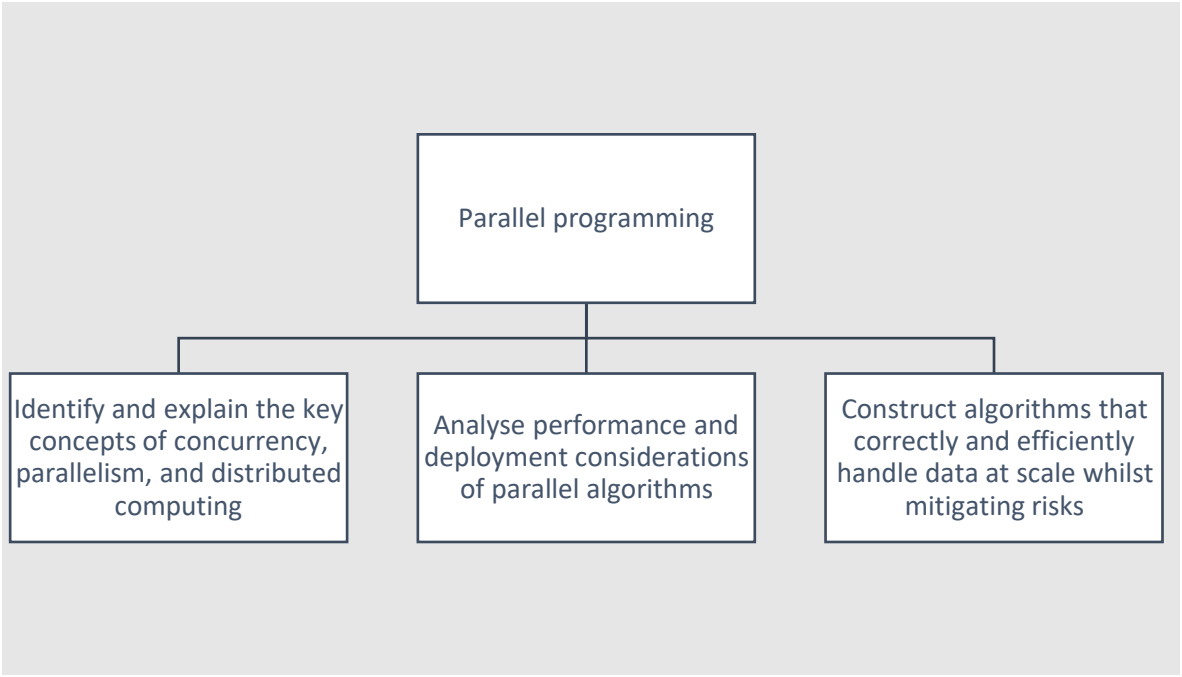
The sequence of topics in this module is carefully designed so that your knowledge and skills will develop as you progress.

The next topic is **Parallel programming**.

Topic 7 – Parallel programming

Topic Learning Outcomes

As a step towards build your skills towards the final module assessment, the learning objectives for this topic are:


Parallel programming
Identify and explain the key concepts of concurrency, parallelism, and distributed computing
Analyse performance and deployment considerations of parallel algorithms
Construct algorithms that correctly and efficiently handle data at scale whilst mitigating risks

Introduction

Parallel programming is a crucial skill in the era of big data and high-performance computing. This topic will equip you with the ability to identify and explain key concepts of concurrency, parallelism, and distributed computing, and their application within Python programming, a language known for its simplicity and power. We'll delve deep into these foundational concepts, exploring how they differ and complement each other. You'll learn about threads, processes, synchronization mechanisms, and distributed systems architecture. Understanding these concepts is essential for leveraging the full power of modern multi-core processors and distributed computing environments.

We will delve into the performance and deployment considerations of parallel algorithms, empowering you to make informed decisions when implementing these in real-world scenarios. This focus on practical aspects of implementing parallel solutions will include examining various metrics for measuring performance gains, such as speedup and efficiency. You'll learn to identify bottlenecks, optimize communication overhead, and balance workloads across multiple processors or nodes. We'll also discuss deployment strategies for different parallel computing platforms, including clusters, grids, and cloud environments.

Additionally, you will explore the advantages of Spark for parallelism and assess other similar platforms, expanding your toolkit for managing large-scale data processing tasks. We'll dive into Spark's distributed computing model, its integration with Python through PySpark, and compare it with other big data processing frameworks. This knowledge will be crucial as we address the challenges of designing parallel algorithms that not only work correctly but also scale efficiently with increasing data sizes and computing resources.

Throughout this topic, we'll explore parallel algorithm design patterns, data partitioning strategies, and techniques for ensuring correctness in the face of concurrency. We'll also discuss common pitfalls and risks in parallel programming, such as race conditions, deadlocks, and data inconsistencies, and strategies to mitigate them. By mastering these concepts and skills, you'll be well-equipped to construct algorithms that correctly and efficiently handle data at scale while mitigating risks.

Acquiring this knowledge and these skills will not only make you a more versatile and effective data engineer but also pave the way for new opportunities in your future career. You'll be prepared to tackle complex computational problems efficiently, handle large-scale data processing tasks, and adapt to the ever-evolving landscape of parallel and distributed computing. This expertise will enhance your capabilities in fields such as scientific computing, machine learning, and cloud-based applications, positioning you at the forefront of modern computing practices.

Structure

Topics for this programme follow a Prepare-Collaborate-Apply structure:

Prepare

This is the stage where you build the knowledge to underpin your learning. This might involve completing interactive e-learning packages, watching videos, or working through reading materials.

It is essential that you make the most of the learning materials provided before attending webinars, as this will allow you to test your knowledge and stretch you understanding further.

Collaborate

This is where you will receive guidance from our expert tutors and coaches to shape and refine your understanding through in-depth explanation, discussion, testing and carrying out more advanced practical and realistic tasks. This also helps to develop valuable team-working skills.

The webinar for this topic will focus on building a data-driven culture, data fundamentals like types/sizes, key standards and best practices in data engineering, a practical lab, and summarised the core learning concepts.

Apply

You now apply the knowledge you have developed to real-world tasks.

Off-the-job learning tasks

This stage is all about ensuring you truly grasp and retain what you've learned. Through completion of off-the-job (OTJ) revision tasks and tests, you'll get plenty of practice applying your knowledge. Plan to dedicate 6-8 hours each week to guided study and portfolio work, with sessions typically on the same day each week.

Task 1 brief:

This task involves setting up and executing a series of PySpark operations in Microsoft Azure Databricks. The steps include:

Setup: Log into Microsoft Azure using the Github Student Pack, select Azure Databricks, create a new deployment using the free trial tier, create a new cluster, and create a new notebook.

Cell 1 - Initialise PySpark: Set up the PySpark environment and initialize it.

Cell 2 - Load and Collect Data: Upload a file named “Example.csv” to Databricks, load the text file using the SparkContext, split the lines into a list, and collect the dataset into local RAM. You’re also asked to research and write about the pros and cons of running collect() in Spark.

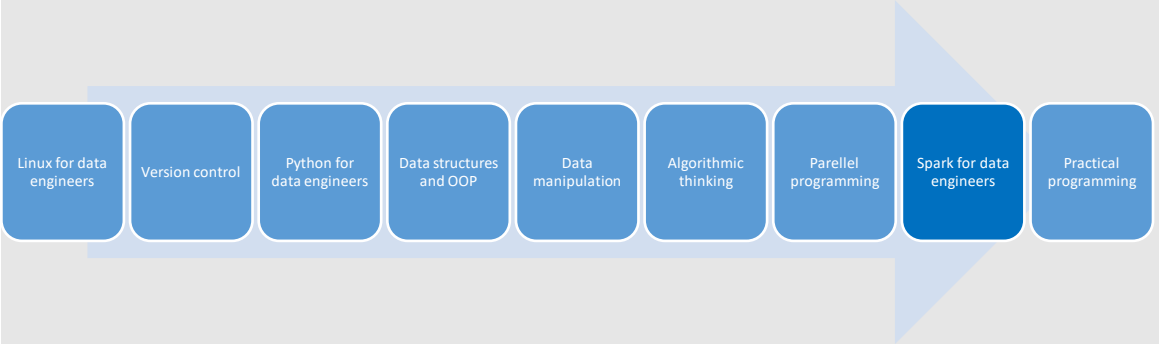
Cell 3 - GroupBy Operator: Group the records by the name of the person, show the first group, count the groups, and collect the job counts. You’re asked to explain the output in plain English.

Cell 4 - Map vs FlatMap: Compute a relation of words by line using map and flatMap functions, and explain the difference between these two functions with examples and a plain English explanation.

Throughout the task, you’re expected to fill in the blanks in the provided code snippets and provide explanations for the outputs and operations.

Link

This topic is for one of 9 topics for this Module.



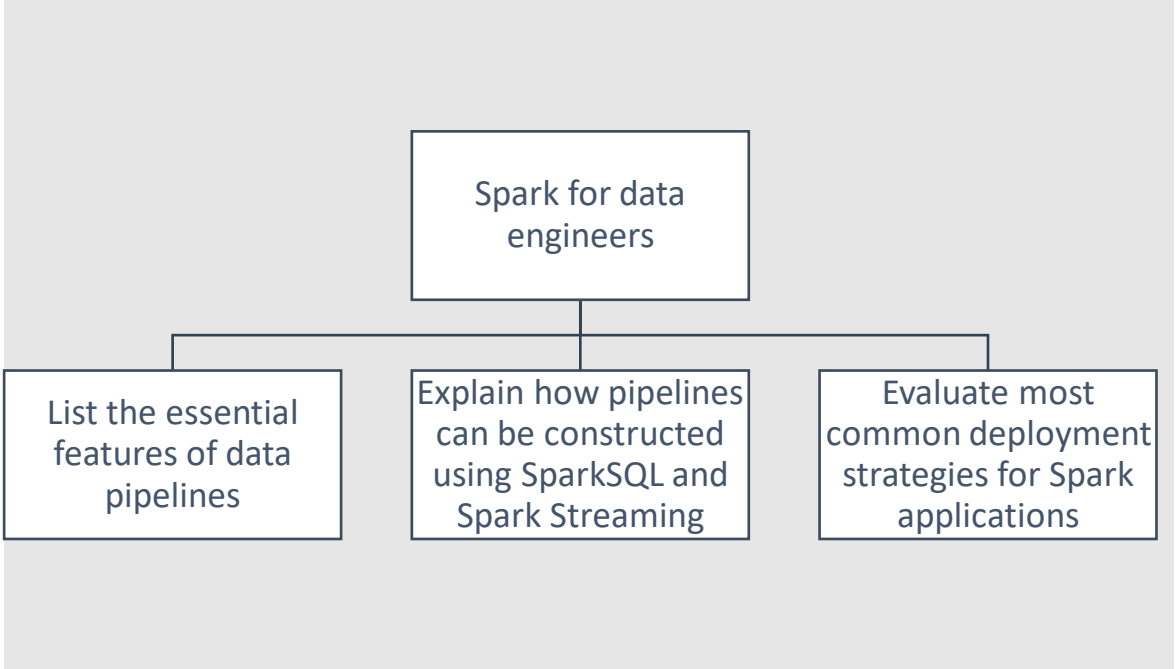
The sequence of topics in this module is carefully designed so that your knowledge and skills will develop as you progress.

The next topic is **Spark for data engineers**.

Topic 8 – Spark for data engineers

Topic Learning Outcomes

As a step towards build your skills towards the final module assessment, the learning objectives for this topic are:



Introduction

In the era of big data, the ability to process and analyse large datasets quickly and efficiently is crucial. Apache Spark, with its ability to handle large-scale data processing tasks, has emerged as a leading tool in this space. This topic is designed to equip you with the knowledge and skills to harness the power of Spark for data processing, and to construct efficient data pipelines using SparkSQL and Spark Streaming.

Data pipelines are essential for the efficient and reliable processing of data. They allow for the automation of data flow between sources and destinations, and they ensure that data is cleaned, transformed, and stored in a way that is ready for analysis. With SparkSQL and Spark Streaming, data engineers can construct robust data pipelines that can handle both batch and real-time data processing.

SparkSQL allows for the processing of structured and semi-structured data using SQL-like syntax, making it accessible for those familiar with SQL. It also integrates seamlessly with the Spark ecosystem, allowing for the use of other Spark libraries in conjunction with SparkSQL. On the other hand, Spark Streaming enables the processing of live data streams in real-time, making it ideal for applications that require immediate insights from data.

When it comes to deploying Spark applications, there are several strategies that data engineers can evaluate. These include standalone deployment, where Spark runs on its own cluster; on-premise deployment, where Spark runs on a company's internal hardware; and cloud-based deployment, where Spark runs on a cloud platform like AWS or Azure. Each of these deployment strategies has its own advantages and considerations, and the choice of strategy would depend on the specific requirements of the application.

By the end of this topic, you will be able to evaluate the use of Spark clusters for data processing, understand the essential features of data pipelines, and construct your own using SparkSQL and Spark Streaming. You will also have a deeper understanding of how data engineers with Spark skills can deliver real-world value.

Structure

Topics for this programme follow a Prepare-Collaborate-Apply structure:

Prepare

This is the stage where you build the knowledge to underpin your learning. This might involve completing interactive e-learning packages, watching videos, or working through reading materials.

It is essential that you make the most of the learning materials provided before attending webinars, as this will allow you to test your knowledge and stretch you understanding further.

Collaborate

This is where you will receive guidance from our expert tutors and coaches to shape and refine your understanding through in-depth explanation, discussion, testing and carrying out more advanced practical and realistic tasks. This also helps to develop valuable team-working skills.

Apply

You now apply the knowledge you have developed to real-world tasks.

Off-the-job learning tasks

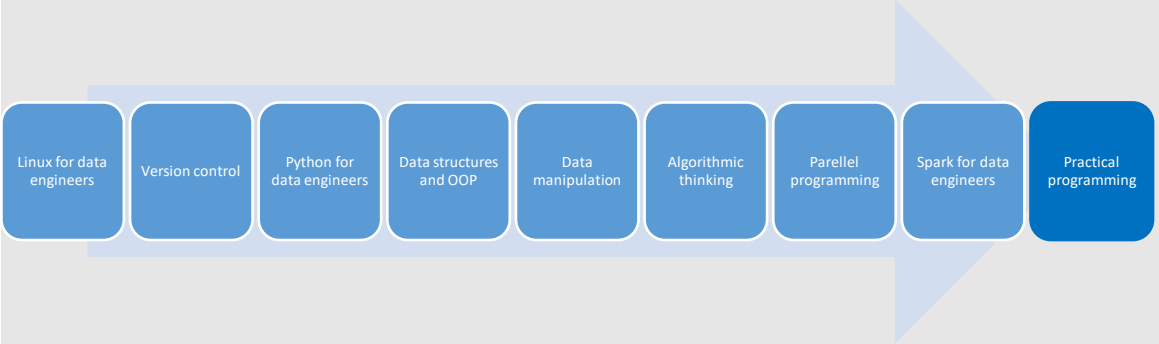
This stage is all about ensuring you truly grasp and retain what you've learned. Through completion of off-the-job (OTJ) revision tasks and tests, you'll get plenty of practice applying your knowledge. Plan to dedicate 6-8 hours each week to guided study and portfolio work, with sessions typically on the same day each week.

Task 1 brief:

Your tutor will provide you with a brief on how to complete the Python notebook for this topic.

Link

This topic is for one of 9 topics for this Module.



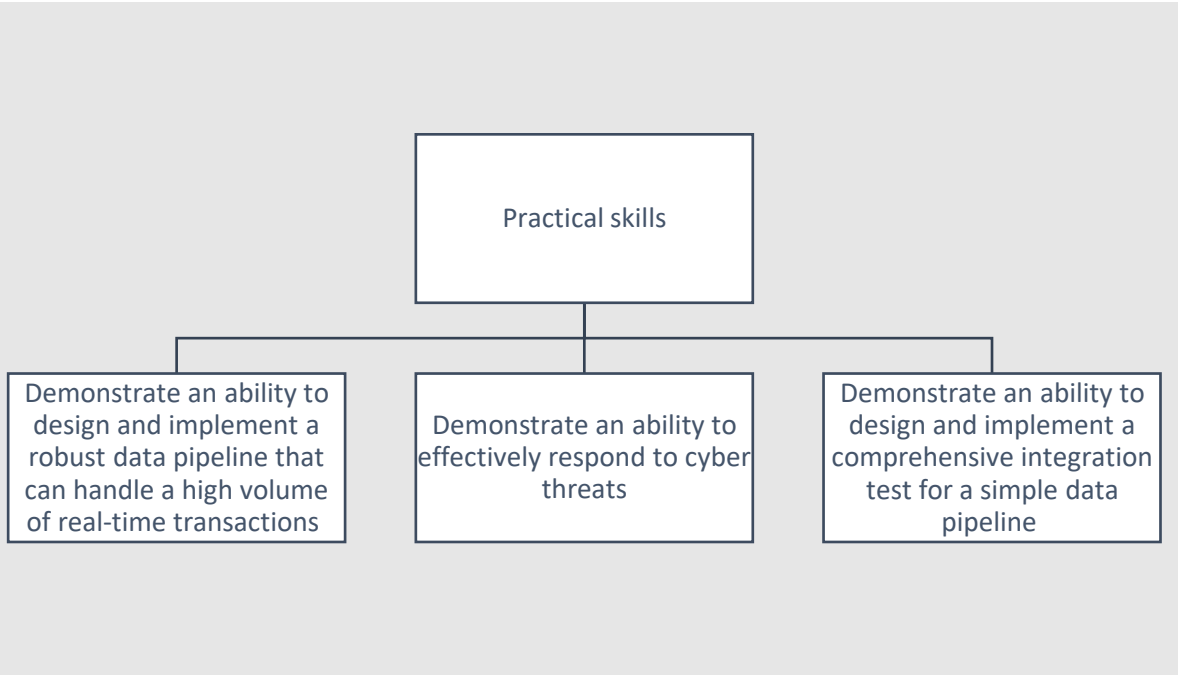
The sequence of topics in this module is carefully designed so that your knowledge and skills will develop as you progress.

The next topic is **Practical programming**.

Topic 9 – Practical programming

Topic Learning Outcomes

As a step towards build your skills towards the final module assessment, the learning objectives for this topic are:



Introduction

This topic is designed with a focus on real-world applications and problem-solving, diving deep into the construction of robust data pipelines capable of handling a high volume of real-time transactions. The ability to design and implement such a pipeline is not just a theoretical skill, but a crucial one for tackling the challenges you will face in the field of data engineering.

In addition to building the pipeline, we will also explore how to ensure its security, demonstrating an ability to effectively respond to cyber threats. This involves understanding potential vulnerabilities and implementing measures to safeguard against them, thereby ensuring the integrity and confidentiality of the data being processed.

Furthermore, we will delve into the importance of comprehensive integration testing for data pipelines. You will learn how to design and implement effective test suites that can validate the functionality and performance of a simple data pipeline, ensuring that it operates as expected under various conditions.

Structure

Topics for this programme follow a Prepare-Collaborate-Apply structure:

Prepare

This is the stage where you build the knowledge to underpin your learning. This might involve completing interactive e-learning packages, watching videos, or working through reading materials.

It is essential that you make the most of the learning materials provided before attending webinars, as this will allow you to test your knowledge and stretch you understanding further.

Collaborate

This is where you will receive guidance from our expert tutors and coaches to shape and refine your understanding through in-depth explanation, discussion, testing and carrying out more advanced practical and realistic tasks. This also helps to develop valuable team-working skills.

Apply

You now apply the knowledge you have developed to real-world tasks.

Off-the-job learning tasks

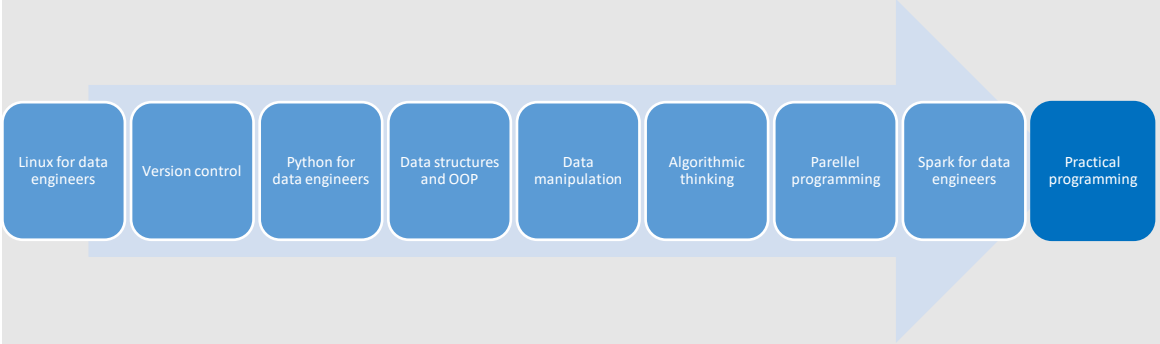
This stage is all about ensuring you truly grasp and retain what you've learned. Through completion of off-the-job (OTJ) revision tasks and tests, you'll get plenty of practice applying your knowledge. Plan to dedicate 6-8 hours each week to guided study and portfolio work, with sessions typically on the same day each week.

Task 1 brief:

Your tutor will provide you with a brief on how to complete the Python notebook for this topic.

Link

This handbook is for one of 9 topics for this Module.



The sequence of topics in this module is carefully designed so that your knowledge and skills will develop as you progress.

This is the last topic in the module.