B.Sc. DATA SCIENCE
SYLLABUS
(Common core syllabus under CBCS)

With effect from the academic year: 2025-2026 and onwards

Accredited with ‘A^{+}’ by NAAC

DEPARTMENT OF COMPUTER SCIENCE
KAKATIYA UNIVERSITY
WARANGAL-506009,
TELANGANA

KAKATIYA UNIVERSITY, WARANGAL

DEPARTMENT OF COMPUTE SCIENCE

B.Sc. (DATA SCIENCE)

with effect from Academic Year: 2025 – 2026

YEAR/ Semester	**Type of Course**	**Paper Title**	**Hrs Per Week T: Theory P: Practical**	**No. of Credits**	**Max. Marks**		
					Intr. Marks	**External marks**	**Total marks**
I/I	DSC-1	Programming with C	4 (T)	4	20	80	125
	DSC-1	Programming with C (Lab)	2 (P)	1	--	25	
I/II	DSC-2	Problem Solving and Python Programming	4 (T)	4	20	80	125
	DSC-2	Problem Solving and Python Programming (Lab)	2 (P)	1	--	25	
II/I	DSC-3	Data Engineering with Python	4 (T)	4	20	80	125
	DSC-3	Data Engineering with Python (Lab)	2 (P)	1	--	25	
II/II	DSC-4	Machine Learning	4 (T)	4	20	80	125
	DSC-4	Machine Learning (Lab)	2 (P)	1	--	25	

B.Sc. DATA SCIENCE

I Year: Semester-I

Course type	Paper Title	Hours per week	Marks	
DSC-1	Programming with C	**Theory: 04**	**Internal**	**external**
		Credit: 04	**20**	**80**

Unit - I

Computing Concepts: Types of Software, Programming Languages, Translator Programs, Problem Solving Techniques Using Computer. Overview of C: History of C, Importance of C, Sample rograms, Basic Structure of C Programs, Programming Style, Executing a ' C ' Program. **Constants, Variables, and Data Types**: Introduction, Character set, C tokens, Keywords and Identifiers, Constants, Variables, Data Types, Declaration of Variables, Declaration of Storage Class, Assigning Values to Variables, Defining Symbolic Constant, Declaring Variable as Constant and Volatile, Overflow and Underflow of Data. **Managing Input and Output Operations**: Introduction, reading a Character, Writing a Character, Formatted Input and Output. Operators and Expressions: Introduction, Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Increment and Decrement Operator, Conditional Operators, Bitwise Operators, Special Operators, Arithmetic Expressions, Evaluation of expressions, Precedence of Arithmetic Operators, Some Computational Problems, Type Conversions in Expressions, Operator Precedence and Associativity, Mathematical Functions.

Unit – II

Decision Making and Branching: Introduction, Decision Making with IF Statement, Simple If Statement, The If....Else Statement, Nested if, The else if Ladder, The Switch Statement, The ?: Operator, The Goto Statement. Decision Making and Looping: Introduction, The While statement, the do Statement, the For Statement, Jumps in Loops. Arrays: Introduction, One-dimensional Arrays, Declaration and Initialization of Onedimensional Arrays, Two-dimensional Arrays, Initializing two-dimensional Arrays, Multi-dimensional Arrays, Dynamic Arrays.

Unit III

Character Arrays and Strings: Introduction, Declaring and Initializing String Variables, Reading Strings from Terminal, Writing Strings to Screen, Arithmetic Operations on Characters, Putting Strings together, Comparison of Two Strings, String-handling Functions, Table of Strings, Other Features of Strings**. User-Defined Functions** : Introduction, Need for User-Defined Functions, A Multifunction Program, Elements of user-defined Functions, Definition of Functions, Return Values and their Types, Function Calls, Function Declaration, Category of Functions, No Arguments and No Return Values, Arguments but No Return Values, Arguments with Return Values, No Arguments but Returns a Value, Functions that Return Multiple Values, Nesting of Functions, Recursion, Passing Arrays to Functions, Passing Strings to Functions, The Scope, Visibility and Lifetime of Variables.

Unit- IV

Pointers : Introduction, Understanding Pointers, Accessing the Address of a Variable, Declaring Pointer Variables, Initialization of pointer Variables, Accessing a Variable through its pointer, Chain of Pointers, Pointer Expressions, Pointer Increments and Scale Factor, Pointers and Arrays, Pointer and Arrays, Pointers and Character Strings, Arrays of Pointers, Pointers as Function Arguments, Functions Returning Pointers, Pointers to Functions, Pointers and Structures. **Structures and Unions**: Introduction, Defining a Structure, Declaring Structure Variables, Accessing Structure Members, Structure Initialization, copying and Comparing Structure Variables, Operations on Individuals Members, Arrays of Structures, Arrays within Structures, Structures within Structures, Structures and Functions, Unions, Size of Structures, Bit Fields. File Management in C : Introduction, Defining and Opening a File, Closing a File, Input/Output Operations on Files, Error Handling During I/O Operations, Random Access to Files, Command Line Arguments.

Text Book:

1. Computing Fundamentals & C Programming – by E Balagurusamy, 1st Edition McGrawHill Education.

References:

1. Spoken Tutorial on “C”, as E-resource for Learning. http://spoken-tutorial.org

Course type	Paper Title	Hours per week	Marks	
DSC-1	Programming with C Lab	Practical: 02	Internal	external
		Credits: 01	--	25

Exercises:

1. Write a c program for electricity bill tacking different categories of users, different slabs in each category.(using nested if else statement)
2. Write a c program to evaluate the following using loops
   i 1+x^2/2!+x^4/4!+...up to 5 terms
   ii x+x^3/3!+x^5/5!+...up to 5 terms
3. Write a c program to check whether the given number is Prime or not and perfect or not
4. Write a c program to find the mean, mode, median, and variance of list of values by using one dimensional array
5. Write a menu driven program to read a list of numbers and perform the following operations
   1. Print the list 2. Delete duplicates from the list 3. Reverse the list.
6. Write a program to read a list of numbers and search for given number using binary search algorithm and if found display its index otherwise display the message "element not found in the list" using functions
7. Write a menu driven program to read two matrices and compute their sum and product using functions .
8. Write a menu driven program to read list of student names and perform the following operations using functions.
   1. To print list of names 2. To sort them in ascending order 3. To print list after sorting
9. Write a c program that consists of recursive functions to find factorial of a given number and Print the Pascal triangle using binomial theorem .
10. Write a menu driven program to read list of student names and perform the following operations using array of character pointers.
11. a) To insert a student name b) To delete a name c) To print the names 12. Write a program to create an array of structures and display their content.
13. Write a program to demonstrate nested structures.
14. Write a program to create 10 student records and display those using pointers.
15. Write a program to display the contents of a file.
16. Write a program to copy the contents of one file into another file.
17. Write a program to append the contents of one file to another file.
18. Write a program to demonstrate the command line arguments.

B.Sc. DATA SCIENCE

I Year: Semester-II

Course type	Paper Title	Hours per week	Marks	
DSC-2	**Problem Solving and Python Programming**	**Theory: 04**	**Internal**	**external**
		Credit: 04	**20**	**80**

Unit-I

Introduction to Computing and Problem Solving: Fundamentals of Computing – Computing Devices – Identification of Computational Problems – Pseudo Code and Flowcharts – Instructions – Algorithms – Building Blocks of Algorithms. **Introduction to Python Programming**: Python Interpreter and Interactive Mode– Variables and Identifiers – Arithmetic Operators – Values and Types – Statements, Reading Input, Print Output, Type Conversions, The type() Function and Is Operator, Dynamic and Strongly Typed Language. **Control Flow Statements**: The if, The if…else, The if…elif…else Decision Control Statements, Nested if Statement, The while Loop, The for Loop, The continue and break Statements.

Unit-II

Functions: Built-In Functions, Commonly Used Modules, Function Definition and Calling the Function, The return Statement and void Function, Scope and Lifetime of Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command Line Arguments. **Strings**: Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings.

Unit-III

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list Parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension; Illustrative programs: selection sort, insertion sort, mergesort, histogram. **Files and exception**: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file.

Unit-IV

Object-Oriented Programming: Classes and Objects, Creating Classes in Python, Creating Objects in Python, The Constructor Method, Classes with Multiple Objects, Class Attributes versus Data Attributes, Encapsulation, Inheritance The Polymorphism. **Functional Programming**: Lambda. Iterators, Generators, List Comprehensions.

References:

1. Introduction to Python Programming. Gowrishankar S, Veena A. CRC Press, Taylor & Francis Group, 2019
2. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist'', 2nd edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016 (http://greenteapress.com/wp/think- python/)

Course type	Paper Title	Hours per week	Marks	
DSC-2	Problem Solving and Python Programming Lab	Practical: 02	Internal	external
		Credits: 01	--	25

Exercises

I. Programs to demonstrate the usage of operators and conditional statements

1. Write a program that takes two integers as command line arguments and prints the sum of two integers.
2. Program to display the information:

   Your name, Full Address, Mobile Number, College Name, Course Subjects
3. Program to find the largest number among 'n' given numbers.
4. Program that reads the URL of a website as input and displays contents of a webpage.

II. Programs to demonstrate usage of control structures

5. Program to find the sum of all prime numbers between 1 and 1000.
6. Program that reads set of integers and displays first and second largest numbers.
7. Program to print the sum of first 'n' natural numbers.
8. Program to find the product of two matrices.
9. Program to find the roots of a quadratic equation

III. Programs to demonstrate the usage of Functions and Recursion

10. Write both recursive and non-recursive functions for the following:
    a. To find GCD of two integers
    b. To find the factorial of positive integer
    c. To print Fibonacci Sequence up to given number 'n'
    d. To convert decimal number to Binary equivalent
11. Program with a function that accepts two arguments: a list and a number 'n'. It should display all the numbers in the list that are greater than the given number 'n'.
12. Program with a function to find how many numbers are divisible by 2, 3,4,5,6 and 7 between 1 to 1000

IV. Programs to demonstrate the usage of String functions

13. Program that accept a string as an argument and return the number of vowels and consonants the string contains.
14. Program that accepts two strings S1, S2, and finds whether they are equal are not.
15. Program to count the number of occurrences of characters in a given string.
16. Program to find whether a given string is palindrome or not

V. Programs to demonstrate the usage of lists, sets, dictionaries, tuples and files.

17. Program with a function that takes two lists L1 and L2 containing integer numbers as parameters. The return value is a single list containing the pair wise sums of the numbers in L1 and L2.
18. Program to read the lists of numbers as L1, print the lists in reverse order without Using reverse function.

22. Write a program that combines lists L1 and L2 into a dictionary.

19. Program to find mean, median, mode for the given set of numbers in a list.
20. Program to find all duplicates in the list.
21. Program to o find all the unique elements of a list.
22. Program to find max and min of a given tuple of integers.
23. Program to find union, intersection, difference, symmetric difference of given two sets.
24. Program to display a list of all unique words in a text file
25. Program to read the content of a text file and display it on the screen line wise with a line number followed by a colon
26. Program to analyse the two text files using set operations
27. Write a program to print each line of a file in reverse order.

VI. Programs to demonstrate the usage of Object Oriented Programming

28. Program to implement the inheritance
29. Program to implement the polymorphism

VII. Programs to search and sort the numbers

30. Programs to implement Linear search and Binary search
31. Programs to implement Selection sort, Insertion sort

B.Sc. DATA SCIENCE
II Year: Semester-I

Course type	Paper Title	Hours per week	Marks	
DSC-3	**Data Engineering with Python**	**Theory: 04**	**Internal**	**external**
		Credit: 04	**20**	**80**

Unit – I

Data Science: Data Analysis Sequence, Data Acquisition Pipeline, Report Structure [Reference 1(Chapter 1-Unit1 to Unit 3)]]. **Files and Working with Text Data:** Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, The Pickle Module, Reading and Writing CSV Files, Python os and os. Path Modules. [Reference 2, Chapter 9)] **Working with Text Data:** JSON and XML in Python[Reference 2, Section12.2]

Unit – II

Working with Text Data: Processing HTML Files, Processing Texts in Natural Languages [Reference 1(Chapter3 –Unit 13, and Unit16). **Regular Expression Operations:** Using Special Characters, Regular Expression Methods, Named Groups in Python Regular Expressions, Regular Expression with *glob* Module [Reference 2-Chapter 10]

Unit – III

Working with Databases: Setting Up a MySQL Database, Using a MySQL Database: Command Line, Using a MySQL Database, Taming Document Stores: MongoDB [Reference 1(Chapter4-Unit17toUnit20)]. **Working with Tabular Numeric Data(Numpy with Python)**: NumPy Arrays Creation Using *array()* Function, Array Attributes, NumPy Arrays Creation with Initial Placeholder Content, Integer Indexing, Array Indexing, Boolean Array Indexing, Slicing and Iterating in Arrays, Basic Arithmetic Operations on NumPy Arrays, Mathematical Functions in NumPy, Changing the Shape of an Array, Stacking and Splitting of Arrays, Broadcasting in Arrays. [Reference 2: Section 12.3)]

Unit – IV

Working with Data Series and Frames: Pandas Data Structures, Reshaping Data, Handling Missing Data, Combining Data, Ordering and Describing Data, Transforming Data, Taming Pandas File I/O [Reference 1 (Chapter 6-Unit 31 to Unit 37)]. **Plotting**: Basic Plotting with PyPlot, Getting to Know Other Plot Types, Mastering Embellishments, Plotting with Pandas [Reference 1(Chapter8-Unit 41 to Unit 44)]

References:

1. Data Science Essentials in Python: Collect, Organize, Explore, Predict, Value. Dmitry Zinoriev,The Pragmatic Programmers LLC, 2016
2. Introduction to Python Programming**.** Gowrishankar S., Veena A. CRC Press, Taylor & Francis Group, 2019

Suggested Reading

3. Python for Everybody: Exploring Data Using Python 3. Charles R Severance, 2016
4. Python Data Analytics – Data Analysis and Science using Pandas, matplotlib and the Python Programming Language. Fabio Nelli, Apress, 2015
5. Website Scraping with Python. Using BeautifulSoup and Scrapy. GáborLászlóHajba, Apress, 2018

Course type	**Paper Title**	**Hours per week**	**Marks**	
DSC-3	**Data Engineering with Python Lab**	**Practical: 02**	**Internal**	**external**
		Credits: 01	**--**	**25**

Objective:

The main objective of this laboratory is to put into practice the ETL (extract, transform, load) pipeline which will extract raw data, clean the data, perform transformations on data, load data and visualize the data. This requires mentoring by TCS.

Libraries

In this course students are expected to extract, transform and load input data that can be text files, CSV files, XML files, JSON, HTML files, SQL databases, NoSQL databases etc.,. For doing this, they should learn the following Python libraries/modules: pandas, numpy, BeautifulSoup, pymysql, pymongo, nltk, matplotlib

Datasets

For this laboratory, appropriate publicly available datasets, can be studied and used.
Example: MNIST (http://yann.lecun.com/exdb/mnist/), UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets.html), Kaggle (https://www.kaggle.com/datasets)
Twitter Data

Exercises

1. Write programs to parse text files, CSV, HTML, XML and JSON documents and extract relevant data. After retrieving data check any anomalies in the data, missing values etc.
2. Write programs for reading and writing binary files
3. Write programs for searching, splitting, and replacing strings based on pattern matching using regular expressions
4. Design a relational database for a small application and populate the database. Using SQL do the CRUD (create, read, update and delete) operations.
5. Create a Python MongoDB client using the Python module pymongo. Using a collection object practice functions for inserting, searching, removing, updating, replacing, and aggregating documents, as well as for creating indexes
6. Write programs to create numpy arrays of different shapes and from different sources, reshape and slice arrays, add array indexes, and apply arithmetic, logic, and aggregation functions to some or all array elements
7. Write programs to use the pandas data structures: Frames and series as storage containers and for a variety of data-wrangling operations, such as:
    - Single-level and hierarchical indexing
    - Handling missing data
    - Arithmetic and Boolean operations on entire columns and tables
    - Database-type operations (such as merging and aggregation)
    - Plotting individual columns and whole tables
    - Reading data from files and writing data to

B.Sc. DATA SCIENCE
II Year: Semester-II

Course type	Paper Title	Hours per week	Marks	
DSC-3	**Machine Learning**	**Theory: 04**	**Internal**	**external**
		Credit: 04	**20**	**80**

Unit-I

Introduction: What does it mean to learn, Some canonical Learning Problems, The Decision Tree Model of Learning, Formalizing the Learning Problem ID3 Algorithm [Reference1, 2] **Limits of Learning:** Data Generating Distributions, Inductive Bias, Not Everything islearnable, Under fitting and Overfitting, Separation of training and test Data, Models, parameters and Hyperparameters, Real World Applications of Machine Learning **Geometry and Nearest Neighbours:** From Data to Feature Vectors, k-Nearest Neighbours, Decision Boundaries, k-means Clustering, High Dimensions [Reference 1]

Unit-II

The Perceptron: Bio-inspired Learning, The Perceptron Algorithm, Geometric Interpretation, Interpreting Perceptron Weights, Perceptron Convergence and Linear Separability, Improved Generalization, Limitations of the Perceptron Practical **Issues:** Importance of Good Features, Irrelevant and Redundant Features, Feature Pruning and Normalization, Combinatorial Feature Explosion, Evaluating Model Performance, Cross Validation, Hypothesis Testing and Statistical Significance, Debugging Learning Algorithms, Bias Variance tradeoff. **Linear Models:** The Optimization Framework for Linear Models, Convex Surrogate Loss Functions, Weight Regularization, Optimization and Gradient Descent, Support Vector Machines [Reference 1]

Unit-III

Probabilistic Modelling: Classification by Density Estimation, Statistical Estimation, Naïve Bayes Models, Prediction [Reference 1]. **Neural Networks:** Bio-inspired Multi-Layer Networks, The Back-propagation Algorithm, Initialization and Convergence of Neural Networks, Beyond two layers, Breadth vs Depth, Basis Functions [Reference 1]

Unit IV

Unsupervised Learning: Clustering Introduction, Similarity and Distance Measures,Agglomerative Algorithms, Divisive Clustering, Minimum Spanning Tree [Reference 2]. **Association Rules:** Introduction, large Itemsets, Apriori Algorithm [Reference 2]

References:

1. A Course in Machine Learning (CIML). Hal Daume III, 2017 (freely available online) http://ciml.info/
2. Data Mining: Introductory and Advanced Topics. Margaret H Dunham, Pearson Education, 2003

Suggested Reading:

3. Hands on Machine Learning with SciKit-Learn, Keras and Tensor Flow. AurélienGéron. O'Reily, 2019
4. Machine Learning with Python Cookbook. Chris Albo, O'Reily, 2018
5. Introduction to Machine Learning with Python: A guide. Andreas C Miller, Sarah Guido. O'Reily, 2017

Course type	Paper Title	Hours per week	Marks	
DSC-4	Machine Learning Lab	Practical: 02	Internal	external
		Credits: 01	--	25

Objective:

The main objective of this laboratory is to put into practice the various machine learning algorithms for data analysis using Python and Weka.

ML Toolkits

Students are expected to learn

1. Scikit-learn(https://scikit-learn.org/) an open source machine learning Python library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities.
2. Weka (http://www.cs.waikato.ac.nz/ml/weka/)is another widely used ML toolkit.

Datasets

1. The sklearn datasets package embeds small toy datasets. It includes utilities to load these datasets. It also includes methods to load and fetch popular reference datasets and features some artificial data generators. Students are expected to study and make use of these datasets
2. Weka also has provides various data sets.

References:

1. Scikit-learn user guide.https://scikit-learn.org/stable//_downloads/scikit-learn-docs.pdf
2. Ian Witten, Eibe Frank, and Mark Hall, Chris Pal. DATA MINING: Practical Machine Learning Tools and Techniques, 4th Edition. Morgan Kaufmann.

Exercises

1. Write a Python program using Scikit-learn to split the iris dataset into 70% train data and 30% test data. Out of total 150 records, the training set will contain 120 records and the test set contains 30 of those records. Print both datasets

2. Write Python program to use sklearn's Decision Tree Classifier to build a decision tree for the sklearn's datasets. Implement functions to find the importance of a split (entropy, information gain, gini measure)

3. Write a Python program to implement your own version of the K-means algorithm. Then apply it to different datasets and evaluate the performance.

4. Design a perceptron classifier to classify handwritten numerical digits (0-9). Implement using scikit or Weka.

5. Write a Python program to classify text as spam or not spam using the Naïve Bayes Classifier

6. Use WEKA and experiment with the following classifiers: Association Rule Mining (Apriori), Agglomerative and Divisive Clustering.

Annexure – I (Credits)
Proposed CBCS Structure from 2025-26 for Under Graduate Courses

Courses		Papers	Total Credits	Credits for each paper / Semester						Credits for each paper / Semester						Credits for each paper / Semester					
				BA						B.Com.						B.Sc.					
				I	II	III	IV	V	VI	I	II	III	IV	V	VI	I	II	III	IV	V	VI
Core Courses DSC	Major-1	6	30	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	Major -2	6	30	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	Minor-1	4	20	5	5	5	5	-	-	5	5	5	5	-	-	5	5	5	5	-	-
MIL/AEC (First Language)	English	4	20	5	5	5	5	-	-	5	5	5	5	-	-	5	5	5	5	-	-
Second Language (Telugu, Hindi, Urdu, etc.)		4	20	5	5	5	5	-	-	5	5	5	5	-	-	5	5	5	5	-	-
Multi-Disciplinary Course	MDC 1	1	4	-	-	-	-	4	-	-	-	-	-	4	-	-	-	-	-	4	-
Sec 1, 2		2	4					2	2					2	2					2	2
Sec 3, 4		2	4					2	2					2	2					2	2
Value added course (VAC)	VAC 1, 2	2	6	-	-	-	-	3	3	-	-	-	-	3	3	-	-	-	-	3	3
Internships	Internship / Project	1	4	-	-	-	-	-	4	-	-	-	-	-	4	-	-	-	-	-	4
Total Credits in each semester			142	25	25	25	25	21	21	25	25	25	25	21	21	25	25	25	25	21	21
Total Credits in UG				142						142						142					
Credits under Non-CGPA (Community engagement and service)		NSS /NCC /sports / Extra curricular	6	Upto 6 (2 in each year)						Upto 6 (2 in each year)						Upto 6 (2 in each year)					
		IKS	4	Upto 4 (2 in each, after I & II years)						Upto 4 (2 in each, after I & II years)						Upto 4 (2 in each, after I & II years)					