IPS Academy, Institute of Engineering & Science

(A UGC Autonomous Institute, Affiliated to RGPV, Bhopal) Scheme Based on AICTE Flexible Curriculum

Department of Computer Science & Engineering Bachelor of Technology (B. Tech.) [Computer Science & Engineering (Data Science)]

V – Semester (Scheme & Syllabus)

S.No.	Course Type	Course Code	Course Title	Hrs./Week			Credits
				L	T	P	
1	**PCC**	**DS10**	**Database Management System**	**2**	**1**	**-**	**3**
2	**PCC**	**DS11**	**Operating System**	**2**	**1**	**-**	**3**
3	**PCC**	**DS12**	**Machine Learning-I**	**2**	**1**	**-**	**3**
4	**PEC**	**DS01**	**Professional Elective-I**	**3**	**-**	**-**	**3**
5	**HSMC**	**HS06**	**Humanities and Social Sciences Open Courses - II**	**2**	**-**	**-**	**2**
6	**IFC**	**EC01**	**Interdisciplinary Foundation Course-II**	**2**	**-**	**-**	**2**
7	**LC**	**DS10(P)**	**Database Management System Lab**	**-**	**-**	**2**	**1**
8	**LC**	**DS11(P)**	**Operating System Lab**	**-**	**-**	**2**	**1**
9	**LC**	**DS12(P)**	**Machine Learning-I Lab**	**-**	**-**	**2**	**1**
10	**SBC**	**DS03(P)**	**JAVA-Programming**	**-**	**-**	**2**	**1**
11	**PROJ**	**DS01**	**Mini Project**	**-**	**-**	**4**	**2**
	LLC	**LLC03**	**Liberal Learning Course -III**	**Credits to be added in Sixth Semester**			
	PROJ	-	**Internship-I**	**Credit to be added in Sixth Semester**			
12	**MLC**	**MLC03**	**Environmental Studies**	**1**	**-**	**-**	**Audit**
			Total Credits	13	3	12	22
				30			

	Humanities and Social Sciences Open Courses (HSMC) – II, HS06 (Any One Course)	**Liberal learning Course-III, LLC03 (Any One Course**
	(A) Industrial Safety Psychology	**(A) Sociology**
	(B) Business Communication	**(B) Interior Design**
	(C) Project Management	**(C) Graphic Design**
		(D) Animation
		(E) Corporate Culture
	Professional Elective Course(PEC) –I,DS01 (Any One Course)	
	(A) Software Engineering and Project Management	**(C) Agile Software Development**
	(B) Data Warehousing and Mining	**(D) Theory of Computation**

PCC- DS10	**Database Management System**	**2L:1T:0P (4 hrs.)**	**Credits:03**

Course Objective:

The main objective of this course is to understand fundamental of database management system.

Course Content (40hrs)

Module 1: **(07hrs.)**

Introduction to DBMS, File system vs DBMS, Advantages of database systems, Database System architecture, Data models, Schemas and instances, Data independence, Functions of DBA and designer, Design issues, Entity-Relationship model: Basic concepts, Design process, E-R diagrams, weak entity sets, extended E-R features – generalization, specialization and aggregation.

Module2: **(11hrs.)**

Structure of relational databases, Relation algebra fundamental operators and syntax, relational algebra queries, Tuple calculus, Keys, Types of Keys.

SQL: Data retrieval queries, Data extraction from single, multiple tables equi- join, non equi-join, self - join, outer join. Usage of like, any, all, exists, in Special operators. DDL, DML, integrity constraints, Complex queries, Referential integrity,

Module3: **(8hrs.)**

Data Base Design: Introduction to normalization, Normal forms, Functional dependency, Types of FD, closure of attributes, irreducible set of FD, Decomposition, Dependency preservation and lossless join, problems with null valued and dangling tuples, multivalued dependencies.

Module4: **(08 hrs.)**

Transaction Processing Concepts: -Transaction system, ACID Property, Testing of Serializability, Serializability of schedules, conflict & view Serializable schedule, recoverability, Recovery from transaction failures. Log based recovery. Concurrency Control Techniques: Concurrency Control, locking Techniques for concurrency control, time stamping protocols for concurrency control, validation-based protocol, Recovery with concurrent transaction.

Module 5: **(06hrs.)**

Data Storage and Indexes - file organizations, primary, secondary index structures, various index structures - hash-based, dynamic hashing techniques, multi-level indexes, B and B+ trees. Cursor management: nested and parameterized cursors, Triggers, Types of triggers.

Course Outcomes:

1. Describe basic concepts of DBMS and Design ER model.
2. Solve queries using Relational Algebra and SQL.
3. Understanding of functional dependencies and Explain normalization by applying Normalizations techniques to database.
4. Explain basic the fundamental concepts of transaction processing and concurrency control.
5. Understanding various storage systems and indexing.

List of Text/Reference Books:

1. DateCJ, "An Introduction to Database System",Pearson Educations,8thEdition,2003.

2. Korth, Silbertz,Sudarshan, "Fundamental of Database System", McGrawHill,5thEdition,2006.

3. PeterRob,"DataBaseSystem:Design Implementation& Management",Cengage Learninig4thEdition,2000.

4. Elmasri, Navathe,"Fundamentals of Database Systems",PearsonEducations,7thEdition2017.

5 Atul Kahate,"Introduction to Database Management System",Pearson Educations,2004.

6. Oracle9i Database Administration Fundamental-I,VolumeI, OraclePress,TMH.

7. Paneerselvam,"Database Management System", PHILearning,3rdEdition,2018.

8. J.D.Ullman,"Principles o fDatabase and Knowledge–BaseSystems",ComputerSciencePress,2ndEdition1988.

9. Serge Abiteboul, RichardHull, VictorVianu,"FoundationsofDatabases",Addison-Wesley,1995.

PCC- DS11	**Operating System**	**2L:1T:0P(4Hr)**	**Credits:03**

Course Objective:

This Course provides a comprehensive introduction of Operating System, Process Management, Memory Management, File Management and I/O management.

Course Contents: (38Hrs)

Module 1: **(06hrs.)**

Introduction to Operating Systems: Function, Evolution, Different Types, Desirable Characteristics and features of an O/S, Operating Systems Services: Types of Services, Different ways of providing these Services – Utility Programs, System Calls, Operating System Structure, and Spooling & Buffering.

Module2: **(08 hrs.)**

CPU Scheduling : Process Concept, Scheduling Concepts, Types of Schedulers, Scheduling Criteria, Process State Diagram, Scheduling Algorithms, Operation on Process, Algorithms Evaluation, System calls for Process Management; Multiple Processor Scheduling; Concept of Threads. Concurrent Processes : Real and Virtual Concurrency, Mutual Exclusion, Synchronization, Inter- Process Communication, Critical Section Problem, Solution to Critical Section Problem : Semaphores – Binary and Counting Semaphores, WAIT & SIGNAL Operations and their implementation. Deadlocks: Deadlock Problems, Characterization, Prevention, Avoidance, Recovery.

Module3: **(10hrs.)**

Memory Management: Different Memory Management Techniques – Partitioning, Swapping, Segmentation, Paging, Paged Segmentation, Comparison of these techniques, Techniques for supporting the execution of large programs: Overlay, Dynamic Linking and Loading, Virtual Memory – Concept, Implementation by Demand Paging etc., Page replacement algorithms.

Module4: **(08 hrs.)**

File Systems: File Concept, User’s and System Programmer’s view of File System, Disk Organization, Tape Organization, Different Modules of a File System, Disk Space Allocation Methods – Contiguous, Linked and Indexed. Directory Structures, File Protection, System Calls for File Management, Disk Scheduling Algorithms.

Module5: **(06hrs.)**

Introduction to Network, Distributed and Multiprocessor Operating Systems. Case Studies: Unix/Linux, WINDOWS and other Contemporary Operating Systems.

Course Outcomes:

1. State the core concepts of operating system, evolution and types of operating System.
2. Illustrate various input output concepts, inter process communication and deadlock
3. Illustrate process scheduling and memory management techniques.
4. Describe the concept of file and disk management.
5. State the core concepts of network, distributed and multiprocessor operating system.

List of Text/Reference Books:

1. Avi Silberschatz, Peter Galvin, Greg Gagne, “Operating System Concepts Essentials”, WileyAsia Student Edition, 10thEdition, 2018.
2. William Stallings,“Operating Systems :Internal sand Design Principles”,Prentice Hall of India, 5th Edition, 2005.
3. Charles Crowley,“Operating System:A Design-oriented Approach”,Irwin Publishing,1stEdition.
4. GaryJ.Nutt,“Operating Systems:A Modern Perspective”,Addison-Wesley,2nd Edition.
5. Maurice Bach,“Design of the Unix Operating Systems”,Prentice-Hallof India,8th

1.Edition.

6. DanielP.Bovet,Marco Cesati,“Understanding the Linux Kernel”,O'Reilly and Associates,3rd Edition.
7. AndrewS.Tanenbaum,“Modern Operating Systems”,Prentice Hall,3rd Edition,2007.
8. Bovet& Cesati,“Understanding the Linux Kernel”,O’Reily,3rd Edition.

PCC-DS12	Machine Learning-I	2L: 1T: 0P (4hrs.)	Credits:03

Prerequisite: Python, Statistics & Probability, Data Pre-processing

Course Objective:

The main objective of this course is to understand fundamental of Machine learning.

Course Contents:(38Hrs)

Module 1: **(06hrs.)**

Basic Mathematical and Statistical concepts: mean, median, mode, variance, co-variance, correlation, dispersion matrix, binomial distribution, normal distribution, multi-variate normal distribution, basic concepts in probability theory such as Bayes theorem, Chebyshev's inequality, Laws of large numbers, Central limit theorem, Unbiased estimate, consistent estimate.

Module2: **(08 hrs.)**

Introduction to ML: Environment Setup- Anaconda; AI vs ML, Fundamentals of Statistics, Regression- Linear Regression, Multiple Linear Regression, Logistic Regression; Optimisation of Regressions, Gradient Descent, Maximum likelihood estimation.

Module3: **(10hrs.)**

Classification: Bayes decision rule, examples, normal distribution cases, training and test sets, prob. of misclassification, estimation of parameters for normal distribution, minimum distance classifier, standardization, normalization, Mahalanobis distance, Naive-Bayes rule, K-NN decision rule, its properties, Density estimation, Perceptron (linear separable case), MLP, Assessment of classifiers.

Module4: **(08 hrs.)**

Unsupervised learning: Unsupervised vs. Supervised Learning, Similarity measures, minimum within cluster distance criterion, K-means algorithm, Hierarchical clustering, Density based clustering, FCM, cluster validation. DBSCAN clustering.

Module5: **(06hrs.)**

Dimensionality Reduction & Validation: What is cross validation, Boosting- Basics, equations, implementation on iris dataset, boosting vs bagging, Principal Component Analysis, Feature Selection, Extraction & their algorithms.

Course Outcomes:

1. Getting introduced to Machine Learning.
2. Building the foundation of Machine Learning
3. Understanding the classification models of Supervised Learning.
4. Understand unsupervised machine learning techniques.
5. Learning about Dimensionality Reduction & additional aspects of Machine Learning.

List of Text/Reference Books:

1. Introduction to Machine learning, Nils J.Nilsson.
2. Machine learning for dummies, IBM Limited ed, by Judith Hurwitz and Daniel Kirsch. Introduction to Machine Learning with Python A guide for data scientists, Andreas, C. Muller & Sarah Guido, O'Reilly.
3. [2022] Machine Learning and Deep Learning Bootcamp in Python: Course from Udemy.

PEC- DS01(A)	Software Engineering and Project Management	4L:0T:0P (4hrs.)	Credits:03

Course Objective:

1. The purpose of this subject is to cover the underlying concepts and techniques used in Software Engineering & Project Management

Course Contents:(43Hrs)

Module 1: (09hrs.)

The Software Product and Software Process: Software Product and Process Characteristics, Software Process Models: Linear Sequential Model, Prototyping Model, RAD Model, Evolutionary Process Models like Incremental Model,Spiral Model, Component Assembly Model, RUP and Agile processes. Software Process customization and improvement, CMM, Product and Process Metrics, Feasibility Analysis, Cost Estimation Model.

Module2: (10hrs.)

Requirement Elicitation, Analysis, and Specification: Functional and Non-functional requirements, Requirement Sources and Elicitation Techniques, Use case Modeling, System and Software Requirement Specifications, Requirement Validation, Traceability.

Module3: (10hrs.)

Software Design: The Software Design Process, Design Concepts and Principles, Software Modeling and UML, Architectural Design, Architectural Views and Styles, User Interface Design, Function- oriented Design, SA/SD Component Based Design and Design Metrics

Module4: (08 hrs.)

Software Analysis and Testing: Software Static and Dynamic analysis, Code inspections, Software Testing, Fundamentals, Software Test Process, Testing Levels, Test Criteria, Test Case Design, Test Oracles, Test Techniques, Black-Box Testing, White-Box Unit Testing and Unit, Testing Frameworks, Integration Testing, System Testing and other Specialized, Testing, Test Plan, Test Metrics, Testing Tools.

Module5: (06hrs.)

Software Maintenance & Software Project Measurement: Need and Types of Maintenance, Software Configuration Management (SCM), Software Change Management, Version Control, Change control and Reporting, Program Comprehension Techniques, Re-engineering, Reverse Engineering, Tool Support. Project Management Concepts, Project and Process Planning, Resources Allocations, Project Scheduling and Tracking, Risk Assessment and Mitigation, Software Quality Assurance (SQA). Project Plan, Project Metrics.

Course Outcomes:

1. Decompose the given project in various phases of a lifecycle.

2. Choose appropriate process model depending on the user requirements.

3. Perform various life cycle activities like Analysis, Design, Implementation, Testing & Maintenance.

4. Know various processes used in all the phases of the product.

5. Apply the knowledge, techniques, and skills in the development of a software product

List of Text/Reference Books:

1. Pankaj Jalote ,”An Integrated Approach to Software Engineering”, Narosa Pub, 2005

2. Rajib Mall, “Fundamentals of Software Engineering” Second Edition, PHILearning,Fouth Edition, 2014.

3. P, S. Pressman “Software Engineering. A Practitioner's Approach” New edition, McGraw Hills,7th edition,2010.

4. Sommerville,”Software Enginerring”,Pearson Education, 9 th Edition,2011.

5. Richard H.Thayer,”Software Enginerring & Project Managements”, WileyIndia

6. Waman S.Jawadekar,”Software Enginerring”, TMH,2004.

7. Bob Hughes, M.Cotterell, Rajib Mall “Software Project Management”, McGrawHill,Sixth Edition,2017

8. Schwalbe, Kathy “Information Technology Project Management” 8 th Edition, 2016.

9. Kieron Conway “Software project Management from concept to development Black Book” Dreamtech Press.

10.Deepak Jain, “Software Engineering principle and practices” Oxford University Press,2008.

11. Bell Douglas “Software Engineering for students” ,Pearson Education.,4thEdition,2005.

12. Kelkar “Software Project Management,” PHI Learning,3rd edition 2012.

PEC- DS01(B)	Data Warehousing and Mining	4L: 0T: 0P (4hrs)	Credits: 03

Prerequisite Course

Objective::

The main objective of this course is to understand fundamental of Data Warehousing and Data Mining.

Course Contents:(40hrs)

Module 1: **(06hrs.)**

Data warehouse: Introduction to Data warehouse, Difference between operational database systems and data warehouses, Data warehouse Characteristics, Data warehouse Architecture and its Components, Extraction- Transformation-Loading, Logical(Multi-Dimensional), Data Modeling, Schema Design, Star and Snow-Flake Schema, Fact Constellation, Fact Table, Fully Addictive, Semi Addictive, Non Addictive Measures; Fact-Less-Facts, Dimension Table Characteristics; OLAP Cube, OLAP Operations, OLAP Server Architecture-ROLAP, MOLAP and HOLAP.

Module2: **(10 hrs.)**

Introduction: Fundamentals of data mining, Data Mining Functionalities, Classification of Data Mining systems, Data Mining Task Primitives, Integration of a Data Mining System with a Databaseor Data Warehouse System, Major issues in Data Mining. Data Preprocessing: Need for Preprocessing the Data, Data Cleaning, Data Integration &Transformation, Data Reduction, Discretization and Concept Hierarchy Generation.

Module3: **(10hrs.)**

Association Rules: Problem Definition, Frequent Item Set Generation, The APRIORI Principle, Support and Confidence Measures, Association Rule Generation; APRIOIRI Algorithm, The Partition Algorithms, FP-Growth Algorithms, Compact Representation of Frequent Item Set- Maximal Frequent
Item Set, Closed Frequent Item Set.

Module4: **(08 hrs.)**

Web Scraping: What is web scraping, legality of web scraping, web scraping vs data mining, basics of web scraping, useful libraries (selenium, pandas, beautiful soap) and their related processes, Demo case study (flipkart). Fundamentals of Scrapy.

Module5: **(06hrs.)**

Data Collection Methodologies: What is data collection, primary data collection methods: quantitative & qualitative collection methods, secondary data collection methods: published & unpublished data, tools of data collection. Collecting data through google forms & webpages.

Course Outcomes:

1. Understand why the data warehouse in addition to database systems.
2. Understand data mining and Pre-processing of data.
3. Identify the association rules.
4. Understand different classification in large data sets.
5. Understand different clustering in large data sets.

List of Text/Reference Books:

1. Data Mining- Concepts and -techniques- Jiawei Han, Micheline Kamber, Morgan Kaufmann Publishers, Elsevier, 2 Edition, 2006.

2. Introduction to Data Mining, Psng-Ning Tan, Vipin Kumar, Michael Steinbanch, Pearson .

3. Data Mining Techniques, Arun K Pujari, 3rd Edition, Universities Press.

4. Data Warehousing Fundament's, Pualraj Ponnaiah, Wiley Student Edition.

5. The Data Warehouse Life Cycle Toolkit — Ralph Kimball, Wiley StudentEdition .

6. Data Mining, Vikaram Pudi,P Radha Krishna, Oxford University Press.

7. Scrapy documentation.

8. Scrapy tutorial.

9. Web Crawling with Beautiful Soup.

PEC- DS01	Agile Software Development	4L: 0T: 0P (4hrs.)	Credits: 03

Prerequisite:

Course Objective:

To learn best practices and methods of software development .

Course Contents:(38hrs)

Module 1: **(06hrs.)**

Fundamentals of Agile Process: Introduction and background, Agile Manifesto and Principles, Stakeholders and Challenges, Overview of Agile Development Models: Scrum, Extreme Programming, Feature Driven Development, Crystal, Kanban, and Lean Software Development.

Module2: **(10 hrs.)**

Agile Projects: Planning for Agile Teams: Scrum Teams, XP Teams, General Agile Teams, Team Distribution; Agile Project Lifecycles: Typical Agile Project Lifecycles, Phase Activities, Product Vision, Release Planning: Creating the Product Backlog, User Stories, Prioritizing and Estimating, Creating the Release Plan; Monitoring and Adapting: Managing Risks and Issues, Retrospectives.

Module3: **(10hrs.)**

Introduction to Scrum: Agile Scrum Framework, Scrum Artifacts, Meetings, Activities and Roles, Scrum Team Simulation, Scrum Planning Principles, Product and Release Planning, Sprinting: Planning, Execution, Review and Retrospective; User story definition and Characteristics, Acceptance tests and Verifying stories, Burn down chart, Daily scrum, Scrum Case Study.

Module4: **(08 hrs.)**

Introduction to Extreme Programming (XP): XP Lifecycle, The XP Team, XP Concepts: Refactoring, Technical Debt, Timeboxing, Stories, Velocity; Adopting XP: Pre-requisites, Challenges; Applying XP: Thinking- Pair Programming, Collaborating, Release, Planning, Development; XP Case Study.

Module5: **(06hrs.)**

Agile Software Design and Development: Agile design practices, Role of design Principles, Need and significance of Refactoring, Refactoring Techniques, Continuous Integration, Automated build tools, Version control; Agility and Quality Assurance: Agile Interaction Design, Agile approach to Quality Assurance, Test Driven Development, Pair programming: Issues and Challenges.

Course Outcomes:

1. Describe the fundamental principles and practices associated with each of the agile development methods.
2. Explain the Agile project lifecycle.
3. Describe the Agile Scrum framework.
4. Apply core values and principles of Agile Methods in software development.
5. Apply core values and principles of Agile Methods in software development.

List of Text/Reference Books:

1. Robert C. Martin, “Agile Software Development- Principles, Patterns and Practices, Prentice Hall, 2013.

2. Kenneth S. Rubin, “EssentialScrum: A Practical Guide to the Most Popular Agile Process”, Addison Wesley, 2012.

3. James Shore and Shane Warden, “The Art of Agile Development”, O’ReillyMedia, 2007.

4. Craig Larman, “Agile and Iterative Development: Amanager’s Guide”, Addison-Wesley,2004.

5. Ken Schawber, Mike Beedle, “Agile Software Development with Scrum”, Pearson, 2001.

6. Cohn, Mike, “Agile Estimating and Planning”, PearsonEducation, 2006.

7. Cohn, Mike, “User Stories Applied: For Agile Software Development “Addison Wisley,2004.

PEC- DS01(D)	**Theory of Computation**	**4L: 0T: 0P (4 hrs.)**	**Credits: 03**

Prerequisite: Nil

Course Objective:

The main objective of this course is to understand fundamental of Theory of Computation.

Course Contents:(36hrs)

Module 1: **(06hrs.)**

Introduction of Automata Theory: Examples of automata machines, Finite Automata as a language acceptor and translator, Moore machines and mealy machines, composite machine, Conversion from Mealy to Moore and vice versa.

Module2: **(08hrs.)**

Types of Finite Automata: Non-Deterministic Finite Automata (NDFA), Deterministic finite automata machines, conversion of NDFA to DFA, minimization of automata machines, regular expression, Arden's theorem. Meaning of union, intersection, concatenation and closure, 2- way DFA.

Module3: **(08hrs.)**

Grammars: Types of grammar, context sensitive grammar, and context free grammar, regular grammar. Derivation trees, ambiguity in grammar, simplification of context free grammar, conversion of grammar to automata machine and vice versa, Chomsky hierarchy of grammar, killing null and unit productions. Chomsky normal form and Greibach normalform.

Module4: **(06hrs.)**

Push down Automata: example of PDA, deterministic and non-deterministic PDA, conversion of PDA into context free grammar and vice versa, CFG equivalent to PDA, Petrinet model.

Module5: **(08hrs.)**

Turing Machine: Techniques for construction. Universal Turing machine Multitap, multihead and multidimensional Turing machine, N-P complete problems. Decidability and Recursively Enumerable Languages, decidability, decidable languages, undecidable languages, Halting problem of Turing machine & the post correspondence problem.

Course Outcomes:

1. Explain the basic concepts of switching and finite automata theory & languages.
2. Relate practical problems to languages, automata, computability and complexity.
3. Construct abstract models of computing, check their power to recognize the languages and analyze the grammar, its types, simplification and normal form.
4. Interpret rigorously formal mathematical methods to prove properties of languages, grammars and automata.
5. Develop an overview of how automata theory, languages and computation are applicable in engineering application.

List of Text / Reference Books:

1. Daniel I.A. Cohen,"Introduction to Computer Theory",Wiley India, 2nd Edition,2003.

2. John E Hopcroft, Jeffrey D. Ullman and Rajeev Motwani, "Introduction to Automata Theory, Languages and Computation", Pearson Education, 2 nd Edition,2001.

3. K.L.P Mishra & N.Chandrasekaran,"Theory of Computer Science", PHI Learning, 3 rdEdition,2006.

4. Peter Linz, "Introduction to Automata Theory and Formal Languages", Narosa Publishing. 3 rdEdition,2007.

5. John C Martin, "Introduction to languages and the theory of computation", TATA McGraw Hill, 3 rd Edition 2013.

6. Harry R. Lewis and Christos H. Papadimitriou, "Elements of the Theory ofComputation", Pearson Education Asia,2nd edition, 1998.

7. Dexter C. Kozen, "Automata and Computability", Undergraduate Texts in ComputerScience, Springer,1st edition,2012.

8. Michael Sipser, "Introduction to the Theory of Computation", PWS Publishing., 3 rd edition,2012.

LC- DS10(P)	**Database Management System Lab**	**0L: 0T: 2P (2 hrs.)**	**Credits:01**

Prerequisite:

Course Objective:

To introduce solutions available for data storage, Core elements of a data center infrastructure, role of each element in supporting business activities

Course Contents:

Module 1:

Installation: SQL Server or Oracle or MySQL, Overview of using SQL tool, SQL Syntax, Data types in SQL: Numeric data types, Date and Time data types, Character and String data types, Unicode character string data types, Binary data types, and Miscellaneous data types, Creating Tables, Oracle and SQL Data Definition language (DDL), Data manipulation language (DML) Data Control language (DCL).

Module2:

DDL Commands, Create, Alter, Drop, Truncate, and Rename. Data Definition Language Operations, Create a Database, Use Database, Rename a Database, Drop Database, Add a Column to exiting Table, Add multiple columns to existing Table, Modify an existing column, Rename a Column, Drop a Column, Truncate a Table, and Drop a Table. Integrity Constraints

Module3:

DML Commands, select, insert, update, and delete. Data Manipulation Language Operations, retrieving data from a table, inserting data into a table, updating existing data into a table, and deleting all records from a table. logical operation (and, or, not), like, wildcards, in, between, aliases, any, all, union, intersect, SQL aggregate functions, count, sum, avg, max, min, clauses: order by, group by, having.

Module4:

Joins: inner join, left join, right join, full join, self-join, sub query: nested query. Correlated sub query exists, not exists, DCL commands such as grant and revoke, views: views creation and dropping. Introduction to PL/SQL, practicing on triggers - creation of trigger.

Module5:

Introduction to Basics of NoSQL and MongoDB, Overview of MongoDB Architecture, CRUD Operations in MongoDB

Course Outcomes:

1. Describe basic concepts of SQL and its data type.
2. Apply data definition language command and analyze its output.
3. Apply data manipulation language command and analyze its output.
4. Apply data control language command, PLSQL and analyze its output.
5. Understand the fundamentals of MongoDB, including its architecture and core concepts, and perform basic CRUD operations to manage databases and collections within a NoSQL environment.

List of Practical’s:

1. Introduction to Oracle and SQL
2. Write the queries for Data Definition language (DDL)
3. Write the queries for Data manipulation language (DML)
4. Use of various types of Integrity Constraints
5. Write the queries for Data Control language (DCL)
6. Use of SELECT command with different clauses.
7. Write SQL queries using logical operation (AND, OR, NOT)
8. Write SQL queries for aggregate functions (Max, Min, Sum, Avg, and Count)
9. Write SQL queries for group by and having.
10. Write SQL queries for sub queries and nested queries.
11. Write an SQL query to implement JOINS.
12. Write SQL queries to create views.
13. Write program by the use of PL/SQL
14. Installing and setting up MongoDB.
15. Creating and dropping databases & collections.

LC- DS11 (P)	**Operating System Lab**	**0L: 0T: 2P(2 hrs.)**	**Credits:01**

Course Objective: Implement and compare CPU scheduling, inter-process communication, memory management, disk scheduling algorithms, and the Banker's algorithm for deadlock avoidance.

Course Contents:

Module 1: CPU Scheduling Algorithms

1. To implement FCFS CPU scheduling algorithm.
2. To implement SJF CPU scheduling algorithm.
3. To implement Priority CPU Scheduling algorithm.
4. To implement Round Robin CPU scheduling algorithm.
5. To compare various CPU Scheduling Algorithms over different Scheduling Criteria

Module 2: Classical Inter-Process Communication Problems

1. To implement classical inter-process communication problem (producer consumer).
2. To implement classical inter-process communication problem (Reader Writers).
3. To implement classical inter-process communication problem (Dining Philosophers)

Module 3: Memory Management

1. To implement & compare various page replacement algorithms.

Module 4: Disk Scheduling

1. To implement & compare various Disk & Drum scheduling Algorithms.

Module5: Deadlock and Banker's Algorithm

1. To implement Banker’s algorithm.

List of Text/ Reference Books:

1. Avi Silberschatz , Peter Galvin, Greg Gagne, “Operating System Concepts Essentials ”,Wiley Asia Student Edition, 10th Edition, 2018.
2. William Stallings, “Operating Systems: Internals and Design Principles”, Prentice Hall of India, 5th Edition, 2005.
3. Charles Crowley, “Operating System: A Design-oriented Approach”, Irwin Publishing, 1st Edition.
4. Gary J. Nutt, “Operating Systems: A Modern Perspective”, Addison-Wesley, 2nd Edition.
5. Maurice Bach, “Design of the Unix Operating Systems”, Prentice-Hall of India, 8thEdition.
6. Daniel P. Bovet, Marco Cesati, “Understanding the Linux Kernel”, O'Reilly and Associates, 3rd Edition.
7. Andrew S. Tanenbaum , “Modern Operating Systems”, Prentice Hall, 3rd Edition,2007.
8. Bovet &Cesati, “Understanding the Linux Kernel”, O’Reily, 3rd Edition.

List of Experiment:

Write a program:

1. To implement FCFS CPU scheduling algorithm.
2. To implement SJF CPU scheduling algorithm.
3. To implement Priority CPU Scheduling algorithm.
4. To implement Round Robin CPU scheduling algorithm.
5. To compare various CPU Scheduling Algorithms over different Scheduling Criteria.
6. To implement classical inter process communication problem (producer consumer).
7. To implement classical inter process communication problem (Reader Writers).
8. To implement classical inter process communication problem (Dining Philosophers).
9. To implement &Compare various page replacement algorithms.
10. To implement & Compare various Disk & Drum scheduling Algorithms.
11. To implement Banker’s algorithms.

LC-DS12(P)	**Machine Learning Lab-I**	**0L: 0T: 2P (2 hrs.)**	**Credits:01**

Module 1: Basic Mathematical and Statistical Concepts

Descriptive Statistics Calculation, Covariance and Correlation, Probability Distributions, Multivariate Normal Distribution, Bayes Theorem and Probability

Module 2: Introduction to ML

Environment Setup with Anaconda, Linear Regression Implementation, Multiple Linear Regression, Logistic Regression, Gradient Descent Visualization

Module 3: Classification

Naive Bayes Classifier,K-Nearest Neighbors (K-NN), Perceptron Algorithm.

Module 4: Unsupervised Learning

K-Means Clustering, Hierarchical Clustering. Density-Based Clustering (DBSCAN), Principal Component Analysis (PCA), Boosting Algorithm, Model Validation

Module 5: Keras Tensor Flow

Tensorflow, Keras, Neural Networks, Linear Regression using Tensorflow, MNIST, Character, Recognition, Image classification using CNN.

Course Outcomes:

1. Explain fundamentals of Machine learning using function implementation.
2. Apply threshold learning using feed forward neural networks.
3. Explain descion trees for uni variance Statistical Learning.
4. Apply Inductive Logic Programming.
5. Understand unsupervised machine learning techniques and notion of distances.

List of Text / Reference Books:

1. Introduction to Machine learning, Nils J.Nilsson
2. Machine learning for dummies, IBM Limited ed, by Judith Hurwitz and Daniel Kirsch
3. Introduction to Machine Learning with Python A guide for data scientists,
   Andreas, C. Muller & Sarah Guido, O'Reilly

List of Experiments:

1. Implement and demonstrate Calculate mean, median, mode, variance, and standard deviation for a given dataset.data samples, Use libraries like SciPy to generate distributions and Matplotlib to visualize them.

2. Use libraries like SciPy to generate distributions and Matplotlib to visualize them.,Use Scikit-learn to fit a linear model and visualize the regression line on a scatter plot, Use a multivariate dataset to predict outcomes and assess the performance using metrics like R-squared and adjusted R-squared,

3. Implement Naive Bayes classifier, Use a dataset to classify and evaluate the performance using metrics like accuracy, precision, and recall.,algorithm. Use an appropriate data set for building the decision tree and apply thisknowledge to classify a new sample.

4. Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.

5. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.

6. Implement PCA for dimensionality reduction. Use Scikit-learn to reduce the dimensions of a dataset and visualize the explained variance.

7. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.

8. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library lasses/API.

9. Implement the Perceptron algorithm for a linearly separable dataset.,Train a Perceptron model using Scikit-learn and visualize the decision boundary

10. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library lasses/API in the program.

11. Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.

12. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs

SBC-DS03(P)	Java Programming	0L: 0T: 2P (2 hrs.)	Credits:01

Course Objective:

The objective is to impart programming skills used in this object oriented language java. The course explores all the basic concepts of core java programming.

Course Contents:

Module 1:

Java Basics: Concepts of OOP, Features of Java, How Java is different from C++, Environmental,setup, Basic syntax, Objects and classes, Basic Data Types, Variable Types, Modifier Types, Basicoperators, Loop Control, Decision Making, Strings and Arrays, Methods, I/O.

Module 2:

Java Object Oriented: Inheritance, Overriding, Polymorphism, Abstraction, Encapsulation, Interfaces, Packages, Exploring java.util package.

Module 3:

Exception Handling and Threading: Exception Hierarchy, Exception Methods, Catching Exceptions, Multiple catch Clauses, Uncaught Exceptions Java‟s Built-in Exception. Creating,Implementing and Extending thread, thread priorities, synchronization suspending, resuming and stopping Threads, Multi-threading.

Module 4:

Event Handling and AWT: Event handling Mechanism, Event Model, Event Classes, Sources of Events, Event Listener Interfaces AWT: Working with Windows, AWT Controls, Layout Managers.

Module 5:

Java Advanced: Applet Class, Architecture, Skeleton, Display Methods. Swings: Japplet, Icons,labels, Text Fields, Buttons, Combo Boxes. Socket Programming: Socket methods, Server Socket methods, Socket Client and Socket Server examples.

Course Outcome:

1. Understand basic JAVA concepts and basics of Java programming.
2. Understand JAVA Object oriented concepts
3. Apply Exception Handling and Threading in Java programming.
4. Apply Event Handling and AWT in Java programming.
5. Understand Java Advanced concepts

List of Text / Reference Books:

1. “JAVA The Complete Reference” by Patrick Naughton & Herbert Schild, TMH
2. “Introduction to JAVA Programming a primar”, Balaguruswamy.
3. “Introduction to JAVA Programming” Daniel/Young PH
4. “Java Script”, Jeff Frentzen and Sobotka, Tata McGraw Hill,1999

List of Experiments:

1. WAP to find the average and sum of the N numbers Using Command line argument.
2. WAP to Demonstrate Type Casting.
3. WAP to find the number of arguments provide at runtime.
4. WAP to Test the Prime number.
5. WAP to calculate the Simple Interest and Input by the user.
6. WAP to create a Simple class to find out the Area and perimeter of rectangle and box using super and this keyword.
7. WAP to design a class account using the inheritance and static that show all function of Bank (withdrawal, deposit).
8. WAP to find the factorial of a given number using Recursion.
9. WAP to design a class using abstract Methods and Classes.
10. WAP to design a String class that perform String Method (Equal,Reverse the String, change case).
11. WAP to handle the Exception using try and multiple catch blocks.
12. WAP that Implement the Nested try Statements.
13. WAP to create a package that accesses the member of external class as well as same package.
14. WAP that import the user define package and access the Member variable of classes that Contained by Package.
15. WAP that show the partial implementation of Interface.
16. WAP to Handle the user defined Exception using throw keyword.
17. WAP to demonstrate System clock.

PROJ-DS01	**Mini Project**	**0L: 0T: 4P (4hrs.)**	**Credits:02**

Prerequisite: Knowledge of programming languages, databases, software engineering, problem-solving, teamwork, communication, project planning, and relevant tools and technologies.

Course Objective:

To provide computer science and engineering students with hands-on experience in project development, enhancing their technical, problem solving, teamwork, and project management skills while applying theoretical knowledge to real-world challenges.

Module 1:

Introduction to mini projects, emphasizing real-world applications and problem-solving.Students brainstorm and generate project ideas, conduct feasibility studies, and write detailed proposals covering objectives, scope, literature review, methodology, and expected outcomes. The module concludes with the proposal approval process, involving presentation to and feedback from faculty.

Module 2:

Focusesonproject planning, defining tasks, setting milestones, and creating timelines. Students manage resources, estimate budgets, and form teams, assigning roles and responsibilities. Detailed system design is covered, including high-level architecture, detailed design (UML diagrams, flowcharts), and database design (ER diagrams, normalization, and indexing).

Module 3:

Students set up development environments, adhere to coding standards, and develop project modules, integrating front-end and back-end components. User interface design, including prototyping and user experience, is emphasized. The module also covers comprehensive testing phases: unit testing, integration testing, and system testing.

Module 4:

Teaches technical documentation, including user manuals and comprehensive project reports. Students submit regular progress reports and prepare a final project report. Quality assurance practices are covered, focusing on compliance with standards, peer reviews, and effective bug tracking and management techniques.

Module 5:

Students prepare and deliver project presentations, highlighting key aspects and results. They conduct live project demonstrations and engage in Q&A sessions. The module includes collecting feedback from evaluators, submitting the final report, and reflecting on the learning experience, discussing potential improvements and future project scope.

Course Outcomes:

1. Devise project ideas,conduct feasibility studies,write,and present proposals, incorporating feedback.
2. Develop project plans,set milestones,estimate budgets,form teams,and created etailed system designs.
3. Develop modules,follow coding standards,and perform comprehensive testing.
4. Prepare documentation,submit progress reports,ensur equality,and manage bug tracking effectively.
5. Ability to demonstration of project,gather feedback from evaluators, submit project reports with software engineering standards.

List of Text/Reference Books:

1. Project Management: A Systems Approach to Planning,Scheduling,and Controlling, 13th Edition, Harold Kerzner, ISBN: 978-1-119-80537-3
2. "Software Engineering:APractitioner's Approach"byRogerS.Pressman.
3. "Software Engineering"byIanSommerville.
4. “Software Quality Assurance: Principles and Practice"byNinaS.Godbole.