Course Code	Course Title	L	T	P	C
MACSE502	Programming for Data Science	3	0	2	4
Pre-requisite	NIL	Syllabus Version			
		1.0			

Course Objectives

1. Master Python data structures and object-oriented programming for data analysis and web development.
2. Dive deep into the Pandas libraryfor data processing, modelling and visualization.
3. Understand Scala functional programming concepts, data structures, object-oriented programming and exception handling.
4. Explore Scala libraries, Apache Spark architecture and its functions for big data processing and analytics.
5. Learn criteria for selecting the appropriate programming language.

Course Outcomes

1. Demonstrate proficiency in implementing complex data structures and user-defined data structures in Python.
2. Analyse and manipulate data using advanced features of Pandas
3. Evaluate the use of Scala's case classes, companion objects, and traits in building robust applications.
4. Design and deploy scalable data processing pipelines using Scala libraries and Apache Spark.
5. Select the most suitable programming language based on the project requirements.
6. Apply Python and Scala programming skills to design, develop, and deploy real-world applications.

Module:1	Python Data Structures	8 hours

Condition and Branching Statements, Built- in data structures: List, Tuple, Dictionary, Set, User defined data structures: Stack, Queue, Priority Queue, String handling methods, Exception Handling, Object-Oriented Concepts, APIs and Data Collection, Simple API and REST APIs- HTTP Requests, File Handling- Read/Write Frameworks and Libraries, NLTK, ChatterBot

Module:2	Python Libraries	9 hours

Pandas-Series, DataFrame, Handling Missing Values, Built-in functions, Data Operations, Filtering Data in DataFrame, Data Extraction, Working with Text Data, Merging DataFrames - Data Mining – Scrapy - Beautiful Soup - Data Processing and Modelling: NumPy - SciPy - Pandas - Keras - Scikit - Learn - PyTorch -TensorFlow, XGBoost, Data Visualization: Matplotlib – Seaborn – Bokeh – Plotly - Folium.

Module:3	Scala Data Structures and Object-Oriented Programming	12 hours

Expanded Function Format, Variables and Strings, Getting user input, Numbers, Variable types, Operators, Booleans Data Structures-Arrays, Lists, Tuple, Sets, Hash set, Maps - Functional Combinators Map, Scala Object and Class, Anonymous object-Singleton, Companion Object-Case Classes, Objects-Constructors-Method Overloading - This Keyword – Inheritance - Field Overriding - Final, Abstract Class-Trait, Trait Mixins, Access Modifiers, Scala Array-REPL

Module:4	Scala Libraries and Spark Basics	8 hours

Scala Libraries- Breeze, saddle, Exception Handling, Apache Spark Architecture - Spark Big Data - Apache Spark Components.		
Module:5	**Programming language selection criteria**	**6 hours**
Size of the Deployment: Data, Resource and Load - Security - Skill Set - The targeted platform - The elasticity of a language - The time to production - The performance - The support and community – Purpose – Programmer experience - Ease of Development and Maintenance - Efficiency - Availability of an IDE -Error Checking and Diagnosis		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**45 hours**

Textbooks	
1	Alvaro Fuentes, Become a Python Data Analyst – By Packt Publishing (2018)
2	Bharti Motwani, Data Analytics using Python – By Wiley (2020)
3	Jules S. Damji, Learning Spark: Lightning-Fast Data Analytics, Second Edition – By Shroff/O'Reilly (2020)
4	Data Science and Machine Learning using Python – 10 August 2022, MGH, 2022
Reference Books	
1.	Tome, E., Bhattacharjee, R. and Radford, D. Data Engineering with Scala and Spark: Build streaming and batch pipelines that process massive amounts of data using Scala. Packt Publishing Ltd. (2024)
2.	Perrin, J.-G. Spark in Action, Second Edition: Covers Apache Spark 3 with Examples in Java, Python, and Scala. Manning. (2020)
3.	Wes McKinney, Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter, OReilly, (2022)
4.	https://www.coursera.org/learn/python-for-applied-data-science-ai
5.	https://www.datacamp.com/blog/top-python-libraries-for-data-science
6.	https://www.udemy.com/course/completescala3/?couponCode=IND21PM
7..	https://www.aalpha.net/blog/factors-to-consider-when-choosing-a-programming-language/
Mode of Evaluation: Quiz, Assignment, Design Project, Case Study, Seminar, CAT and FAT	
List of Experiments (Indicative)	
1.	Basic Data Manipulation with Pandas: Create a DataFrame with columns Name, Age, and City containing data for five individuals. Perform the following operations: select only the Name and Age columns, filter rows where Age is greater than 25, add a new column Country with a default value, and sort the DataFrame by Age in descending order. Finally, calculate the average age of the individuals.
2.	Data Cleaning and Preprocessing with Pandas: Create a DataFrame with some missing values in columns Name, Age, and City. Perform the following operations: fill missing values in the Age column with the mean age, drop rows where Name or City is missing, and convert the Age column to integer

	type. Finally, normalize the Age column using min-max scaling.
3.	Creating and Manipulating Arrays: Using NumPy, create a 2D array of shape (3, 4) with random integers between 0 and 10. Perform the following operations: calculate the mean and standard deviation of the entire array, slice the array to get the first two rows and last two columns, reshape the array to shape (4, 3), and perform element-wise multiplication with another array of the same shape.
4.	Feature Engineering, Exploratory Data Analysis: Create a DataFrame with columns Feature1, Feature2, and Target containing random data. Perform the following operations: create a new feature that is the logarithm of Feature1, bin Feature2 into three categories (low, medium, high), and calculate the correlation matrix of the DataFrame. Finally, create a scatter plot of Feature1 vs. Feature2 colored by Target, and interpret any visible patterns.
5.	Scrape data from a webpage and store it in a structured format like CSV or JSON.
6.	Interactive Bar Chart with Plotly. Create an interactive bar chart showing the population of different cities.
7.	Interactive Scatter Plot with Plotly. Create an interactive scatter plot showing the relationship between house size and price.
8.	Interactive Line Plot with Bokeh. Create an interactive line plot showing the daily temperatures over a week using Bokeh.
9.	Interactive Bar Chart with Bokeh: Problem Statement: Create an interactive bar chart showing the sales figures of different products using Bokeh.
10.	Interactive Scatter Plot with Bokeh: Problem Statement: Create an interactive scatter plot showing the relationship between petal length and petal width from the Iris dataset using Bokeh.
11.	Visualizing Clusters with scikit-learn and Plotly Problem Statement: Perform K-means clustering on the Iris dataset and visualize the clusters using an interactive 3D scatter plot in Plotly.
12.	Visualizing PCA with scikit-learn and Plotly: Problem Statement: Perform Principal Component Analysis (PCA) on the Iris dataset and visualize the first two principal components using an interactive 2D scatter plot in Plotly.
13.	Introduction to arrays in Scala: Create an array of integers with the values 2, 5, 9, 14, 20. Write a function that takes this array and returns the sum of its elements. Next, create an array of strings with the names of five different fruits. Write a function that concatenates all elements of this array, separated by commas. Finally, iterate over the array of integers and print each element to the consol.
14.	Understanding lists, sets, and tuples in Scala: Create a list of the first five prime numbers. Write a function that takes this list and returns a new list with each element squared. Then, create a set of unique characters from the string "hello world" and write a function that takes two sets and returns their intersection. Create a tuple with three elements: an integer, a string, and a boolean, then access and print each element.

15.	Handling collisions and resolving conflicts in HashMap's: Create a HashMap with keys representing student names and values representing their grades, then insert multiple entries including a duplicate key with a different grade. Write a function to merge two HashMap's, resolving conflicts by taking the higher grade. Finally, write a function to handle collisions by chaining, using lists to store multiple values for a single key.
16.	Exploring advanced functional combinators beyond the basic set: Create a list of integers from 1 to 10, then use filter to create a new list with only even numbers. Use map to create a new list where each element is multiplied by 3. Use flatMap to create a list of tuples where each integer is paired with its square. Finally, use foldLeft to calculate the product of all elements in the list.
17.	Case Classes: Create a case class Person with fields name, age, and city. Create a list of Person objects representing five different individuals. Write a function to filter out people older than 30. Next, write a function that groups people by their city. Then, write a function that transforms each Person object into a string in the format "Name is Age years old and lives in City". Finally, write a function to sort the list of people by age in ascending order.
18.	Use Saddle to load and manipulate a dataset of fruit prices and quantities, filtering for apples and calculating their average price. You will then visualize the price trend over time using Vegas. This exercise covers basic data manipulation with Saddle and interactive visualization with Vegas.
19.	Use Breeze library to calculate total revenue for each fruit type by multiplying price and quantity from a dataset. You will then create a bar chart to visualize these revenues using Plotly. Scala. This exercise introduces numerical computations with Breeze and dynamic visualizations with Plotly. Scala.
20.	Analyse fruit prices using Spire for precise statistics and Saddle for data manipulation. You will compute the mean and variance of prices for each fruit and visualize these trends over time using Vegas. This exercise demonstrates the use of Spire for numerical precision, advanced data manipulation with Saddle, and effective visualization with Vegas.

Total hours:	**30 hours**

Mode of Evaluation: Continuous Assessments and FAT			
Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	