BAGALKOT UNIVERSITY, JAMKHANDI

PROGRAM/COURSE STRUCTURE AND SYLLABUS

As per the Choice Based Credit System (CBCS)
designed in accordance with
Learning Outcomes-Based Curriculum Framework (LOCF)
Of State Education Policy (SEP) 2025
for

Bachelor of Science
(Computer Science)

(With Three Majors General Degree)

w.e.f.

Academic Year 2025-26 and onwards

PREAMBLE

Computer Science (CS) has been evolving as an important branch of science and technology in last two decade and it has carved out a space for itselflike engineering. Computer Science spans theory and more application and it requires thinking both in abstract terms and in concrete terms.

The ever -evolving discipline of computer science has strong connections to other disciplines. Many problems in science, engineering, health care, business, and other areas can be solved effectively with computers, but finding a solution requires both computer science expertise and knowledge of the particular domain. Computer science has a wide range of specialties. These include Computer Architecture, Software Systems, Graphics, Artificial Intelligence, Mathematical and Statistical Analysis, Data Science, Computational Science, and Software Engineering.

Universities and other HEIs introduced programmes of computerscience. Information Technology is growing rapidly. Increasing applications of computers in almost all areas of human endeavour has led to vibrant industries with concurrent rapid change in technology. Unlike other basic disciplines, developing core competency in this discipline that can be reasonably stable becomes a challenge. In India, it was initially introduced at the Master (postgraduate) level as MCA and M.Tech. Later on, engineering programmes such as B.Tech and B.E in Computer Science & Engineering and in Information Technology were introduced in various engineering College/Institutions to cater to the growing demand for trained engineering manpower in IT industries. Parallelly , BCA, BSc and MSc programmes with specialisation in Computer Science were introduced to train manpower in this highly demanding area.

BSc and BSc (Hons) are aimed at undergraduate level training facilitating multiple career paths. Students so graduated, can take up postgraduate programmes in CS(M.Sc) or MCA leading to research as well as R&D, can be employable at IT industries, or can pursue a teaching profession or can adopt a business management career. BScand BSc (Hons) aimsat laying a strong foundation of computer scienceat

an early stage of the career. There are several employment opportunities and after successful completion of BSc, graduating students can fetch employment directly in companies as programmer, Web Developer, Software Engineer, Network Administrator, Data Scientist, or AI/ML personnel.

The Program outcomes in BSc are aimed at allowing flexibility and innovation in design and development of course content, in method of imparting training, in teaching learning process and in assessment procedures of the learning outcomes. The emphasis in BSc courses, in outcome-basedcurriculum framework, help students learn solving problems, accomplishing IT tasks, and expressing creativity, both individually and collaboratively. The proposed framework will help Students learn programming techniques and the syntax of one or more programming languages.

All students must, therefore, have access to a computer with a modern programming language installed. The computer science framework does not prescribe a specific language. The teacher and students will decide which modern programming languages students will learn. More importantly, students will learn to adapt to changes in programming languages and learn new languages as they are developed. The present Curriculum Framework for BSc degrees is intended to facilitate the students to achieve the following.

- To develop an understanding and knowledge of the basic theory of Computer Science and Information Technology with goodfoundation on theory,systems and applications such as algorithms, data structures, data handling, data communication and computation
- To develop the ability to use this knowledge to analyse new situations inthe application domain
- To acquire necessary and state-of-the-art skills to take up industry challenges. The objectives and outcomes are carefully designed to suit to the above-mentioned purpose.
- The ability to synthesize the acquired knowledge, understanding and experience for a better and improved comprehension of the real-life problems

- To learn skills and tools like mathematics, statistics and electronics to find the solution, interpret the results and make predictions for the future developments
- To formulate, to model, to design solutions, procedure and to use software tools to solve real world problems and evaluate

PROGRAMOUTCOMES:

1. **Discipline knowledge:** Acquiring knowledge on basics of ComputerScienceand ability to apply to design principles in the development of solutions for problems of varying complexity
2. **Problem Solving**: Improved reasoning with strong mathematical ability to Identify, formulate and analyze problems related to computer science and exhibiting a sound knowledge on data structures and algorithms.
3. **Programming a computer**: Exhibiting strong skills required toprogram a computer for various issues and problems of day-to-day applications with thorough knowledge on programming languages of various levels.
4. **Application Systems Knowledge:** Possessing a sound knowledge on computer application software and ability to design and develop app for applicative problems.
5. **Communication:** Must have a reason ably good communication knowledge both in oral and writing.
6. **Ethics on Profession, Environment and Society:** Exhibiting professional ethics to maintain the integrality in a working environment and also have concern on societal impacts due to computer-based solutions for problems.
7. **Lifelong Learning:** Should become an independent learner. So, learn to learn ability.
8. **Motivation to take up Higher Studies:** Inspiration to continue educations towards advanced studies on Computer Science.

By the end of the program the students will be able to:

The Bachelor of Computer Science (BSc (Hons)) program enables students to attain following additional attributes besides the afore-mentioned attributes, by the time of graduation:

1. Apply standard Software Engineering practices and strategies in real - time software project development
2. Acquaint with the contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems
3. The ability to apply the knowledge and understanding noted above to the analysis of a given information handling problem.
4. The ability to work independently on a substantial software project and as an effective team member.

Board of Studies (UG) Committee

(Computer Science)

Bachelor of Science(Computer Science) Programme

2025-2026

1	Dr.DAYANAND G SAVAKAR PROFESSOR,DEPARTMENT OF COMPUTER SCIENCE,BAGALKOT UNIVERSITY,JAMAKHANDI	CHAIRMAN
2	Dr.AZIZ MAKANDAR PROFESSOR,DEPARTMENT OF COMPUTER SCIENCE,KARNATAKA STATE AKKAMAHADEVI WOMEN UNIVERSITY,VIJAYAPUR	MEMBER(EXTERNAL)
3	Dr.RAMESH B MAGURI ASSISTANT PROFESSOR,DEPARTMENT OF COMPUTER SCIENCE,JANATA SHIKSHNA SANGHA SHREE TAMMANNAPPA CHIKODI ARTS AND COMMERCE COLLEGE,BANHATTI	MEMBER
4	Shri.PRAVEENKUMAR S AKKIMARADI ASSISTANT PROFESSOR,DEPARTMENT OF COMPUTER SCIENCE,BASAVESHWAR SCIENCE COLLEGE,BAGALKOT	MEMBER

Dr.DAYANAND G SAVAKAR

CHAIRMAN BOS(UG)

PROFESSOR,DEPARTMENT OF COMPUTER SCIENCE,

BAGALKOT UNIVERSITY,JAMAKHANDI

SEMESTER-III										
Category	**Course code**	**Title of the Paper**	**Marks**			**Teaching hours/week**			**Credit**	**Duration Of Exam (Hrs)**
			IA	**SEE**	**Total**	**L**	**T**	**P**		
L5		Language-I	20	80	100	4	-	-	3	3
L6		Language-II	20	80	100	4	-	-	3	3
DSC7	2A3COMM03T	Object oriented programming in JAVA	20	80	100	4	-	-	3	3
	2A3COMM03L	JAVA Lab	10	40	50	-	-	4	2	3
DSC8	Theory	Theory Course Title	20	80	100	4	-	-	3	3
	Lab	Lab Course Title	10	40	50	-	-	4	2	3
DSC9	Theory	Theory Course Title	20	80	100	4	-	-	3	3
	Lab	Lab Course Title	10	40	50	-	-	4	2	3
CEC1	2A3COME01T	Multimedia and Animation	20	80	100	3	-	-	3	3
	2A3COME02T	Open Source Tools								
	2A3COME03T	Digital Marketing								
Total Marks					**750**	**Semester Credits**			**24**	

SEMESTER-IV										
Category	**Course code**	**Title of the Paper**	**Marks**			**Teaching hours/week**			**Credit**	**Duration Of Exam (Hrs)**
			IA	**SEE**	**Total**	**L**	**T**	**P**		
L7		Language-I	20	80	100	3	-	-	3	3
L8		Language-II	20	80	100	3	-	-	3	3
DSC10	Theory	Database Management Systems	20	80	100	3	-	-	3	3
	Lab	DBMS Lab	10	40	50	-	-	4	2	3
DSC11	Theory	Theory Course Title	20	80	100	3	-	-	3	3
	Lab	Lab Course Title	10	40	50	-	-	4	2	3
DSC12	Theory	Theory Course Title	20	80	100	3	-	-	3	3
	Lab	Lab Course Title	10	40	50	-	-	4	2	3
CEC2	2A3COME04T 2A3COME05T 2A3COME06T	Machine Learning Internet of Things Unix Operating System	20	80	100	3	-	-	3	3
COM2		Skill/Practical based Learning	10	40	50	1	-	2	2	2
Total Marks					**800**	**Semester Credits**			**26**	

BSc (CS)
III SEMESTER

Year	I	Course Code: 2A3COMM03T	Credits	03
Sem.	2	Course Title: Object oriented programming in JAVA	Hours	52

Course Pre-requisites, if any	NA	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration: 03 Hrs.

Course Outcomes

After completing this course satisfactorily, a student will be able to:

1. Understand the features of Java and the architecture of JVM
2. Write, compile, and execute Java programs that may include basic data types and control flow constructs and how type casting is done.
3. Identify classes, objects, members of a class and relationships among them needed for a specific problem and demonstrate the concept of polymorphism and inheritance.
4. The students will be able to demonstrate programs based on interfaces and threads and explain the benefits of JAVA"s Exception handling mechanism compared to other Programming Languages
5. Write, compile, execute Java program that include GUIs and event driven programming and also programs based on files.

Unit No.	Course Content	Hours
Unit-I	Fundamentals of Object Oriented Programming (OOP) - Object, classes, abstraction, encapsulation, inheritance, polymorphism. Difference between Procedural and Object oriented programming. History of Java, features of Java, Identifiers and Keywords, Constants, Variables, Data types, Input and output statements, Math class, Decision making statements, Looping statements, Arrays and its methods,	13
Unit-II	Defining a class with instance variables and methods, object creation, accessing class members, Constructors, Constructor overloading, Method overloading, static members, this and super keywords, Inheritance and its types, Overriding methods, Dynamic method Dispatch, final variables and methods, final classes, finalizer(), Abstract methods and classes, Visibility controls.	13

Unit Ⅲ	Strings, String methods, Interfaces: Defining interfaces, Extending interfaces, Implementing interfaces, Accessing interface variables. Importing classes, user defined packages, Java Input Output: Java IO package, File, Class Byte/Character Stream, Buffered reader / writer, File reader / writer, Print writer File Sequential / Random Serialization and de serialization. Exception handling: Types of Exceptions, try, catch, finally, throw, throws keywords, creating your own exception, nested try blocks, multiple catch statements, user defined exceptions.	14
Unit Ⅳ	Event and GUI programming: Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, ScrollBars, Sliders, Windows, Menus.	12

Recommended Learning Resources

Reference Books:

1.Herbert Schildt, The Java
2.The Complete Reference, Fourth edition, TMH,
3. Balaguruswamy, Programming with JAVA, A primer, TATA McGraw-Hill Company.
4. https://docs.oracle.com/javase/tutorial/ Additional Reading:
5. Peter Van der Liden, Just Java, Prentice Hall
6. H. M. Deitel, P. J. Deitel, Java: how to program, 5th edition, Prentice Hall of India.

Year	I	Course Code: 2A3COMM03L	Credits	02
Sem.	II	Course Title: JAVA Lab	Hours	50

Course Pre-requisites ,if any:	Knowledge of Programming	
Formative Assessment Marks:10	Summative Assessment Marks : 40	Duration of ESA : 03hrs.

Part A:

1. Program to demonstrate class and object.
2. Program to find the biggest of two integer numbers entered by the user.
3. Program to list the factorial of the numbers 1 to 10 .To calculate the factorial value, use while loop.(Hint: Fact of 4=4*3*2*1)
4. Program to find the area and circumference of the circle by accepting the radius from the user.
5. Program to demonstrate the use of method over loading.
6. Program to demonstrate the use of static members.
7. Program to demonstrate the use of method overriding.
8. Program to demonstrate the concept of dynamic method dispatch.
9. Program to demonstrate the use of abstract methods and classes.
10. Create a package "BSC" in your current working directory. a. Create a class student in the BSC package with the following attributes: Name, age, gender. Include appropriate constructor and a method for displaying the details. b. Import above package and access the member variables and function contained in a package.

PART B:

1. Program to Illustrate creation of thread by a) Extending Thread class. b)Implementing Runnable Interfaces
2. Program to demonstrate the concept of inheritance.
3. Program to demonstrate the use of implementing interfaces.
4. Program to demonstrate the use of any 4 string methods.
5. Program to demonstrate exception handling with try, catch and finally.
6. Program which creates and displays a message on the window
7. Program which creates a frame with two buttons father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother button similar details of mother also appear.
8. Create a frame which displays your personal details with respect to a button click
9. Program to create a window with Text Fields and Buttons. The "ADD" button adds the two integers and displays the result. The "CLEAR" button shall clear all the text fields.
10. Demonstrate the various mouse handling events using suitable example.

Year	II	Course Code: 2A3COME01T	Credits	03
Sem.	III	Course Title: Multimedia & Animation	Hours	42
Course Pre-requisites, if any		NA		
Formative Assessment Marks: 20		Summative Assessment Marks:80	Duration of ESA: 03 hours.	
Course Outcomes		1. Write a well-designed, interactive Web site with respect to current standards and practices. 2. Demonstrate in-depth knowledge of an industry-standard multimedia development tool and its associated scripting language. 3. Determine the appropriate use of interactive versus standalone Web applications.		

Unit No.	Course Content	Hours
Unit I	**Web Design:** Origins and evolution of HTML, Basic syntax, Basic text markup, Images, Lists, Tables, Forms, Frame, Overview and features of HTML5. CSS: Introduction, Levels of style sheets, Style specification formats, Selector forms, Property value forms, Font properties, List properties, Color, Alignment of text, The and tags; Overview and features of CSS3. JavaScript: Object orientation and JavaScript; General syntactic characteristics; Primitives, operations, and expressions; Scree output and keyboard input.	10 hours
Unit II	**Animation:** What is an Animation? The Start and End States, Interpolation Animations in HTML. All About CSS Animations, Creating a Simple Animation, Detailed Look at the CSS Animation Property, Keyframes Declaring Multiple Animations, Wrap-up. All About CSS Transitions Adding a Transition, Looking at Transitions in Detail, The Longhand Properties, Longhand Properties vs. Shorthand Properties, Working with Multiple Transitions.	10 hours
Unit III	**Bootstrap 5:** Introduction to Bootstrap 5 (What is Bootstrap, Advantages, CDN vs Local Setup, Folder Structure), Bootstrap Grid System (Containers, Rows, Columns, Breakpoints, Responsive Layouts), Typography and Utilities (Headings, Display classes, Text utilities, Spacing, Colors, Borders, Shadows), Bootstrap Components (Buttons, Alerts, Badges, Breadcrumbs, Cards, Dropdowns, Modals, Navs and Tabs, Navbar, Pagination, Progress bars, Spinners, Toasts), Forms and Inputs (Form controls, Form layout, Form validation, Input	10 hours

	groups, Floating labels), Images and Media (Responsive images, Figures, Embeds, Ratios), Tables and List Groups (Table classes, Striped, Hover, Bordered, Responsive tables, List groups with content), Utilities and Helpers (Flex utilities, Position, Display, Sizing, Visibility, Overflow, Z-index).	
Unit IV	**HTML5 – CANVAS:** The Rendering Context, Browser Support, HTML5 Canvas Examples, Canvas - Drawing Rectangles, Canvas - Drawing Paths Canvas - Drawing Lines, Canvas - Drawing Bezier Curves, Canvas – Drawing Quadratic Curves, Canvas - Using Images, Canvas - Create Gradients, **HTML5 - Styles and Colors, Canvas** - Text and Fonts, Canvas - Pattern and Shadow, Canvas - Save and Restore States, Canvas - Translation, Canvas - Rotation, Canvas - Scaling, Canvas - Transforms, HTML5 Canvas - Composition, Canvas – Animations	**12 hours**
Reference Resources for Learning		
References Books: **1.** The Complete Reference HTML and CSS, 5th Edition, Thomas A Powell, 2017. **2.** Animation in HTML, CSS, and JavaScript, Kirupa Chinnathambi, Createspace Independent Pub, 2013. 1st Edition **3.** https://www.w3.org/Style/CSS/current-work#CSS3 **4.** http://bedford-computing.co.uk/learning/cascading-style-sheets-css/		

Year	II	Course Code: 2A3COME02T	Credits	03
Sem.	III	Course Title: Open Source Tools	Hours	42
Course Pre-requisites, if any		NA		
Formative Assessment Marks: 20		Summative Assessment Marks:80	Duration of ESA: 03 hours.	
Course Outcomes		1. Write a well-designed, interactive Web site with respect to current standards and practices. 2. Demonstrate in-depth knowledge of an industry-standard multimedia development tool and its associated scripting language. 3. Determine the appropriate use of interactive versus standalone Web applications.		

Unit No.	Course Content	Hours
Unit I	**Open Source Software:** Introduction to Open sources, Need of Open Sources, Open Source –Principles, Standard Requirements, Advantages of Open Sources –Free Software – FOSS. Licenses – GPL, LGPL, Copyrights, Patents, Contracts & Licenses and Related Issues Application of Open Sources. Open Source Operating Systems : FEDORA, UBUNTU	10 hours
Unit II	**Programming Tools & Techniques**. Usage of design Tools like Argo UML,Basics of Git and GitHub (version control concepts, git init, add, commit, push, pull, clone, simple branching); Introduction to Linux Command Line (basic shell commands for developers); Open Source IDE - Visual Studio Code (installation, extensions, basic usage); Package Managers in Linux- Introduction to apt &dpkg (for ubuntu os) and rpm for fedora; Code Formatting Tools	10 hours
Unit III	**Introduction to Docker:** Understanding Docker and containerization, benefits of containers over virtual machines; Overview of Docker images and containers, their use in development and production; Basic Docker commands (docker run, docker ps, docker stop, docker rm); Introduction to Docker files for creating custom images; Using Docker Hub for accessing public images; Overview of Docker Compose for managing multi-container applications; Introduction to Docker networking and volumes for data persistence.	10 hours
Unit IV	Introduction to Open Source Movement; Case studies of Apache HTTP Server, Postman Tool: Introduction to Postman;	12 hours

	Setting up Postman and creating workspaces; Sending GET, POST, PUT, DELETE requests; Using Postman Collections to organize requests; BSD Unix, Mozilla Firefox, Wikipedia (MediaWiki), Joomla CMS, GNU Compiler Collection (GCC), and Libre-Office; Comparative analysis of success factors, community governance, licensing, and future trends in open source.	

Reference Resources for Learning

References Books:

1. "Open Source Licensing: Software Freedom and Intellectual Property Law" by Lawrence Rosen
2. "Producing Open Source Software" by Karl Fogel
3. "Pro Git" by Scott Chacon and Ben Straub
4. "Linux Command Line and Shell Scripting Bible" by Richard Blum and Christine Bresnahan
5. "Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers" by Bruce Johnson
6. "Docker: Up & Running" by Karl Matthias and Sean P. Kane
7. "The Docker Book: Containerization is the New Virtualization" by James Turnbull
8. "The Art of Community" by Jono Bacon
9. "Postman Essentials" by Arvind M. and Krishna Sankar

Year	II	Course Code: 2A3COME03T	Credits	03
Sem.	III	Course Title: Digital Marketing	Hours	42
Course Pre-requisites, if any		NA		
Formative Assessment Marks: 20		Summative Assessment Marks:80	Duration of ESA: 03 hours.	
Course Outcomes		1. Define and implement digital marketing strategies using key channels like SEO, SEM, social media, content, and email marketing. 2. Apply SEO and SEM techniques to optimize websites and manage paid search campaigns. 3. Design and execute social media and content marketing campaigns to enhance engagement and conversions. 4. Create effective email marketing campaigns and utilize affiliate marketing and digital tools for business growth.		

Unit No.	Course Content	Hours
Unit I	**Introduction to Digital Marketing:** Definition, importance, and scope of digital marketing. Differences between digital and traditional marketing. Digital marketing channels: SEO, SEM, social media marketing, content marketing, and email marketing. Creating digital marketing strategies, identifying target audiences, setting KPIs. Website fundamentals and the importance of online presence.	10 hours
Unit II	**SEO and SEM (Search Engine Optimization and Search Engine Marketing)** SEO principles: keyword research, on-page SEO (title tags, meta descriptions), off-page SEO (backlinking). Introduction to SEM: paid search advertising, Google Ads, campaign targeting. Analyzing performance using Google Analytics and Google Search Console. Improving website rankings and driving traffic.	10 hours
Unit III	**Social Media Marketing and Content Marketing:** Creating social media strategies for Facebook, Instagram, LinkedIn, Twitter. Planning posts, audience engagement, and social media management tools (Buffer, Hootsuite). Content marketing: blogging, video marketing, podcasting. Creating content calendars, distribution, and optimization. Measuring success through engagement, reach, and conversions.	10 hours
Unit IV	**Email Marketing, Affiliate Marketing, and Tools** Email marketing basics: creating campaigns, managing mailing lists, designing newsletters. Key performance metrics: open	12 hours

	rates, click-through rates. Affiliate marketing basics. Digital marketing tools: HubSpot, MailChimp, Canva, Google Analytics. Career opportunities in digital marketing and professional networking strategies.	

Reference Resources for Learning

References Books:

1. "Digital Marketing: A Practical Approach" by R. K. Suri and Rajeev Kumar
2. "Digital Marketing for Beginners" by Shubham Agarwal
3. "The Art of Digital Marketing" by Pradeep Gohil
4. "Internet Marketing: A Practical Approach" by S. R. S. Sharma and Sumati Reddy

BSc (CS) IV SEMESTER

Year	II	Course Code: 2A4COMM04T Course Title: Database Management System	Credits	03
Sem.	IV		Hours	52
Course Pre-requisites, if any		NA		
Formative Assessment Marks: 20		Summative Assessment Marks:80	Duration of ESA:03 hours.	
Course Out comes		1. Explain the various database concepts and the need for database systems. 2. Identify and define database objects, enforce integrity constraints on a database using DBMS. 3. Demonstrate a Data model and Schemas in RDBMS. 4. Identify entities and relationships and draw ER diagram for a given real-world problem. 5. Convert an ER diagram to a database schema and deduce it to the desired normal form. 6. Formulate queries in Relational Algebra, Structured Query Language (SQL) for database manipulation. 7. Explain the transaction processing and concurrency control techniques & NOSQL MongoDB Database System.		

Unit No.	Course Content	Hours
Unit I	**Database Architecture:** Introduction to Database system applications. Characteristics and Purpose of database approach. People associated with Database system. Data models. Database schema. Database architecture. Data independence. Database languages, interfaces, and classification of DBMS.	13
Unit II	**E-R Model:** Entity-Relationship modeling: E – R Model Concepts: Entity, Entity types, Entity sets, Attributes, Types of attributes, key attribute, and domain of an attribute. Relationships between the entities. Relationship types, roles and structural constraints, degree and cardinality ratio of a relationship. Weak entity types, E -R diagram.	13
Unit III	**Relational Data Model:** Relational model concepts. Characteristics of relations. Relational model constraints: Domain constrains, key constraints, primary & foreign key constraints, integrity constraints and null values. Relational Algebra: Basic Relational Algebra operations. Set theoretical operations on relations. JOIN operations Aggregate Functions and Grouping. Nested Sub Queries-Views. Introduction to PL/SQL & programming of above operations in PL/SQL.	13
Unit IV	**Data Normalization:** Anomalies in relational database design. Decomposition. Functional dependencies. Normalization. Normal Forms: 1NF 2NF 3NF & BCNF. Query Processing Transaction Management: Introduction Transaction Processing. Transactions: read & write operations. Need of concurrency control: The lost update, & Dirty read problem. Types of failures. Transaction	13

	states. Desirable properties (ACID properties) of Transactions. **NOSQL &MongoDB:** Introduction, Relational vs. NoSQL, Document Model), Mongosh Shell, Databases, Collections, Documents (BSON), _id, Insert (insertOne, insertMany), Read (find, Query, Project), Update (updateOne, $set), Delete (deleteOne).	

References:

1. Fundamentals of Database Systems, Ramez Elamassri, Shankant B. Navathe, 7th Edition, Pearson, 2015
2. An Introduction to Database Systems, Bipin Desai, Galgotia Publications, 2010.
3. Introduction to Database System, C J Date, Pearson, 1999.
4. Database Systems Concepts, Abraham Silberschatz, Henry Korth, S.Sudarshan, 6th Edition, McGraw Hill, 2010.
5. Database Management Systems, Raghu Rama Krishnan and Johannes Gehrke, 3rd Edition, McGraw Hill, 2002

Year	II	Course Code: 2A4COMM04L Course Title: Database Management System LAB	Credits	02
Sem.	IV		Hours	52
Course Pre-requisites, if any		NA		
Formative Assessment Marks: 10		Summative Assessment Marks:40	Duration of ESA: 03 hours.	

PART-A

I. Execute DDL (Data Definition Language) Commands

Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

Tasks:

1. Create table with appropriate constraints (PRIMARY KEY)
2. Alter table to add a new column (email)
3. Drop email column
4. Rename a table student to student1
5. Drop a table student1

II. Execute DML (Data Manipulation Language) Commands

Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

Tasks:

1. Insert multiple records using INSERT
2. Update percentage for specific students
3. Delete students with percentage < 40

III. Execute DCL and TCL Commands

Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

Context: Admin role simulation

Tasks:

1. Grant SELECT, INSERT, UPDATE privileges to a user
2. Revoke privileges
3. Demonstrate use of SAVEPOINT, ROLLBACK, and COMMIT

IV. Execute Single-Line Queries and Group Functions

Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

Tasks:

1. Select students by specific course or percentage
2. Use functions like COUNT(), SUM(), MAX(), MIN(), AVG() on percentage column
3. Use GROUP BY and HAVING to calculate average % by course

V. Create Views for a Particular Table

Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

Tasks:

1. Create a view of students from BCA course
2. Create a view with selected columns (rollno, name, percentage)
3. Insert into and update data through view (check if updatable)
4. Drop the view

PART-B

VI. Implement Join Operations in SQL

Tables:

STUDENT(student_id, name, department)
COURSE(course_id, course_name, department)
ENROLLMENT(enroll_id, student_id, course_id, enroll_date)

Tasks:

1. Inner Join: List student names with their course names
2. Left Join: List all students with course details (if any)
3. Join + GROUP BY: Number of students enrolled in each course

VII. Implement Nested Queries

Tables:

STUDENT(rollno, name, department)
COURSE(course_id, course_name, department)
ENROLLMENT(enroll_id, student_id, course_id, enroll_date)

Tasks:

1. Find the student(s) with the highest percentage
2. Use subqueries to fetch students enrolled in a specific department's course

VIII. Implement lock Table operation with Data Insertion and commit for

Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

IX. Implement PLSQL Stored procedure add_student to insert row into table

Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

X. Implement following exercise using NOSQL Database MongoDB

Tasks

1. Create a database named myFirstDB.
2. Create a collection named items.
3. Insert the following documents into the items collection:
    i. { name: "Apple", color: "red", price: 1 }
    ii. { name: "Banana", color: "yellow", price: 0.5 }
    iii. { name: "Orange", color: "orange", price: 1.2 }
4. Retrieve all documents from the items collection.
5. Find all items that have a price less than 1.1.
6. Find the item with the name "Banana".
7. Update the price of the "Apple" to 1.1.
8. Delete the item with the name "Orange".

Year	II	Course Code: 2A3COME04T	Credits	03
Sem.	IV	Course Title: Machine Learning	Hours	42
Course Pre-requisites, if any		NA		
Formative Assessment Marks: 20		Summative Assessment Marks:80	Duration of ESA:03hrs.	
Course Outcomes		1. Understand ML concepts and data preprocessing techniques. 2. Implement and evaluate regression and classification models. 3. Apply clustering and dimensionality reduction techniques.		

Unit No.	Course Content	Hours
Unit I	**Introduction to Machine Learning:** Fundamental concepts of Machine Learning (ML), applications, supervised, unsupervised, and reinforcement learning. Real-world ML applications in finance, healthcare, and recommendation systems. Data preprocessing techniques, handling missing values, feature scaling (normalization and standardization), encoding categorical variables. Python ecosystem for ML, including NumPy, Pandas, and Matplotlib. Hands-on exercises in data manipulation, visualization, exploratory data analysis (EDA) using Pandas and Matplotlib	10 hours
Unit II	**Regression Analysis & Evaluation Metrics** Regression models, simple linear regression, multiple linear regression, polynomial regression. Mathematical intuition and implementation using Scikit-Learn. Evaluation metrics including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared. Practical exercises in implementing regression models on real datasets such as predicting house prices or stock prices. Fine tuning hyper parameters and understanding model performance	10 hours
Unit III	**Classification Algorithms & Model Optimization** Classification algorithms, Logistic Regression, k-Nearest Neighbors (k-NN), Decision Trees, Support Vector Machines (SVM). Training and testing classification models, performance evaluation using confusion matrices, precision-recall curves, and F1-scores. Practical	10 hours

	implementation of binary and multiclass classification tasks, such as spam email detection or digit classification using the MNIST dataset.	
Unit IV	**Clustering & Dimensionality Reduction** Unsupervised learning techniques, K-Means Clustering, Hierarchical Clustering. Implementing clustering on datasets to identify patterns and groupings without predefined labels. Principal Component Analysis (PCA), t-SNE for dimensionality reduction while retaining essential features. Hands-on exercises in customer segmentation and visualizing high-dimensional data. Limitations of clustering algorithms and effective cluster interpretation	**12 hours**

Reference Resources for Learning

References Books:

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow – Aurélien Géron, 3rd Edition, O'Reilly Media.
2. Python Machine Learning – Sebastian Raschka, Vahid Mirjalili, 3rd Edition, Packt Publishing.
3. Machine Learning for Absolute Beginners – Oliver Theobald, 2nd Edition, Scatterplot Press.
4. Introduction to Machine Learning with Python – Andreas C. Müller, Sarah Guido, 1st Edition, O'Reilly Media.
5. Applied Machine Learning with Python – Andrea Giussani, 1st Edition, Apress.
6. Machine Learning Yearning – Andrew Ng, Self-Published.

Year	II	Course Code: 2A3COME05T	Credits	03
Sem.	IV	Course Title: Internet of Things	Hours	42
Course Pre-requisites, if any		NA		
Formative Assessment Marks: 20		Summative Assessment Marks:80	Duration of ESA: 03 hours.	
Course Outcomes		1. Understand IoT architecture and hardware components. 2. Program microcontroller devices using Python. 3. Integrate devices with cloud platforms and visualize data. 4. Address IoT security concerns and manage data.		

Unit No.	Course Content	Hours
Unit I	**Introduction to IoT and Hardware Basics** Definition and characteristics of IoT, evolution and applications of IoT in smart homes, health, agriculture, and industries. IoT architecture: perception, network, and application layers. Overview of microcontrollers and microprocessors. Introduction to sensors and actuators. Basic setup of Arduino and Raspberry Pi. Interfacing sensors (temperature, light, motion) with Arduino.	10 hours
Unit II	**Programming IoT Devices using Python** Installing Python on Raspberry Pi, basic Python programming review. GPIO pin configuration. Reading data from sensors using Python. Controlling devices (LED, buzzer, relay) using Python scripts. Simple Python projects like smart light and motion alarm	10 hours
Unit III	**IoT Communication Protocols and Cloud Integration** IoT communication protocols: HTTP, MQTT, Bluetooth, Zigbee, Wi-Fi. Implementing MQTT protocol using Python (paho-mqtt). Connecting devices to open IoT platforms like ThingSpeak or Adafruit IO. Sending sensor data to cloud. Visualizing data using Python and ThingSpeak.	10 hours
Unit IV	**IoT Security and Data Management** Security issues in IoT devices. Authentication, encryption basics. Secure communication protocols (TLS, HTTPS). Introduction to JSON and data serialization. Storing and retrieving IoT data. Managing data using SQLite with Python.	10 hours

Reference Resources for Learning

References Books:

1. "Internet of Things with Python" by Gaston C. Hillar, Packt Publishing.
2. "Mastering IoT with Python" by Colin Dow, Packt Publishing.
3. "Getting Started with Raspberry Pi" by Matt Richardson and Shawn Wallace, Make:
4. "Python Programming for Arduino" by Pratik Desai, Packt Publishing.
5. "Hands-On Internet of Things with MQTT" by Tim Pulver, Packt Publishing.

Year	II	Course Code: 2A3COME06T	Credits	03
Sem	IV	Course Title: UNIX AND SHELL PROGRAMMING	Hours	42
Formative Assessment Marks: 20	Summative Assessment Marks:80			Duration of ESA: 03 hours
Course Outcomes	1. **Understands shell concept in Linux environment and practice basic commands of UNIX/Linux.** **2. Study of advance commands and filters of UNIX/Linux.** **3. Using vi editor for writing shell scripts.**			

Unit	Course Content	Hours
Unit I	**Introduction and Basic Commands**: Brief history, Salient features of UNIX system, The UNIX architecture, Internal and External Commands along with options: cal, bc, date, echo, cat, who, cp, mv, rm, tty, sty, uname, pwd, passwd., man, Script commands Flexibility of CommandUsage. **File System and File Handling Commands:** more, lp, wc, spell, ispell, directory commands: mkdir, cd,rmdir, ls command & its options:(-x,-f,-a,-d,-l), cmp, comm, diff: Compressing files, gzip, gunzip, and unzip commands, changing file permissions: chmod, changing group permissions: chgrp, changing file ownership: chown, umask, locating files: find, path.	10 Hours
Unit II	**Standard I/O, Redirection Pipes & Filters:** Standard I/O, Redirection, pipe & pipeline, filter, tee command, terminal & trash files, pr, head, tail, cut, paste, sort, uniq, tr, grep, **The Process:** Process Basics, ps: Process Status& its options, mechanism of Process Creation, Internal and External Commands, Running jobs in Background, nice: Job Execution with Low Priority, Killing Processes with Signals, Job Control, fg and bg commands at and batch: Execute. Later, cron: Running Jobs Periodically, time: Timing Processes.	10 Hours
Unit III	**The vi Editor**: vi Basic, Input Mode - Entering and Replacing Text, Saving Text and Quitting - The ex-Mode, Navigation, Editing Text, Undoing Last Editing Instructions, (u and U) Repeating the Last Command (.), Searching for a Pattern (I and ?), Substitution Search and Replace (:s) **Shell:** The Shell's Interpretive Cycle, Pattern Matching - The Wild - cards, Escaping and quoting, redirection: The three Standard Files, /dev/null and /dev/tty: Two Special Files, Command Substitution.	10 Hours
Unit IV	**Shell programming**: Shell scripts, shell variables, read, using command line arguments, exit &exit status of command,	12 Hours

	arithmetic operators, relational operators, increment & decrement operators, assignment operators, the logical operators && and ||,if conditional, test and [ ] to evaluate expressions, case conditional, expr: computation & string handling, while looping, for looping, break & continue statements.	

Reference Resources for Learning

Reference Books

1. Sumitabha Das, Unix Concepts and Applications, McGraw Hill publications.
2. Venkateshmurthy, Introduction to UNIX and Shell Programming, Pearson Education.
3. Glass, UNIX for Programmers and Users, 3/e Pearson Education.
4. Kernighan, The UNIX Programming Environment.
5. Kochan, UNIX Shell Programming Pearson.

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Activity–1 from Part A	Write up on the activity/task	07
	Demonstration of the activity/ task	08
Activity-2 from Part B	Write up on the activity/task	07
	Demonstration of the activity/ task	08
Viva based on Lab Activities		10
Total		**40**