गोंय विद्यापीठ
ताळगांव पठार,
गोंय - ४०३ २०६
फोन : + ९१ - ८६६९६०९०४८

Goa University
Taleigao Plateau,Goa - 403 206
Tel : +91-8669609048
Email : registrar@unigoa.ac.in
Website : www.unigoa.ac.in

(Accredited by NAAC with Grade A+)

GU/Acad –PG/BoS -NEP/2024/844 Date: 27.03.2026

CIRCULAR

In supersession to the Circular No. GU/Acad –PG/BoS -NEP/2024/523 dated 28.10.2025, the syllabus of the **Master of Science (Integrated) in Data Science** Programme is enclosed.

The Dean/ Vice-Deans of the Goa Business School are requested to take note of the above and bring the contents of the Circular to the notice of all concerned**.**

ASHWIN V LAWANDE
Digitally signed by ASHWIN V LAWANDE
Date: 2026.03.27 11:04:32 +05'30'

(Ashwin V. Lawande)
Deputy Registrar – Academic

To,
1. The Dean, Goa Business School, Goa University.
2. The Vice-Deans, Goa Business School, Goa University.

Copy to:
1. The Chairperson, BOS in Data Science and Artificial Intelligence.
2. The Controller of Examinations, Goa University.
3. The Assistant Registrar, UG Examinations, Goa University.
4. The Assistant Registrar, PG Examinations, Goa University
5. Directorate of Internal Quality Assurance, Goa University for uploading the Syllabus on the University website.

Programme Structure for Semester I to X – Master of Science (Integrated) in Data Science										
Semester	**Major -Core**	**Minor/VET**	**MC**	**AEC**	**SEC**	**I**	**D**	**VAC**	**Total Credits**	**Exit**
I	**Major-1 IDS-100 Mathematical Foundations (4T)**	**Minor-1 IDS-111 Problem Solving and Program Design (3T+1P)**	**MC-1 IDS-131 E-Commerce (3T)**	**AEC-1 (English-I) (2)**	**IDS-141 Programming in R (1T+2P)**			**VAC-1 (2) VAC-2 (2)**	**20**	
II	**Major-2 IDS-101 Relational Database and SQL (3T+1P) (DELETED) Major-2 IDS-102 Data Management (2T+2P)**	**Minor-2 IDS-112 Fundamentals of Statistics (4T)**	**MC-2 IDS-132 Office Software Automation (3T)**	**AEC-2 (MIL-I) (2)**	**IDS-142 Programming in Python (1T+2P)**			**VAC-3 (2) VAC-4 (2)**	**20**	

III	Major-3 IDS-200 Introduction to Data Science (4T) Major-4 IDS-201 Linear Algebra (4T)	Minor-3 IDS-211 Deductive and Inferential Mathematics (4T) OR IDS-212 Discrete Mathematics (4T)	MC-3 IDS-231 Website Design (3T)	AEC-3 (English-II) (2)	IDS-241 Basic Toolkit for Research (1T+2P)				20	
IV	Major-5 IDS-202 Data Modeling and Visualization (3T+1P) Major-6 IDS-203 Linear Programming and Optimization (4T)	Minor-4 VET IDS-221 Rapid App Development (2T+2P)		AEC-4 (MIL-II) (2)					20	

	Major-7 **IDS-204** **Fundamentals of Computing Systems Design (4T)** **Major-8** **IDS-205** **LINUX Fundamentals (2P)**									
V	**Major-9** **IDS-300** **Introduction to Machine Learning (2T+2P)** **Major-10** **IDS-301** **Data Mining and Warehousing (2T+2P)**	**Minor-5 VET** **IDS-321** **Software Engineering (2T+2P)**				**IDS-361** **Internship (2)**			**20**	

	Major-11 **IDS-302** **Business Data Management** **(4T)** **Major-12** **IDS-303** **Data Structures** **(2T)**									
VI	**Major-13** **IDS-304** **Cloud Computing** **(2T+2P)** **Major-14** **IDS-305** **Business Analytics and Intelligence** **(2T+2P)** **Major-15** **IDS-306** **Thematic Ideas in Data Science** **(4T)**	**Minor-6 VET** **IDS-322** **Big Data** **(2T+2P)** **OR** **IDS-323** **Usability and Quality Assurance** **(2T+2P)**							**20**	

	Major-16 IDS-307 Project (4)									
VII	**IDS-400 Design and Analysis of Algorithms (4T) IDS-401 Advanced Data Management (3T+1P) IDS-402 Design Thinking for Data Science Applications (4T) IDS-403 Artificial Intelligence (3T+1P)**	**IDS-411 Programming Paradigms (3T+1P) OR IDS-412 High Performance Computing (3T+1P)**							**20**	

VIII	**IDS-404 Numerical Methods (4T)** **IDS-405 Building Data Science Applications (3T+1P)** **IDS-406 Predictive Analytics (3T+1P)** **IDS-407 Theory of Sampling and Design of Experiments for Data Analysis (4T)**	**IDS-413 Natural Language Processing (4T)** **OR** **IDS-414 Visual Computing (4T)**							**20**	
IX	**IDS-500 Research Methodology (4)**	**IDS-511 Text Mining and Sentiment Analysis (4T)**							**20**	

	IDS-501 Deep Learning (4T) **RSEC Course (4)** **RSEC Course (4)**	**OR** **IDS-512 Image Processing (4T)**								
X	**RSEC Course (4)**					**IDS-561 Dissertation (16)**			**20**	

List of Research Specific Elective Courses RSEC (students choose any three from a minimum of five)
IDS-502 (Speech Processing) (4)
IDS-503 (Recommender Systems) (4)
IDS-504 (Financial Machine Learning) (4)
IDS-505 (Bayesian Data Analysis) (4)
IDS-506 (Reinforcement Learning) (4)
IDS-507 (Time Series Analysis) (4)
IDS-508 (Topological Data Analysis) (4)

SEMESTER I

Major Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-100
Title of the Course : Mathematics Foundations
Number of Credits : 4
Effective from AY : 2024-25

Pre-requisites for the Course:	NIL	
Course Objectives:	To introduce the students to 1. develop logical reasoning among students 2. organize all aspects of mathematics 3. understand on the fundamental concepts, assumptions and principles of mathematics	
Units	**Content**	**No. of hours**
I	**Logic and Propositional Calculus**: Propositions and Compound Statements; Basic Logical Operations; Propositions and Truth Tables; Tautologies and Contradictions; Logical Equivalence; Algebra of Propositions; Conditional and Biconditional Statements. **Sets**: Sets and their representation; The empty set; Finite and Infinite Sets; Equal Sets; Subsets; Power Set; Universal Set; Union and Intersection of sets; Venn Diagrams; Operations on Sets; Complement of a set. **Relations and Functions**: Cartesian product of sets; Relation and their types; Functions and their types; Algebra of functions; Composition of functions; Invertible functions; Binary operations.	**15**

II	**Limits**: Geometric meaning of limits; Standard limits. **Continuity**: Geometric meaning of continuity; Continuous functions; Algebra of continuous functions; Examples of continuous functions; Discontinuities; Types of Discontinuities. **Differentiability**: First principle of differentiation; Algebra of differentiability namely sum/product/quotient rule; Examples; Result that every differentiable function is continuous; Derivative of the composition; Chain rule; (Statements of these results only. However, the idea of the proof, though not a part of the syllabus, is encouraged) Examples; Optimization problems.	**15**
III	**Understanding Integrals** integration and its relation to differentiation; antiderivatives; indefinite integrals and methods; definite integrals and interpretation as area under curve; linearity of integration; fundamental theorem of calculus **Application of Integrals** Geometric: area under curve and volumes; Physics: displacement from velocity and velocity from acceleration; Economics: consumer and producer surplus; Probability: probabilities using integrals	**15**
IV	**Complex Numbers**: Algebra of complex numbers; Modulus and Complex conjugate; Argand plane and polar representation. **Vector Algebra**: Types of vectors; Addition of vectors; Multiplication of a vector by a scalar; Dot product and cross product of vectors, and their geometrical interpretation; Concept and computation of gradient, divergence, and curl of a vector field.	**15**
Pedagogy:	Lectures/ Case Analysis/ Assignments/ Classroom Interaction/Quiz	
References/ Reading:	1. Kumar, A., Kumaresan, S., & Sarma, B. K. (2018). A Foundation Course in Mathematics. Alpha Science International. 2. Lipschutz, S., & Lipson, M. (2022). Schaum's Outline of Discrete Mathematics. 3. Mendelson, E. (2008). BEGINNING CALCULUS 3ed. McGraw-Hill. 4. Nicholson, W. K. (2020). Linear algebra with applications. 5. Spiegel, M. R., Lipschutz, S., Schiller, J. J., & Spellman, D. (2017). Schaum's outline of Complex Variables. McGraw Hill Professional. 6. Spiegel, M. R., Lipschutz, S., & Spellman, D. (2017). Vector Analysis	

Course Outcomes:	On completion of the course, students will be able to : 1. Infer the truth of various sentences and its equivalents and outline various properties of sets. 2. Examine and Identify the types of relations and functions. 3. Make use of the strong and weak induction. 4. Solve systems of linear equations. 5. Discuss the properties of determinants.

Minor Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-111
Title of the Course : Problem Solving and Program Design
Number of Credits : 4(3T+1P)
Effective from AY : 2024-25

Pre-requisites for the Course:	NIL	
Course Objectives:	1. To introduce the learner to various concepts and the problem solving process 2. To gain an understanding and attempt solving problems using various tools and techniques 3. To solve complex and/or large problems using composite data structures 4. To be able to design and build computer programs given a problem	
Units	**Content**	**No. of hours**
I	**Basic Concepts** ● Problems solving process; Solution types; Solving by computer ● Constants & Variables; Data Types & Storage; Functions; Operators; Expression & Equations ● Communicate with computer; methods to organize solution; solution planning tools; solution test & code; SDLC	**15T**
II	**Problem Solving Techniques** ● Modules design; Cohesion-Coupling; variable scope, parameters & return value ● Sequential Problem Solving by charts, algorithmic instruction, flowchart symbols and pseudocode ● Decision Problem Solving by various logic types; logic conversion; introduction to decision tables ● Iterative Problem Solving; loop vs go-to; incrementing & accumulating; loop types; Recursion	**15T**
III	**Problem Solving with Composite Data Structures** ● Composite data structures; heterogeneous data in structures; dynamic-sized structures ● Lists (or arrays); element & index; two-dimensional & higher; row/column-major storage; map ● Introduction to linked list, graphs, trees ● Introduction to DBMS, class/object & file processing	**15T**

IV	**Practical Work** The assignments in the practicals will reinforce problem solving, programming logic and program design concepts. It is recommended to use building block tools like PictoBlox, Scratch, TurtleArt, etc. and programming learning languages Kojo, Racket (DrRacket), etc. Following is a suggested (non-exhaustive) list of assignments:- ● Creating simple geometric lines and shapes ● Creating complex geometric shapes ● Creating shapes for alphabets and numbers ● Creating simple drawings of everyday objects ● Moving character(s) on keyboard arrow ● Demonstrating collision between character(s) ● Creating a simple (points/score)-based game ● Write scripts to perform arithmetic operations ● Write scripts to demonstrate logical operations ● Write scripts to perform string operations ● Write scripts to calculate prime number(s), factorial(s), armstrong number(s), etc. ● Write scripts to perform geometric operations ● Write scripts to perform operation(s) on finite and infinite series' ● Write scripts demonstrating use of parentheses, and nested expressions and invocations ● Write scripts for simple games	**30P**
Pedagogy:	Lectures, Tutorials. Assignments, Seminars, Presentations or a combination of some of these can also be used. ICT mode should be preferred. Sessions should be interactive in nature to enable peer group learning.	
References/ Readings:	1. Abelson, H., & Sussman, G. J. (1996). Structure and interpretation of computer programs (p. 688). The MIT Press. 2. Felleisen, M., Findler, R. B., Flatt, M., & Krishnamurthi, S. (2018). How to design programs: an introduction to programming and computing. MIT Press. 3. Kuppuswamy, S., Malliga, S., Kanimozhi Selvi, C. S., & Kousalya, K. (2019). Problem Solving and Programming. Tata McGraw Hill. 4. Maureen Sprankle, Jim Hubbard (2013). Problem Solving and Programming Concepts. Pearson Education India.	
Course Outcomes:	On completion of the course, students will be able to : 1. Remember the basic concepts & terminologies of problem solving, algorithms, flowcharts, pseudo-code, and debugging. 2. Understand basic computing concepts, algorithm design, flowchart design, pseudo-code, programming constructs, and debugging.	

	3. Apply problem solving & programming concepts in designing solutions to simpler problems using algorithm, flowchart and pseudocode. 4. Analyze a problem and apply the concepts learnt to develop & design solutions and programs

Multi-disciplinary Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-131
Title of the Course : E-Commerce
Number of Credits : 3
Effective from AY : 2024-25

Pre-requisites for the Course:	NIL	
Course Objectives:	1. To give fundamental understanding of e-commerce and online marketing 2. To instill ideas of Search Engine Optimization and Marketing, Applications of e- commerce and digital payments 3. To identify, define and differentiate the e-commerce models and risks of electronic commerce	
Units	**Content**	**No. of hours**
I	**Introduction to Electronic Commerce** ● Basic Understanding of E-Commerce ● History of e-commerce ● Business applications of e-commerce ● E-Commerce Models (B2B, B2C, C2C, B2G) ● Applications of M-Commerce **E-Commerce on Internet** ● Websites as marketplace ● Role of website in B2C e-commerce ● Website design principles ● Alternative methods of customer communication **Online Marketing** ● Online marketing and advertising ● Push and pull approaches ● Web counters, Web advertisements ● Content marketing ● Need of Digital Marketing for a business	**15**

II	**Search Engine Optimization** ● Search Engine Optimization (SEO), Search Engine Marketing (SEM) ● Social Media Marketing (SMM), Web Analytics **Applications of E-commerce** ● Applications of e-commerce to Supply chain management ● Applications of e-commerce to Customer Relationship Management ● Product and service digitization, Remote servicing **Electronic Payment System** ● Types of payment systems, credit cards, debit cards, mobile, etc., Electronic Fund Transfer (EFT) ● Operational credit and legal risk of e-payment, Risk management options for e-payment systems	**15**
III	**Business to Consumer E-Commerce** ● Cataloging ● Order planning and order generation, Cost estimation and pricing ● Order receipt and accounting, Order selection and prioritization ● Order scheduling, Order fulfilling, Order delivery ● Order billing, Post sales service **Business to Business E-Commerce** ● Need and Models of B2B e-commerce ● Using public and private computer networks for B2B trading ● EDI and paperless trading, Characteristic features of EDI service arrangement, EDI architecture and standards **Security Issues in E-Commerce** ● Risks of e-commerce ● Types and sources of threats; Security tools ● Risk management approaches	**15**
Pedagogy:	Lectures, Tutorials. Assignments, Seminars, Presentations or a combination of some of these can also be used. ICT mode should be preferred. Sessions should be interactive in nature to enable peer group learning.	
References/ Reading:	1. Kalakota, Ravi, Andrew Whinston (2015). Frontiers of Electronic Commerce. Pearson Education 2. P. T. Joseph (2015). E-Commerce: An Indian Perspective Paperback. PHI Learning. 3. V. Rajaraman (2015). Essentials of E-Commerce Technology. PHI Learning.	

Course Outcomes:	On completion of the course, students will be able to: 1. Understand the foundation of e-commerce, e-commerce websites and Online Marketing and Security Issues 2. Explain the importance of Search Engine Optimization, Applications of E-commerce and Electronic Payment Systems 3. Have an in-depth understanding of B2B and B2C e-commerce models

Skill Enhancement Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-141
Title of the Course : Programming in R
Number of Credits : 3 (1T+2P)
Effective from AY : 2024-25

Pre-requisites for the Course:	NIL	
Course Objectives:	1. To understand critical programming language concepts 2. To make use of R loop functions and debugging tools 3. To configure and use various libraries & packages with R	
Units	**Content**	**No. of hours**
I	**The R programming language** ● R language; variables, constants & operators; data types, objects & in-built library calls ● Decision & iterative structures; functions ● Vectors, lists, matrices; arrays; data frames, factors ● Visualization using various charts/plots ● Statistical calculations ● Packages & libraries with R	**15T**
II	**Practical Work (Basic Programming in R)** ● Assignments to practice arithmetic operations and expressions ● Assignments to practice working with an infinite series ● Assignments to practice working with arrays, matrix, vectors, list, factors ● Assignments to practice with dataframe ● Assignments to work with file processing ● Assignments to practice with various statistical calculations	**30P**
III	**Practical Work (Visualization, Packages in R)** ● Assignments to practice with various charts / graphs ● Assignments to use package(s) related to loading, manipulating & modeling data ● Assignments to use package(s) related to visualizing & reporting data ● Assignments to use package(s) related to work with spatial, time-series and financial data	**30P**
Pedagogy :	Lectures, Tutorials. Assignments, Seminars, Presentations or a combination of some of these can also be used. ICT mode should be preferred. Sessions should be interactive in nature to enable peer group learning.	

References/ Reading:	1. Crawley, M. J. (2012). The R book. John Wiley & Sons. 2. Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023). R for data science. " O'Reilly Media, Inc.". 3. https://cran.r-project.org/manuals.html [Accessed: Jan 20, 2024]
Course Outcomes:	On completion of the course, students will be able to : 1. Understand the various aspects and concepts in the R programming language 2. Write code in R to load, manipulate, and visualize data 3. Leverage using various packages/libraries of R for various data science related work

SEMESTER II

Major Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-102
Title of the Course : Data Management
Number of Credits : 2T+2P
Effective from AY : 2025-26

Pre-requisites for the Course:	**NIL**	
Course Objective :	● Enables the learner to understand the different issues involved in the design and implementation of a database system ● Provides both theoretical knowledge and practical skills required in the creation and management of a Relational DataBase Management System.	
Units	**Content**	**No. of hours**
I	**Foundational Concepts** ● Understanding Data; Database (DB) & DB Users; Characteristics; Structured and Unstructured Databases ● DB Architectures; Data Models; Schemas & Instances ● Data Modeling using the ER approach ● Relational Data Model & Introduction to Relational Algebra ● Relational DB Design; Functional Dependencies; ● Introduction to Normalization & Basic Normal Forms ● Basic structure of Database Queries ● SQL - DDL ● Set Operations ● Null Values and Aggregate Functions	**15T**
II	**Advanced SQL** ● Nested Subqueries ● Joins ● Correlated Subqueries ● Database Modification ● Introduction to Advanced SQL	**15T**

III	**Practical Work Foundational Concepts** ● Getting Started ○ Installation of DBMS Software(s) ○ Creation/modification of database tables using DDL statements and GUI tools of the DBMS software ○ Populating tables, Adding Primary Keys, Candidate Keys, Foreign Keys and other Integrity Constraints to relations ○ Updating and Deleting Rows of tables ○ Altering schema of existing tables ○ Writing simple SQL queries ● Querying Data ○ Querying the data dictionary ○ Various Operators ○ Using alias to control column headings ○ Using compound clauses and wildcards ○ Sorting data ○ Arithmetic Operators and Built-in Functions	**30P**
IV	**Practical Work Advanced SQL** ● Writing Complex Queries ○ Joins ○ Aggregate Functions ○ Group by and Having Clause ○ Subqueries – single row and multiple rows ○ Correlated Subqueries	**60P**
Pedagogy :	Lectures, Tutorials. Assignments, Seminars, Presentations or a combination of some of these can also be used. ICT mode should be preferred. Sessions should be interactive in nature to enable peer group learning.	
References/ Reading :	1. Connolly, T. M., & Begg, C. E. (2005). Database systems: a practical approach to design, implementation, and management. Pearson Education. 2. Elmasri, R., Navathe, S. B., Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems. (7th Edition). Pearson Publisher 3. Garcia-Molina, H. (2008). Database systems: the complete book. Pearson Education India. 4. Ramakrishnan, R., Gehrke, J., & Gehrke, J. (2003). Database management systems (Vol. 3). New York: McGraw-Hill. 5. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). Database system concepts. (7th Edition). McGraw-Hill.	

Course Outcomes:	On completion of the course, students will be able to : CO1: Understand and evaluate the role of a DBMS in Organizations. CO2: Recognize a good design and understand the relational database design principles. CO3: Use logical design methods used in the design of DB applications. CO4: Understand the basics of SQL and construct queries using SQL.

Minor Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-112
Title of the Course : Fundamentals of Statistics
Number of Credits : 4
Effective from AY : 2024-25

Pre-requisites for the Course:	NIL	
Course Objectives:	To introduce the students to 1. organize, summarize and analyze data 2. draw appropriate conclusions from data 3. apply statistical tools and techniques to real-world problems	
Units	**Content**	**No. of hours**
I	**Introductory concepts**: Definition and scope of Statistics; Concept of population and sample. **Types of data**: Quantitative; Qualitative; Attributes; Variates. **Tabulation of data**: Class intervals; Frequency tables. **Presentation of data**: Diagrams and graphs: Bar diagrams and their types; Pie charts; Frequency polygon; Histogram; Ogives. Consistency and independence of data with special reference to attributes. **Scales of measurement**: Nominal, Ordinal, Interval, Ratio. **Measures of Central Tendency**: Mathematical and Positional – Mean, Median, Mode, Quartiles, Percentiles. **Measures of Dispersion**: Range, Quartile deviation, Standard deviation, Coefficient of variation.	**15**
II	**Bivariate data**: Definition; Scatter diagram. **Correlation and Regression**: Simple, Partial and Multiple Correlation (3 variables only); Rank correlation; Simple linear regression.	**15**
III	**Probability**: Introduction; Random experiments; Sample space; Events and algebra of events; Definitions of Probability – Classical, Statistical, and Axiomatic; Conditional Probability; Addition and Multiplication theorem of probability; Independent events; Theorem of Total probability; Bayes' theorem and its applications.	**15**

IV	**Statistical Quality Control**: Introduction; Causes of variation in quality; Objective, advantages, and techniques of SQC. **Attribute data**: P chart, U chart, C chart. **Numerical data**: X bar chart, R bar chart, S bar chart. **Sampling techniques**: Various methods of data collection; Census survey and sample survey. **Sampling Methods**: Simple random sampling; Systematic sampling; Stratified sampling; Clustered sampling. **Non – probability Sampling Methods**: Convenience sampling; Consecutive sampling; Quota sampling; Purposive or Judgmental sampling; Snowball sampling.	**15**
Pedagogy:	Lectures/ Case Analysis/ Assignments/ Classroom Interaction/ Quiz	
References/ Reading:	1. Bruce, P., Bruce, A., & Gedeck, P. (2020). Practical statistics for data scientists: 50+ essential concepts using R and Python. O'Reilly Media. 2. Goon, A. M., Gupta, M. K., & Dasgupta, B. (2016). Fundamentals of Statistics. Vol 1. World Press Private Limited. 3. Gupta, S. C. (2018). Fundamentals of statistics. Himalaya Pub. 4. Gupta, S. C., & Kapoor, V. K. (2020). Fundamentals of mathematical statistics. Sultan Chand & Sons. 5. Gupta, S. P. (2017). Statistical methods. Sultan Chand & Sons.	
Course Outcomes:	On completion of the course, students will be able to: CO 1. Interpret data and graphically represent it CO 2. Calculate measures of central tendencies and variations CO 3. Analyze correlation and regression CO 4. Solve problems in Probability theory CO 5. Understand different data sampling techniques CO 6. Apply statistical quality control	

Multi-disciplinary Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-132
Title of the Course : Office Software Automation
Number of Credits : 3
Effective from AY : 2024-25

Pre-requisites for the Course:	NIL	
Course Objectives:	1. To understand the basics of office automation software and its applications. 2. To develop proficiency in using word processing, spreadsheet, and presentation software. 3. To diagnose and troubleshoot common PC issues and optimize the performance of a PC.	
Units	**Content**	**No. of hours**
I	**Basic Concepts** ● Understanding office automation software and its applications; Types of office automation software ● Office Suites like (Libre Office, Microsoft Office Suite, Google Workspace) **Spreadsheets** ● Introduction to spreadsheet software (LibreOffice Calc, MS Excel, Google Sheet) ● Creating and formatting spreadsheets ● Working with formulas and functions ● Charts and graphs; Collaboration tools	**15**
II	**Word Processing** ● Introduction to word processing software (LibreOffice Writer, MS Word, Google Doc) ● Creating and formatting documents ● Working with templates ● Mail merge and labels; Collaboration tools **Presentation Software** ● Introduction to presentation software (LibreOffice Impress, MS Powerpoint, Google Slide) ● Creating and formatting presentations ● Working with images, videos, and animations ● Collaboration tools	**15**

III	**Internet and Email** ● Introduction to the Internet; Web Browsers; Searching Engines; Configuring web browsers ● Introduction to Email; Setting up and configuring email accounts; Composing and sending emails; Managing Email Accounts **PC Troubleshooting** ● Hardware Troubleshooting: Basic hardware components of a PC, Common hardware issues and their solutions, Maintenance and optimization of hardware ● Software Troubleshooting: Common software issues and their solutions, Malware and virus removal, System recovery and backups ● Network Troubleshooting	**15**
Pedagogy:	Lectures, Tutorials. Assignments, Seminars, Presentations or a combination of some of these can also be used. ICT mode should be preferred. Sessions should be interactive in nature to enable peer group learning.	
References/ Reading:	1. A+ Guide to IT Technical Support (MindTap Course List) by Jean Andrews 2. Discovering Computers 2022: Digital Technology, Data, and Devices by Misty E. Vermaat 3. Meyers, M. (2017). CompTIA A+ Certification All-in-One Exam Guide, Ninth Edition (Exams 220-901 & 220-902). McGraw Hill Professional. 4. Russel, C., & Hoque, M. R. (2018). Google Workspace for Dummies. John Wiley & Sons. 5. Shelly, G. B., & Vermaat, M. E. (2017). Microsoft Office 365 & Office 2016: Introductory. Cengage Learning.	
Course Outcomes:	On completion of the course, students will be able to: CO 1. To create and format documents, create and format tables and mail merge CO 2. Understand the use and various functions of spreadsheets CO 3. Apply the knowledge of tools to create effective presentations CO 4. Understand PC assembling and troubleshooting	

Skill Enhancement Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-142
Title of the Course : Programming in Python
Number of Credits : 3(1T+2P)
Effective from AY : 2024-25

Pre-requisites for the Course:	NIL	
Course Objectives:	1. To understand critical programming language concepts 2. To make use of python loop functions and debugging tools 3. To write program using class and objects 4. To configure and use various libraries & packages with python	
Units	**Content**	**No. of hours**
I	**The Python programming language** ● Python language; variables, constants & operators; data types, objects & in-built library calls ● Selection and Iterative Constructs: Writing conditions, IF-ELSE constructs Conditional operators, SWITCH ,WHILE and FOR loops, Use of BREAK and CONTINUE statements. Nested Loops ● Advance Data types: Lists, Tuples, Set, Dictionaries, Strings, Unicode, formatting strings, docString. Searching and sorting algorithms without using library functions. ● Modular Programming: Importance of User Defined Functions, Hierarchy charts, fan-in/out, cohesion and coupling and loosely coupled modules. Fan-in – Fan-out concepts. ● User Defined Functions: Local and Global Variables, Scoping Rules, Parameters & arguments. Function with variable arguments. Modules, packages, scope. Recursion & Recursive Functions. Recursive v/s Iterative Functions. ● Custom Data Types and File Management: Object of a Class and basic concept of classes & OOP, Files, Exceptions in file handling. ● Introduction to Packages: Python packages for plotting, mathematical computation & linear regression.	**15T**

II	**Practical Work (Basic Programming in Python)** ● Introduction to UNIX environment- Introduction to Fedora/Ubuntu, Basic directory and file handling commands, Editor (vi editor), man pages, installation of Python and Jupyter notebook. ● Assignments to practice decision control, branch and loop control structure ● Assignment to practice List, Set, Tuple, Dictionary & Strings ● Assignments to practice functions & Recursion ● Assignments to practice user-defined data types & file handling ● Assignments to work with reading and writing data to file with exception handling	**30P**
III	**Practical Work (Packages in Python)** ● Assignment to create custom datatype using classes and objects ● Assignments to use package(s) related to loading, manipulating data using pandas ● Assignment to use Linear regression ● Assignments to practice with various charts / graphs/3-D charts/boxplot	**30P**
Pedagogy:	Lectures, Tutorials. Assignments, Seminars, Presentations or a combination of some of these can also be used. ICT mode should be preferred. Sessions should be interactive in nature to enable peer group learning.	
References/ Reading:	1. Guttag, J. V. (2016). Introduction to computation and programming using Python: With application to understanding data. MIT press. 2. Kumar, N., & Taneja, S. (2018). Python Programming: A Modular Approach. 3. Sprankle, M., & Hubbard, J. (2012). Problem solving and programming concepts. Prentice Hall Press. 4. https://scikit-learn.org/stable/index.html [Accessed: Jan 20, 2024]	
Course Outcomes:	On completion of the course, students will be able to: CO 1. Analyze a given problem and develop a Python program to solve it. CO 2. Identify test cases for a given problem. CO 3. Understand, test, trace programs written in Python language. CO 4. Working with python Standard Libraries	

SEMESTER III

Major Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-200
Title of the Course : Introduction to Data Science
Number of Credits : 4
Effective from AY : 2025-26

Pre-requisites for the Course:	NIL	
Course Objectives:	To gain a thorough understanding of the data science fundamentals.	
Units	**Content**	**No. of hours**
I	Introduction - meaning, definition, significance, Role of data scientist, Data Science Process Life Cycle, Challenges in data science process, applications, the significance of domain knowledge in data science, the significance of mathematics in data science	**15**
II	Data - information vs data, structured and unstructured data, sources of data, data collection methods. Data preprocessing - cleaning, integration, transformation, reduction, discretization, feature selection and extraction, scaling, normalizing, dimensionality reduction.	**15**
III	Data analysis - descriptive analysis, diagnostic analysis, predictive analysis, exploratory analysis. Model building basics - linear regression, classification, trend analysis. Model Evaluation - accuracy, precision, recall, F1-score.	**15**
IV	Ethics, bias, and privacy in data science. Information security. Applying data science to real-world problems. Case studies from industry - Business, Healthcare, Finance, Education, Tourism, etc.	**15**
Pedagogy:	Lectures, Tutorials. Assignments, Seminars, Presentations or a combination of some of these can also be used. ICT mode should be preferred. Sessions should be interactive in nature to enable peer group learning.	

References/ Reading:	1. Kelleher, J. D., & Tierney, B. (2018). Data science. MIT press. 2. Pierson, L. (2021). Data science for dummies. John Wiley & Sons. 3. Blum, A., Hopcroft, J., & Kannan, R. (2020). Foundations of data science. Cambridge University Press.
Course Outcomes:	On completion of the course, students will be able to understand: CO 1. The role of data science in other domains. CO 2. The data science process/pipeline for a given problem. CO 3. Data analytics techniques to collect and process data. CO 4. The applications of data science in various sectors.

Name of the Programme : **Master of Science (Integrated) in Data Science**
Course Code : **IDS-201**
Title of the Course : **Linear Algebra**
Number of Credits : **4**
Effective from AY : **2025-26**

Pre-requisites for the Course:	NIL	
Course Objectives:	1. Understand fundamental concepts of linear algebra, including matrices, determinants, vector spaces, and transformations. 2. Develop problem-solving skills using methods like Gaussian elimination, eigenvalues, and singular value decomposition. 3. Apply linear algebra to real-world problems in computing, data science, and engineering.	
Units	**Content**	**No. of hours**
I	**Linear Systems:** Systems of linear equations, elementary row operations, rank of a matrix, Gaussian elimination, elementary matrices, inverse of a matrix. **Determinants:** Determinant as area and volume, determinant using permutations, properties of determinants, Singular and Non-singular matrices, Rank of a matrix	**15**
II	**Vector Spaces:** Vector Spaces over fields, subspaces, linear independence, bases, dimension. **Linear Transformations:** Linear transformations, algebra of linear transformations, Rank Nullity Theorem and applications, isomorphism, change of bases, transpose of a linear transformation.	**15**
III	**Inner Product Spaces:** Inner products, Gram-Schmidt orthogonalization, orthogonal projections and best approximation, linear functionals and adjoint operator, bilinear maps, quadratic forms, symmetric, hermitian, unitary and normal operators.	**15**

IV	**Eigenvalues and Eigenvectors:** Eigenvalues and eigenvectors, characteristic polynomial, minimal polynomials, Cayley-Hamilton Theorem, triangulation and diagonalization, Finite dimensional spectral theorem for normal operators, Singular Value Decomposition, Jordan canonical form.	**15**
Pedagogy:	Lectures/ Case Analysis/ Assignments/ Classroom Interaction/Quiz	
References/ Reading:	1. Lang, S. (1987). Linear algebra. Springer. 2. Hoffman, K., & Kunze, R. (1978). Linear algebra (2nd ed.). Prentice Hall India Learning Private Limited. 3. Axler, S. (1997). Linear algebra done right (2nd ed.). Springer.	
Course Outcomes:	On completion of the course, students will be able to : CO 1. Learn to use Gaussian elimination and matrix operations to solve systems of equations and find matrix properties. CO 2. Learn about vector spaces, bases, dimension, and linear transformations. CO 3. Compute determinants and use them to analyze matrices. CO 4. Use inner products, orthogonality, and projections in vector spaces. CO 5. Find eigenvalues, eigen vectors, and apply matrix factorizations like SVD and Jordan form.	

Minor Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-211
Title of the Course : Deductive and Inferential Mathematics
Number of Credits : 4
Effective from AY : 2025-26

Pre-requisites for the Course:	NIL	
Course Objectives:	1. To use logical reasoning to solve problems and prove statements. 2. To earn key mathematical concepts like logical equivalence and normal forms. 3. To apply probability and statistics to analyze data and make predictions.	
Units	**Content**	**No. of hours**
I	**Mathematical Logic& Proofs** An open sentence, a closed sentence, Definition of proposition or a Statement.Strong emphasis on the Distinction between Inclusive OR and Exclusive OR. Truth tables.Compound Proposition. Algorithms, Truth tables and Tautologies Equivalent statements (≡). Examples and important logical results. De Morgan Laws for negation.Conditional and Biconditionals, Arguments and Proofs	**15**
II	**Well-formed-formulae** Equivalence of formulae. Various laws governing the well-formed formulae. Duality law. Normal Form. Disjunctive normal form, conjunctive normal form, Principal disjunctive normal form, Principal conjunctive normal form. Propositional Calculus. Predicate Calculus. Predicate Formula. Equivalence of Predicate Formulae. Inference Theory	**15**
III	**Introduction to Inference Theory** Introduction to Probability Theory using Kolmogorov Technique: Definition of an experiment. Outcomes of an experiment. Outcomes which are not decomposable. Sample space as the set of all non-decomposable outcomes of an experiment.Event as any subset of the sample space of an experiment under consideration. Idea of variations. Standard deviation as the root mean square deviation with respect to the mean. Mathematical Expectation and Expected Values. Random Variables: Idea of Distribution of a Function. Some	**15**

	standard Distributions such as Binomial., Normal, Poisson and Exponential. Their standard properties with the stress on Normal Distribution. Use of Normal Distribution Table to solve problems.	
IV	**Logic Programming** Propositional Logic vs. Predicate Logic; Fundamentals of Logical Knowledge Representation (Facts, Rules, and Queries); Unification in logic programming (predicates, variables, constants, logical operators); Recursive rule definitions; solving logical inference using resolution; solving constraint satisfaction problems	**15**
Pedagogy:	Lectures/ Practice Problems/ Assignments/ Classroom Interaction/ Quiz	
References/ Reading:	1. Tarski, A. (1994). Introduction to Logic and to the Methodology of the Deductive Sciences (Vol. 24). Oxford university press. 2. Sahu, P. K., Pal, S. R., & Das, A. K. (2015). Estimation and inferential statistics (pp. 1-317). New Delhi: Springer. 3. Ben-Ari, M. (2012). Mathematical logic for computer science. Springer Science & Business Media. 4. Tremblay, J. P., & Manohar, R. (2017). Discrete mathematical structures with applications to computer science. McGraw-Hill, Inc.	
Course Outcomes:	On completion of the course, students will be able to: CO 1. Explore, conjecture, and reason logically to model and arrive at a solution to a given problem. CO 2. Apply different mathematical methods to solve problems effectively. CO 3. Learn decision-making in the presence of uncertainty and will learn to quantify the uncertainty in estimation /the decision	

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-212
Title of the Course : Discrete Mathematics
Number of Credits : 4
Effective from AY : 2025-26

Pre-requisites for the Course:	NIL	
Course Objectives:	1. To understand fundamental concepts of, sets, relations, and functions for mathematical reasoning. 2. To develop problem-solving skills using combinatorics, graph theory, and recursion. 3. To apply discrete mathematics to algorithms, data structures, and computational problems in computer science.	
Units	**Content**	**No. of hours**
I	**Mathematical Logic:** Introduction, Statements and Notation, Connectives, Well-formed formulas, Tautology, Duality law, Equivalence, Implication, Normal Forms, Functionally complete set of connectives, Inference Theory of Statement Calculus, Predicate Calculus, Inference theory of Predicate Calculus. **Set Theory:** Basic Concepts of Set Theory, Relations and Ordering, The Principle of Inclusion-Exclusion, Pigeonhole principle and its application, Functions, Composition of functions, Inverse Functions, Recursive Functions, Lattices and its properties.	**15**
II	**Algebraic Structures:** Examples and General Properties, Semi-groups and Monoids, Groups, Sub-groups, Homomorphism, Isomorphism, Cosets, Lagrange's Theorem, Group Actions, Cyclic Groups, Abelian Groups, Permutation Groups, Symmetry Groups, Properties of groups and their applications, Normal Subgroups, Quotient Groups, Applications in cryptography and coding theory.	**15**

III	**Elementary Combinatorics and Recurrence Relations** Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutations with Constrained Repetitions, Binomial Coefficients, The Binomial and Multinomial Theorems, Advanced Counting Techniques, Pigeonhole Principle, Inclusion-Exclusion Principle, Stirling Numbers, Recurrence Relations, Generating Functions of Sequences, Calculating Coefficients of Generating Functions, Solving Recurrence Relations by Substitution and Generating Functions, The Method of Characteristic Roots, Solutions of Inhomogeneous Recurrence Relations.	**15**
IV	**Graph Theory** Basic Concepts of Graphs, Types of Graphs, Isomorphism and Sub-graphs, Graph Representations (Adjacency Matrix, Adjacency List), Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler's Formula, Multigraphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four Color Problem, Graph Traversals (DFS, BFS), Network Flow, Bipartite Graphs, Graph Coloring, Applications of Graphs in Networks and Scheduling.	**15**
Pedagogy:	Lectures/Practice Problems/Assignments/ Classroom Interaction/ Quiz	
References/ Reading:	1. Tremblay, J. P., & Manohar, R. (2017). Discrete mathematical structures with applications to computer science. McGraw-Hill, Inc. 2. Kolman, B., Busby, R. C., & Ross, S. C. (2011). *Discrete mathematical structures*. PHI Learning. 3. Malik, D. S., & Sen, M. K. (2004). *Discrete mathematical structures: Theory & applications*. Thomson India Edition.	
Course Outcomes:	On completion of the course, students will be able to: CO 1. Identify and work with finite, infinite, and uncountable sets, apply mathematical induction, and use inclusion-exclusion principles. CO 2. Solve problems involving permutations, combinations, discrete, and conditional probability. CO 3. Apply properties of binary relations and functions, including in databases and job scheduling. CO 4. Work with graphs, Eulerian/Hamiltonian paths, spanning trees, and planar graphs. CO 5. Algebraic Structures and Boolean Algebra: Understand group theory, rings, fields, and manipulate Boolean functions and propositional calculus.	

Multi-disciplinary Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-231
Title of the Course : Website Design
Number of Credits : 3
Effective from AY : 2025-26

Pre-requisites for the Course:	NIL	
Course Objectives:	1. Understand website basics, design principles, and no-code tools. 2. Learn how to structure content, use media, and improve usability. 3. Optimize, maintain, and update websites for better user experience and visibility	
Units	**Content**	**No. of hours**
I	**Introduction to Website Design** 1. Website concept; types of websites; components of website and webpage; UX & UI; accessibility and mobile responsiveness 2. Principles of good design; wireframing and prototyping 3. No-code website builders; choosing/customizing template; adding pages	**15**
II	**Building and Structuring a Website** 1. Structuring a website with navigation menu, content writing and placement, organize sections 2. Adding basic interactive features like embedded media, contact forms, social media integrations 3. Publish/manage website by understanding domain/hosting, basics of SEO, and launching personal website	**15**
III	**Enhancing and Maintaining Websites** 1. Website optimization by improving website loading speed, and image/media optimization without quality loss 2. Security basics: Backups, SSL certificates, and privacy policies 3. Advanced Features like creating forms, pop-ups, and live chat, embedding Google Analytics for visitor insights, simple automation like email signup, appointment booking 4. Website maintenance by regular website updates, monitoring visitor engagement, digital marketing	**15**
Pedagogy:	Lectures, Tutorials. Assignments, Seminars, Presentations or a combination of some of these can also be used. ICT mode should be preferred. Sessions should be interactive in nature to enable peer group learning.	

References/ Reading:	1. Krug, S. (2014). Don't make me think, revisited: A common sense approach to web usability (3rd ed.). New Riders. 2. McFarland, D. S. (2021). CSS: The missing manual (5th ed.). O'Reilly Media. 3. Robbins, J. N. (2018). Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics (5th ed.). O'Reilly Media. 4. Van Duyne, D. K., Landay, J. A., & Hong, J. I. (2006). The design of sites: Patterns for creating winning websites (2nd ed.). Pearson Education. 5. Zeldman, J., & Marcotte, E. (2010). Designing with web standards (3rd ed.). New Riders.
Course Outcomes:	On completion of the course, students will be able to : CO 1. Design User-Friendly Websites by applying basic design and UX principles CO 2. Build and Customize Websites by use of no-code tools CO 3. Optimize Website Performance by improving speed, SEO, and accessibility CO 4. Publish and Manage Websites by connecting domains and maintaining updates.

Skill Enhancement Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-241
Title of the Course : Basic Toolkit for Research
Number of Credits : 3(1T+2P)
Effective from AY : 2025-26

Pre-requisites for the Course:	NIL	
Course Objectives:	1. To have basic knowledge of various software tools and platforms for data handling. 2. To gain hands-on experience with various software tools for research.	
Units	**Content**	**No. of hours (75) (15T+60P)**
I	**Markdown** Document structure; basic text formatting; paragraphs; headings; lists; links and images; code blocks; escape characters; HTML elements; converting markdown to HTML web pages. **Source Version Control** Version Control; introduction to SVN and Git; Git repositories; Git cloning, forks/branches; Git stash; Git pull requests; Git merge conflicts; Git pages. **Scientific Documents** TEX; LaTEX; documents, lists, enumeration, figures/tables; mathematics; symbols; referencing	**15**
II	**Practical Work** Markdown - Setting up portfolio, creating code chunk, formatting, adding links, images, tables, tables, mathematical formulae. Git basics - Setting up a repository, managing branches, remote repositories, analysing git history, push code to GitHub, cherry-pick a range of commits, using git to track changes	**30P**

III	**Practical Work** Latex - Document structure, lists and enumeration, inserting figures, inserting tables, including mathematical symbols and equations, cross-referencing, bibliography and citation, switching templates. IDEs (RStudio/Spyder) - Setting up of IDE, features, exploring the interface, installing and loading packages, using Markdown for documentation. Python Notebooks/ Google Colab: Installation, cell types Code, Markdown, Raw NBConvert, code and syntax highlighting, adding forms in colab, exporting to different formats, extensions, publishing and sharing the notebook using free GPU.	**30P**
Pedagogy:	Lectures, Practical sessions, Assignments, Seminars, Presentations, Mini-projects or a combination of some of these can also be used. ICT mode should be preferred. Sessions should be interactive in nature to enable peer group learning.	
References/ Reading:	1. Cone, M. (2020). Markdown guide. Independently Published. 2. Scott, C., & Ben, S. (2016). Pro Git. 3. Oetiker, T., Partl, H., Hyna, I., & Schlegl, E. (2014). The not so short introduction to LaTeX. Typeset LATEX 2ε. 4. Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023). R for data science. " O'Reilly Media, Inc.".	
Course Outcomes:	On completion of the course, students will be able to : CO 1. Use Markdown for effective documentation. CO 2. Use LaTex for research documentation CO 3. Collaborate on data analysis projects using version control tool. CO 4. Make use of IDEs and python notebooks for programming and research.	

SEMESTER IV

Major Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-202
Title of the Course : Data Modeling and Visualization
Number of Credits : 4 (3T+1P)
Effective from AY : 2025-26

Pre-requisites for the Course:	NIL	
Course Objectives:	1. To introduce the fundamental concepts of and principles of data modeling. 2. To understand the role of data visualization in data analysis.	
Units	**Content**	**No. of hours (75) (45T + 30P)**
I	Data modeling - Meaning, significance, lifecycle, types of models. Relational model - Overview, keys in relational database, data anomalies, normalization and denormalization. ER modeling - ER diagrams, Cardinality constraints, crow's foot notation, recursive relationships, cross entity dependencies.	**15**
II	Dimensional modeling - Star Schema vs. Snowflake Schema, fact tables, dimension tables, slow changing dimensions, data cubes and multidimensional modeling. NoSQL Data Modeling - NoSQL databases (Key-value stores, Document stores, Column-family stores, Graph databases), modeling for NoSQL vs. Relational databases, Schema design for MongoDB and Cassandra, scalability and consistency issues.	**15**
III	Visualization - Meaning, components, significance of visualization in data analysis, design and cognitive principles, ethics and best practices. Data visualization types - right graph for right data. Visualization tools - Overview of popular data visualization tools (Tableau, Power BI etc.), basics of report and dashboard creation, web-based interactive data visualization platforms.	**15**

IV	**Practical Work** Lab assignments/exercises based on a data modeling-visualisation platform like PowerBI/Tableau. Overview - Installation and set up, connecting to data. Modeling - Building a data model for a given business scenario. Charts - bar chart, line graph, pie chart, bubble chart, heat map. Calculations - sum, average, aggregation, custom calculations. Structuring data - sort, filter, pivot, custom filters, tooltips, color formatting. Dashboards - storytelling, design for different displays, interactive dashboards for a business scenario, dashboard sharing .	**30P**
Pedagogy :	Lectures, Practical sessions, Assignments, Seminars, Presentations, Mini-projects or a combination of some of these can also be used. ICT mode should be preferred. Sessions should be interactive in nature to enable peer group learning.	
References/ Reading :	1. Simsion, G., & Witt, G. (2004). Data modeling essentials. Elsevier. 2. Healy, K. (2018). Data visualization: a practical introduction. Princeton University Press. 3. Ware, C. (2019). Information visualization: perception for design. Morgan Kaufmann. 4. Ferrari, A., & Russo, M. (2016). Introducing Microsoft Power BI. Microsoft Press.	
Course Outcomes :	On completion of the course, students will be able to: CO 1. Understand data modeling principles and the usage of corresponding tools. CO 2. Build data models for real world business scenarios. CO 3. Understand the principles of effective data visualization. CO 4. Choose appropriate visualizations for different types of data.	

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-203
Title of the Course : Linear Programming and Optimization
Number of Credits : 4
Effective from AY : 2025-26

Pre-requisites for the Course:	A basic understanding of linear algebra and calculus techniques for optimization, including matrix operations, optimality conditions. Familiarity with at least one software tool for numerical computations, such as Excel, R, MATLAB, Python, or an equivalent platform.	
Course Objectives:	1. To understand and apply problem-solving techniques both theoretically and graphically. 2. To develop the ability to analyze and interpret the results of mathematical models in real-world contexts. 3. To enhance problem-solving skills by applying optimization techniques to diverse scenarios.	
Units	**Content**	**No. of hours**
I	Linear Programming Introduction –Properties of Linear Programming-Basic assumptions-Mathematical formulation of Linear Programming. Limitations or constraints-Methods for the solution of LP	**15**
II	Problem-Graphical analysis of LP-Graphical LP Maximization problem-Graphical LP Minimization problem. Simplex method and its variant: Simplex method, Artificial variable techniques- Two Phase Method; M-Charnes Method, Big M Method. Special cases in LPP.	**15**
III	Duality: Definition of the dual problem, Primal-dual relationships, Economic Interpretation of Duality, Dual simplex Method.	**15**
IV	Sensitivity analysis: Changes in cost and resource vector. Transportation and Assignment Models Introduction:Transportation problem - Balanced - Unbalanced - Methods of basic feasible solution Optimal solution-MODI method. Assignment problem-Hungarian Method.	**15**
Pedagogy:	Lectures/ Practice Problems/ Assignments/ Presentations	

References/ Reading:	1. Taha, H. A. (2008). Operations research: An introduction (8th ed.). Prentice Hall. 2. Hillier, F. S., & Lieberman, G. J. (2010). Introduction to operations research: Concepts and cases (9th ed.). Tata McGraw-Hill. 3. Hadley, G. (2002). Linear programming. Narosa (Reprint). 4. Ravindran, A., Phillips, D. T., & Solberg, J. J. (2005). Operations research: Principles and practice. John Wiley & Sons.
Course Outcomes:	On completion of the course, students will be able to : CO 1. Understand the fundamental concepts and principles of Linear Programming Problems (LPP). CO 2. Apply various LPP techniques to solve real-world problems. CO 3. Differentiate between different methods for solving LPPs based on the type of constraints and the number of variables. CO 4. Evaluate the importance of the solution in terms of uniqueness, bounds, and optimality. CO 5. Formulate mathematical models for management and technical problems using LPP concepts. CO 6. Develop an interest in solving transportation and assignment problems while understanding their physical significance.

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-204
Title of the Course : Fundamentals of Computing Systems Design
Number of Credits : 4
Effective from AY : 2025-26

Pre-requisites for the Course:	Knowledge of Programming	
Course Objectives:	1. To understand the key principles of computing system design, including application-level and system-level considerations. 2. To analyze and design modular, scalable, and efficient computing systems. 3. To explore performance estimation, fault tolerance, and security considerations in system design. 4. To gain hands-on experience through case studies and a team-based design project	
Units	**Content**	**No. of hours (60)**
I	Foundations of Computing System Design 1. Overview of computing systems; mobile computing; System design approach and key challenges 2. Applications; operating systems, and their interactions; Storage systems and file management; Computer architecture and hardware execution models	**15**
II	Modularity, Workloads & Performance Estimation 1. Modularity with clients and services; modularity with virtualization 2. Understanding different types of computing workloads 3. Performance estimation and optimization techniques	**15**
III	Networks, Fault Tolerance & Security 1. Networking fundamentals in system design; Networked applications and distributed systems 2. Designing fault-tolerant and resilient systems; Redundant system design principles 3. Enforcing information security in computing systems; Best practices for secure system architecture	**15**

IV	Case Studies & Project 1. Analysis of real-world systems such as YouTube, Twitter, Instagram, etc. 2. Design a small-scale computing system to address specific problems, so that the system is scalable, maintainable, and fault tolerant	**15**
Pedagogy:	Lectures, Tutorials. Assignments, Seminars, Presentations or a combination of some of these can also be used. ICT mode should be preferred. Sessions should be interactive in nature to enable peer group learning.	
References/ Reading :	1. Saltzer, J., & Kaashoek, M. F. (2009). Principles of computer system design (1st ed.). Elsevier. 2. Jain, R. (1991). The art of computer systems performance analysis. John Wiley & Sons. 3. Anderson, R. (2008). Security engineering: A guide to building dependable distributed systems (2nd ed.). John Wiley & Sons. 4. Perlman, R. (1999). Interconnections: Bridges, routers, switches, and internetworking protocols (2nd ed.). Addison-Wesley	
Course Outcomes :	On completion of the course, students will be able to: CO 1. Analyze computing system components and their interactions. CO 2. Evaluate system architectures for scalability and efficiency. CO 3. Apply fault tolerance, security, and high availability principles. CO 4. Create a small-scale computing system in a team project. CO 5. Justify system design decisions through real-world case studies.	

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-205
Title of the Course : LINUX Fundamentals
Number of Credits : 2
Effective from AY : 2025-26

Pre-requisites for the Course:	NIL	
Course Objectives:	To introduce students to the Linux operating system environment and provide knowledge of basic Linux commands and shell scripting	
Units	**Content**	**No. of hours (60)**
I	1. Introduction - Linux Installation 2. Directory handling - cd, mkdir, rmdir, mv, pwd 3. File manipulation - cat, cp, ls, mv, rm, chmod, chown, find, cat, more, head, tail, cmp, wc, touch, pr 4. General purpose - date, history, man, who, whoami, uptime, finger, cal, uname, tree, bc, tar, zip 5. String manipulation - grep, egrep, cut, paste, tr, sort, rev, awk, sed 6. Process utilities - ps, pid, ppid, tty, time, kill, exit 7. Network utilities - ping, ifconfig, netstat, hostname, traceroute, telnet, ssh, mount, network configuration	**30P**
II	1. Shell Programming basics: Interactive shell script, positional parameters, operators, decision constructs, iterative constructs, Meta characters. 2. Shell programming for file manipulation - display contents, renaming based on extension, concatenation, find and display hidden files, counting the number of files, red-only write only files, file conversion to HTML, automate backup, deletion based on conditions, archive and compress log files. 3. Shell programming for user info - check if user exists, is logged in, password generation with criteria, generate reports of login activity, frequently used commands by the users.	**30P**
Pedagogy:	Practical, tutorials, assignments, mini-projects, self-study	
References/ Reading:	1. Das, S. (2006). UNIX, Concepts and Applications. Tata McGraw-Hill. 2. Glass, G., & Ables, K. (2003). UNIX for Programmers and Users. Pearson Education India. 3. Raymond, E. S. (2003). The art of Unix programming. Addison-Wesley Professional.	

Course Outcomes:	On completion of the course, students will be able to : CO 1. Gain a deeper understanding of shell programming and its implications. CO 2. Implement efficient shell scripts on LINUX OS

Minor VET Courses

Name of the Programme : Master of Science (Integrated) in Data Science
Course Code : IDS-221
Title of the Course : Rapid App Development
Number of Credits : 4 (2T+2P)
Effective from AY : 2025-26

Pre-requisites for the Course:	NIL	
Course Objectives:	1. To introduce learners to rapid application development using no-code and low-code platforms for web, mobile, and IoT applications. 2. To equip learners with the skills to design, build, and deploy functional applications without extensive coding knowledge. 3. To enable learners to integrate external services, automate processes, and develop real-world projects using RAD tools.	
Units	**Content**	**No. of hours (75) (45T+30P)**
I	**Introduction to Rapid Application Development (RAD)** ● RAD principles and methodologies. ● Low-code vs. no-code development. ● Overview of web, mobile, and IoT applications. ● UI/UX principles for rapid development. ● Prototyping and wireframing tools. **No-Code/Low-Code Web App Development** ● Static vs. dynamic websites. ● Database-driven web applications. ● Using platforms for web development. ● Integrating APIs and automation tools.	**15T**
II	**No-Code/Low-Code Mobile App Development** ● Mobile app & development lifecycle. ● Drag-and-drop platforms for mobile app development. ● Integrating mobile sensors and cloud services. ● Exporting and publishing mobile apps. **No-Code/Low-Code IoT App Development** ● IoT application & development lifecycle. ● Platforms for IoT development. ● Integrating sensors and actuators. ● IoT app deployment considerations.	**15T**

III	**Practical Work (Web)** ● No-Code Web Development with a platform (like WordPress); setting up a website; themes, plugins, and customization; forms, databases and automation tools ● No-Code Web App Development with a platform (like Bubble, OutSystems, and Adalo); build simple web applications with drag-and-drop interfaces; connect web apps to external APIs **Practical Work (Mobile)** ● No-Code Mobile App Development with a platform (like MIT App Inventor); build a basic mobile app (UI, logic, and testing); integrate mobile sensors and cloud services ● Export/publish mobile apps ● Connecting mobile and web applications	**30P**
IV	**Practical Work (IoT)** ● IoT Application Development with a platform (like Pictoblox) for IoT and smart applications; connect sensors and microcontrollers; simple automation ● IoT app deployment considerations. **Practical Work (UI/UX Design)** ● UI/UX Design using design tools like Pencil/Figma/Sketch **Practical Work (Project)** ● Projection selection and planning, by brainstorming ideas, defining scope and feature and choosing the right platforms/tools for implementation ● Develop apps, using no-/low-code platforms, implement UI/logic/external integrations, test/debug applications ● Deployment/publish the app, for user feedback and peer review ● The project is ideally done in a group	**30P**
Pedagogy :	Lectures, Tutorials. Assignments, Seminars, Presentations or a combination of some of these can also be used. ICT mode should be preferred. Sessions should be interactive in nature to enable peer group learning.	

References/ Reading :	1. Simon, P. (2022). Low-Code/No-Code: Citizen Developers and the Surprising Future of Business Applications. Wiley. 2. Helzle, S. (2022). Low-Code Application Development with Appian. Packt Publishing. 3. Murru, E. (2020). Hands-On Low-Code Application Development with Salesforce. Packt Publishing. 4. Sabin-Wilson, L. (2024). WordPress All-in-one for Dummies. John Wiley & Sons. 5. Reynolds, M. (2021). No-Code AI and Machine Learning: Building AI Applications Without Coding. Apress. 6. STEMpedia. (2023). Pictoblox for Beginners: AI, IoT, and Robotics with No-Code Programming. STEMpedia Press.
Course Outcomes:	On completion of the course, students will be able to : CO1: Describe the principles and advantages of rapid application development using no-code and low-code tools. CO2: Demonstrate the ability to create web applications using no-code/low-code platforms. CO3: Develop functional mobile applications using platforms like MIT App Inventor. CO4: Integrate APIs, automation tools, and external services into IoT/mobile applications. CO5: Deploy and present a working app prototype based on a real-world use case.