GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT (GITAM)

(Deemed to be University)

VISAKHAPATNAM * HYDERABAD * BENGALURU

Accredited by NAAC with A^{++} Grade

GITAM School of Science

CURRICULUM AND SYLLABUS

2 Year Postgraduate Programme

PCSCI01: Master of Computer Applications

w.e.f. 2020-21 admitted batch

Master of Computer Applications (M.C.A)

REGULATIONS

(W.e.f. 2020-2021 admitted batch)

1.ADMISSION

Admission into 2-year M.C.A program of GITAM University is governed by GITAM University admission regulations.

2.ELIGIBILITY CRITERIA

Passed BCA/Bachelor Degree in Computer Science Engineering or Equivalent Degree

OR

Passed B.Sc/B.Com/B.A with Mathematics at 10+2 level or graduation level (with additional bridge Courses as per the norms of the concerned University).

Obtained at least 50% marks (45%marks in case of candidates belonging to reserved category) in the qualifying examination.

Admission into M.C.A (Master Computer Applications) will be based on an All India GITAM Science Admission Test (GSAT) conducted by GITAM University and the rule of reservation, wherever applicable.

3.CHOICE BASED CREDIT SYSTEM

Choice Based Credit System (CBCS) is introduced with effect from the admitted Batch of2015-16 based on UGC guidelines in order to promote:

Student Centered Learning

Cafeteria approach

Inter-disciplinary learning

Learning goals/ objectives and outcomes are specified leading to what a student should be able to do at the end of the program.

4.STRUCTURE OF THE PROGRAM

The Program Consists of

Foundation Courses (compulsory) which give general exposure to a Student in communication and subject related area.

Core Courses (compulsory).

Discipline centric electives which are supportive to the discipline give expanded scope of the subject give their disciplinary exposure nurture the student skills

Open electives are of general nature either related or unrelated to the discipline.

Practical Proficiency Courses, Laboratory and Project work.

Each course is assigned a certain number of credits depending upon the number of

contact hours (lectures/tutorials/practical) per week.

In general, credits are assigned to the courses based on the following contact hours per week per semester.

One credit for each Lecture / Tutorial hour per week.

One credit for two hours of Practical per week.

Eight credits for project.

The curriculum of the Four Semesters M.C.A program is designed to have a total of 88 credits for the award of M.C.A degree.

5.MEDIUM OF INSTRUCTION

The medium of instruction (including examinations and project reports) shall be in English.

6.REGISTRATION

Every student has to register himself / herself for each semester individually at the time specified by the Institute / University.

7.ATTENDANCE REQUIREMENTS

A student whose attendance is less than 75% in all the courses put together in any semester will not be permitted to attend that end - semester examination and he/she will not be allowed to register for subsequent semester of study. He/she has to repeat the semester along with his / her juniors

However, the Vice Chancellor on the recommendation of the Principal / Director of the Institute/School may condone the shortage of attendance to the students whose attendance is between 66% and 74% on genuine grounds and on payment of prescribed fee.

8.EVALUATION

The assessment of the student's performance in a Theory course shall be based on two components: Continuous Evaluation (40 marks) and Semester-end examination (60 marks).

A student has to secure an aggregate of 40% in the course in continuous and semester end examinations the two components put together to be declared to have passed the course, subject to the condition that the candidate must have secured a minimum of 24 marks (i.e. 40%) in the theory component at the semester-end examination.

Practical / Viva voce etc. course are completely assessed under Continuous Evaluation for a maximum of 100 marks and a student has to obtain a minimum of 40% to secure Pass Grade. Details of Assessment Procedure are furnished below in Table 1.

Table 1: Assessment Procedure

S. No.	Component of assessment	Marks allotted	Type of Assessment	Scheme of Examination
1	Theory	40	Continuous evaluation	(i) Three mid semester examinations shall be conducted for 15 marks each. The performance in best two shall be taken into consideration. (ii) 5 marks are allocated for quiz. (iii) 5 marks are allocated for assignments.
		60	Semester-end examination	The semester-end examination shall be for a maximum of 60 marks.
	Total	100		
2	Practical	100	Continuous evaluation	60 marks for performance, regularity, record/ and case study. Weightage for each component shall be announced at the beginning of the semester. 40 marks (30 marks for experiment(s) and 10 marks for practical Viva-voce.) for the test conducted at the end of the Semester conducted by the concerned lab Teacher.
	Total	100		
3	Project work	200	Project evaluation	150 marks for evaluation of the project work dissertation submitted by the candidate. 50 marks are allocated for the project Viva-Voce. The project work evaluation and the Viva-Voce shall be conducted by one external examiner outside the University and the internal examiner appointed by the Head of the Department.

9.RETOTALING &REVALUATION

Re-totaling of the theory answer script of the semester-end examination is permitted on request by the student by paying the prescribed fee within one week after the announcement of theresults.

Revaluation of the theory answer scripts of the semester-end examination is permitted on request by the student by paying the prescribed fee within one week after the announcement of theresult.

10.PROVISION FOR ANSWER BOOK VERIFICATION & CHALLENGE EVALUATION:

If a student is not satisfied with his/her grade after revaluation, the student can apply for, answer book verification on payment of prescribed fee for each course within one week after announcement of revaluationresults.

After verification, if a student is not satisfied with revaluation marks/grade awarded, he/she can apply for challenge valuation within one week after announcement of answer book verification result/ two weeks after the announcementof revaluation results, which will be valued by the two examiners i.e., one Internal and one External examiner in the presence of the student on payment of prescribed fee. Thechallenge valuation fee will be returned, if the student is succeeded in the appeal with achange for a bettergrade.

11.SUPPLEMENTARY EXAMINATIONS & SPECIAL EXAMINATIONS:

The odd semester supplementary examinations will be conducted on daily basis after conducting regular even semester examinations in April/May.

The even semester supplementary examinations will be conducted on daily basis afterconducting regular odd semester examinations during November/December

A student who has completed his/her period of study and still has **"**F**"** grade in final semester courses is eligible to appear for Special Examination normally held during summer vacation.

12.PROMOTION TO THE NEXT YEAR OF STUDY

A student shall be promoted to the next academic year only if he/she completes the academic requirements of 60% of the credits till the previous academic year.

Whenever there is a change in syllabus or curriculum, he/she has to continue the course with new regulations after detention as per the equivalency established by the BoS to continue his/her further studies

13.BETTERMENT OF GRADES

A student who has secured only a pass or second class and desires to improve his/her class can appear for betterment examinations only in **'**n**'** (where **'**n**'** is no.of semesters of the program) theory courses of any semester of his/her choice, conducted in summer vacation along with the Special Examinations.

Betterment of Grades is permitted 'only once', immediately after completion of the program of study.

14.REPEAT CONTINUOUS EVALUATION

A student who has secured 'F' grade in a theory course shall have to reappear at the subsequent examination held in that course. A student who has secured 'F' grade can improve continuous evaluation marks upto a maximum of 50% by attending special instruction classes held during summer.

A student who has secured 'F' grade in a practical course shall have to attend Special Instruction classes held during summer.

A student who has secured 'F' grade in a combined (theory and practical) course shall have to reappear for theory component at the subsequent examination held in that course. A student who has secured 'F' grade can improve continuous evaluation marks upto a maximum of 50% by attending special instruction classes held during summer.

The RCE will be conducted during summer vacation for both odd and even semester students. Student can register a maximum of 4 courses. Biometric attendance of these RCE classes has to be maintained. The maximum marks in RCE be limited to 50% of Continuous Evaluation marks. The RCE marks are considered for the examination held after RCE except for final semester students.

RCE for the students who completed course work can be conducted during the academic semester. The student can register a maximum of 4 courses at a time in slot of4 weeks. Additional 4 courses can be registered in the next slot.

A student is allowed to Special Instruction Classes (RCE) 'only once' per course.

15.GRADING SYSTEM

Based on the student performance during a given semester, a final letter grade will be awarded at the end of the semester in each course. The letter grades and the corresponding grade points are as given in Table 2.

Table 2: Grades & Grade Points

Sl.No.	Grade	Grade Points	Absolute Marks
1	O (outstanding)	10	90 and above
2	A+ (Excellent)	9	80 to 89
3	A (Very Good)	8	70 to 79
4	B+ (Good)	7	60 to 69
5	B (Above Average)	6	50 to 59
6	C (Average)	5	45 to 49
7	P (Pass)	4	40 to 44
8	F (Fail)	0	Less than 40
9	Ab. (Absent)	0	-

A student who earns a minimum of 4 grade points (P grade) in a course is declared to have successfully completed the course, subject to securing an average GPA (average of all GPAs in all the semesters) of 5 at the end of the Program to declare pass in the program.

Candidates who could not secure an average GPA of 5 at the end of the program shall be permitted to reappear for a course(s) of their choice to secure the same.

16.GRADE POINT AVERAGE

A Grade Point Average (GPA) for the semester will be calculated according to the formula:

$$GPA = \frac{\sum [C*G]}{C}$$

Where

C = number of credits for the course,
G = grade points obtained by the student in the course.

To arrive at Cumulative Grade Point Average (CGPA), a similar formula is used considering the student's performance in all the courses taken, in all the semesters up to the particular point of time.

CGPA required for classification of class after the successful completion of the program is shown in Table 3.

Table 3: CGPA required for award of Class

Class	CGPA Required
First Class with Distinction	**≥ 8.0***
First Class	**≥ 6.5**
Second Class	**≥ 5.5**
Pass Class	**≥ 5.0**

In addition to the required CGPA of 8.0 or more the student must have necessarily passed all the courses of every semester in first attempt.

17.ELIGIBILITY FOR AWARD OF THE M.C.A DEGREE

Duration of the program: A student is ordinarily expected to complete M.C.A program in Four semesters of two years. However, a student may complete the program in not more than Four years including study period.

However the above regulation may be relaxed by the Vice Chancellor in individual cases for cogent and sufficient reasons.

A student shall be eligible for award of the M.C.A Degree if he / she fulfills all thefollowing conditions.
Registered and successfully completed all the courses and projects.
Successfully acquired the minimum required credits as specified in the curriculumcorresponding to the branch of his/her study within the stipulated time.
Has no dues to the Institute, hostels, Libraries, NCC / NSS etc,
andNo disciplinary action is pending against him / her.

The degree shall be awarded after approval by the Academic Council.

18.DISCRETIONARY POWER

Not with standing anything contained in the above sections, the Vice Chancellor may reviewall exceptional cases, and give his decision, which will be final and binding.

19. Vision and Mission of the Department of Computer Science

Vision of the Department

To become a leading hub for education and innovation in computer science, empowering students with emerging technologies for global tech leadership through pioneering research and active community engagement.

Mission of the Department

M1: Foster a new generation of skilled computer science professionals through a well-structured curriculum that encourages continuous learning and prepares students for diverse, dynamic careers in emerging technologies.

M2: Conduct robust research in emerging fields of computer science and engage in strategic collaborations with industry and community partners to make significant contributions to society.

M3: Uphold the highest ethical standards, transparency, and accountability while fostering inclusivity and diversity in pushing the boundaries of technological advancement.

20. Program Educational Objectives, Program Outcomes

Program Educational Objectives

PEO1: Graduates will pursue advanced degrees or engage in research in core and emerging areas of computer science, contributing to the advancement of the field.

PEO2: Graduates will become successful entrepreneurs or excel professionally, applying their skills in core and emerging areas of computer science to address societal challenges.

PEO3: Graduates will become competent professionals in industry, academia, and other organizations, continually adapting to evolving technologies in core and emerging areas of computer science.

Program Outcomes (POs)

PO1: Foundation Knowledge
Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.

PO2: Problem Analysis
Identify, review, formulate and analyse problems for primarily focussing on customer requirements using critical thinking frameworks.

PO3: Development of Solutions
Design, develop and investigate problems with as an innovative approach for solutions incorporating ESG/SDG goals.

PO4: Modern Tool Usage
Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.

PO5: Individual and Teamwork
Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use methodologies such as agile.

PO6: Project Management and Finance
Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.

PO7: Ethics
Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware.

PO8:Life-long learning
Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.

Master of Computer Applications
(M.C.A.) Scheme of Instruction

I SEMESTER

SNo	Course Code	Course Title	**Category**	**L**	**T**	**P**	**C**	**Remarks**
1	20SCA 701	Discrete Mathematics	PC	3	1	0	4	
2	20SCA 703	Object Oriented Programming with C++	PC	3	1	0	4	
3	20SCA 705	Web Programming	PC	3	1	0	4	
4	20SCA 707	Operating Systems	PC	3	1	0	4	
5	VDC111	Venture Discovery	SSE	3	0	0	2	
Practical's								
6	20SCA 721	Object Oriented Programming with C++ Lab	PP	0	0	3	2	
7	20SCA 723	Web Programming Lab	PP	0	0	3	2	
8	20SCA 725	Operating Systems Lab	PP	0	0	3	2	
		Total					24	

II SEMESTER

SNo	Course Code	Course Title	Category	L	T	P	C	Remarks
1	20SCA 702	Probability and Statistics	PC	3	1	0	4	
2	20SCA 704	Data Structures using C++	PC	3	1	0	4	
3	20SCA 706	Database Management Systems	PC	3	1	0	4	
4	20SCA 708	Object Oriented Software Engineering	PC	3	1	0	4	
5	20SAE 702	Technical Communication Skills	AEC	3	1	0	3	
Practicals								
6	20SCA 722	Data Structures using C++ Lab	PP	0	0	3	2	
7	20SCA 724	Database Management Systems Lab	PP	0	0	3	2	
		Total					23	

III SEMESTER

SNo	Course Code	Course Title	Category	L	T	P	C	Remarks
1	20SCA 801	Object oriented programming with JAVA	PC	3	1	0	4	
2	20SCA 803	Design and Analysis of Algorithm	PC	3	1	0	4	
3	20SCA 805	Data mining	PC	3	1	0	4	
4	20SCA 841 20SCA 843 20SCA 845	Generic Elective – I (a) Artificial Intelligence (b) Cloud Computing (c) Network Security	GE	3	1	0	4	
5	SSE 801 SSE 803	Skill Enhancement Digital Marketing Management Information System	SEC	3	0	0	3	
Practical's								
6	20SCA 821	Object oriented programming with JAVA Lab	PP	0	0	3	2	
7	20SCA 824	Data Analysis using R Lab	PP	0	0	3	2	
		Total					23	

IV SEMESTER

SNo	Course Code	Course Title	Category	L	T	P	C	Remarks
1	20SCA842 20SCA844 20SCA846	Generic Elective – II (a) Machine Learning (b) Block Chain Technologies (c) Cyber Security	GE	3	1	0	4	
2	20SCA848 20SCA850 20SCA852	Generic Elective – III (a) Big Data Analytics (b)Python Programming (c) Advanced Java Programming	GE	3	1	0	4	
Practical's								
3	20SCA868 20SCA870 20SCA872	(a)Big Data Analytics Lab (b)Python Programming Lab (c)Advanced Java Programming Lab	pp	0	0	3	2	
4	20SCA891	Project	PP	0	0	3	8	
		Total					18	

Total Credits: 24+23++23+18= 88

MCA I SEMESTER

20SCA 701 : DISCRETE MATHEMATICS

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Discrete Mathematics" course provides a strong foundation in the essential mathematical concepts needed for computer science. It focuses on logic, set theory, lattices, Boolean algebra, and graph theory, which are critical for understanding various computational problems and algorithms. The course also introduces students to formal reasoning, proofs, and abstract mathematical structures.

Course Objectives:

- To introduce the principles of mathematical logic, including connectives, normal forms, and inference theory.
- To provide a comprehensive understanding of set theory, relations, and functions.
- To familiarize students with the concepts of lattices and their properties.
- To teach Boolean algebra, Boolean functions, and techniques for minimizing Boolean expressions.
- To introduce graph theory concepts, including different types of graphs, paths, connectivity, and algorithms for traversing graphs.

UNIT - I

Mathematical Logic: Statements and Notation, Connectives, Normal Forms, The Theory of Inference for the Statement Calculus, The Predicate Calculus, Inference Theory of the Predicate Calculus. 10 Hours

UNIT - II

Set Theory: Basic Concepts of Set Theory, Relations and Ordering.

Functions: Definitions and Introduction, Composite of Functions, Inverse Functions, Binary and n-ary Operations, Characteristic Function of a Set. 10 Hours

UNIT - III

Lattices : Definition and Examples, Properties of Lattices, Sub lattices, Direct Product and Homomorphism, Some Special Lattices. 10 Hours

UNIT - IV

Boolean Algebra: Definition and Examples, sub algebra, Direct product and Homomorphism, Boolean Functions, Boolean forms and free Boolean Algebras, Values of Boolean expressions and Boolean functions, Representation of Boolean functions, Minimization of Boolean functions, Karnaugh maps. 10 Hours

UNIT - V

Graph Theory: Graphs, Multi graphs, Directed Graphs, Complete, Regular and Bipartite Graphs, Planar Graphs, Tree Graphs, Labeled and Weighted Graphs, Basic Definitions, Sub-graphs, Isomorphic Graphs, Paths, Connectivity, The Bridges of Konigsberg, Traversable Multi- graphs. Rooted Trees, Sequential Representation of Directed Graphs, Warshall's Algorithm and Shortest Path (Minima) Algorithm only. 10 Hours

Text Books:

1. Discrete Mathematical Structures with applications to computer science by J.P. Tremblay and R. Monohar, Tata McGraw – Hill.
2. Discrete Mathematics by Seymour Lipschutzand Marc Lipson, Schaum's outlines, Tata McGraw Hill.

3. Introductory Methods of Numerical Analysis by S.S Sastry, Prentice – Hall India.

Reference Books:

1. Discrete Mathematics and its Applications by Kenneth H.Rosen, Tata McGraw – Hill.
2. Numerical Methods for Engineers by Steven C. Chopra and Raymond P.Canale, McGraw Hill

Course Outcomes:

At the end of the Course, the student is able to

- CO1: Understand and apply the principles of mathematical logic to infer conclusions and solve problems. *(L2: Understand)*
- CO2: Analyze and use set theory, relations, and functions to model and solve real-world problems. *(L3: Apply)*
- CO3: Understand the properties of lattices and their applications in various mathematical structures. *(L4: Analyze)*
- CO4: Apply Boolean algebra and techniques like Karnaugh maps to simplify Boolean expressions and design logic circuits. *(L5: Evaluate)*
- CO5: Understand and apply graph theory concepts and algorithms to model networks and solve connectivity problems. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

MCA I SEMESTER

20SCA 703: OBJECT ORIENTED PROGRAMMING WITH C++

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Object-Oriented Programming with C++" course provides students with a deep understanding of object-oriented programming principles and their practical implementation using C++. This course covers fundamental concepts such as classes, inheritance, polymorphism, operator overloading, templates, and exception handling, which are crucial for building robust, reusable, and maintainable software systems.

Course Objectives:

- To introduce the fundamental concepts of object-oriented programming and C++ syntax.
- To teach students how to use classes, objects, and constructors to design object-oriented programs.
- To enable students to implement operator overloading, inheritance, and polymorphism in C++.
- To familiarize students with templates for creating generic functions and classes in C++.
- To provide knowledge on exception handling techniques and file I/O operations in C++.

UNIT –I

Overview of C++: The Origins of C++, What Is Object-Oriented Programming? Encapsulation, Polymorphism, Inheritance, Some C++ Fundamentals, A Sample C++ Program, A Closer Look at the I/O Operators, Declaring Local Variables, No Default to int, The bool Data Type, Old- Style vs. Modern C++, The New C++ Headers, Namespaces, Working with an Old Compiler, Introducing C++ Classes , Function Overloading, Operator Overloading, Inheritance, Constructors and Destructors, The C++ Keywords, The General Form of a C++ Program .

Classes and Objects: Classes and Objects , Classes, Structures and Classes Are Related, Unions and Classes Are Related Anonymous Unions, Friend Functions, Friend Classes, Inline Functions, Defining Inline Functions Within a Class, Parameterized Constructors, Constructors with One Parameter: A Special Case, Static Class, Static Data Members, Static Member Functions, When Constructors & Destructors are Executed, Scope Resolution Operator, Nested, Local Classes, Passing Objects to Functions, Returning Objects, Object Assignment. 10 Hours

UNIT –II

Arrays, Pointers, References, and the Dynamic Allocation Operators: Arrays of Objects , Creating Initialized vs. Uninitialized Array, Pointers to, Type Checking, C++Pointer The this Pointer, Pointers to Derived Types, Pointers to Class, Reference Parameters, Passing References to Objects, Returning References Independent References, References to Derived Types, Restrictions to References, A Matter of Style, C++'s Dynamic Allocation, Initializing Allocated Memory, Allocating Arrays, Allocating objects, The nothrow, The Placement Forms of new and delete.

Function Overloading, Copy Constructors, and Default Arguments: Function Overloading, Copy Constructors, and Default Arguments, Function Overloading, Overloading Constructor Functions, Overloading a Constructor to Gain Flexibility, Allowing Both Initialized and Uninitialized Objects, Copy Constructors, Finding the Address of an Overloaded Function, The overload Anachronism Default Function Arguments, Default

Arguments vs. Overloading, Using Default Arguments Correctly, Function Overloading and Ambiguity. 10 Hours

UNIT –III

Operator Overloading: Operator Overloading , Creating a Member Operator Function, Creating Prefix and Postfix Forms of the Increment and Decrement Operators , Overloading the Shorthand Operators, Operator Overloading Restrictions , Operator Overloading Using a Friend Function, Using a Friend to Overload ++ or – – , Friend Operator Functions Add Flexibility, Overloading new and delete Overloading new and delete for Arrays, Overloading the nothrow Version of new and delete, Overloading Some Special Operators,. Overloading [ ], Overloading (Overloading –>,Overloading the Comma Operator .

Inheritance: Inheritance, Base-Class Access Control, Inheritance and protected Members, Protected Base-Class Inheritance, Inheriting Multiple Base Classes, Constructors, Destructors, and Inheritance, When Constructor and Destructor Functions Are Executed , Passing Parameters to Base-Class Constructors, Granting Access , Virtual Base Classes. 10 Hours

UNIT –IV

Virtual functions: Virtual Functions and Polymorphism, Virtual Functions, calling a Virtual Function Through a Base Class Reference, The Virtual Attribute Is Inherited, Virtual Functions Are Hierarchical, Pure Virtual Functions, Abstract Classes.

Templates: Templates, Generic Functions, A Function with Two Generic Types, Explicitly Overloading a Generic Function, Overloading a Function Template, Using Standard Parameters with Template Functions, Generic Function Restrictions, Applying Generic Functions, A Generic Sort, Compacting an Array Generic Classes, An Example with Two Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes, Using Default Arguments with Template Classes, Explicit Class Specializations. 10 Hours

UNIT-V

Exception Handling: Exception Handling, Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements, Handling Derived-Class Exceptions, Exception Handling, Catching All Exceptions, Restricting Exceptions, Rethrowing an Exception, Understanding terminate( ) and unexpected( ) Setting the, and Unexpected Handlers, The uncaught exception( ) Function, The exception and bad exception Classes, Applying Exception Handling .

Files and I/O : C++ File I/O and the File Classes, Opening and Closing a File, Reading and Writing Text Unformatted and Binary I/O , Characters vs. Bytes, put( ) and get( ), read( ) and write(), More get( ) Functions, getline( ), Detecting EOF, The ignore( ) Function, peek( ) and putback( ), flush( ), Random Obtaining the Current File Position, I/O Status, Customized I/O and Files. 10 Hours

Text Book:

1. The C++ Complete Reference by Herbert Schildt, 4th Edition, 2017
2. Mastering C++ by Venugopal K R, RajkumarBuyya , Tata McGraw Hill, 2nd edition, 2013.
3. C++ Programming Language by BjarneStroustrup, Addison-Wesley Professional, 4th edition, 2013.
4. C++ Primer by Barbara E Moo, Stanley B. Lippman , JoseeLajoie, Pearson Education, 4th edition, 2007.

Course Outcomes:

At the end of the course, the student is able to

- CO1: Understand and implement object-oriented programming concepts such as classes, objects, constructors, and destructors in C++. *(L2: Understand)*
- CO2: Apply operator overloading, function overloading, and inheritance techniques to solve real-world programming problems. *(L3: Apply)*
- CO3: Analyze and implement polymorphism using virtual functions and abstract classes. *(L4: Analyze)*
- CO4: Design and implement generic functions and classes using templates in C++. *(L5: Evaluate)*
- CO5: Demonstrate the ability to handle exceptions and perform file I/O operations in C++. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

MCA I SEMESTER
20SCA 705: WEB PROGRAMMING

Hours per week: 4 End Examination: 60 Marks
Credits: 4 Sessionals: 40 Marks

Preamble:

The "Web Programming" course aims to provide students with a comprehensive understanding of web development technologies. It covers HTML, CSS, JavaScript, and PHP, equipping students with the skills to create dynamic, interactive, and responsive websites. Through hands-on practice, students will learn to build web applications using client-side and server-side programming.

Course Objectives:

- To introduce students to the basic concepts of HTML for structuring web pages and adding multimedia elements.
- To teach students how to style web pages using CSS and create responsive layouts.
- To provide a deep understanding of JavaScript for adding interactivity and handling browser events.
- To introduce students to PHP for server-side scripting, form handling, and data management.
- To enable students to integrate client-side and server-side technologies to develop dynamic web applications.

UNIT-I

Introduction to HTML: Information files creation, Web Server, Web Client/Browser, Hyper TextMarkup Language, Commonly used HTML Commands.

Lists: Types of lists

Adding Graphics to HTML Documents: Using the Attributes- Border, Width, and Height, Align and Alt Attributes.

Tables: Introduction, The Caption Tag, Using the width and boarder, Cellpadding, Cellspacing,Using Background-Color property, Using the Colspan and Rowspan Attribute
12 Hours

UNIT – II

Linking Documents: Links, Images as Hyperlinks. FRAMES: Introduction to Frames.

CSS2 - Introduction, Syntax, Selectors, Color Background Cursor, Text Fonts, Lists Tables, BoxModel, Display Positioning, Floats.

Dynamic HTML: Cascading Style Sheets, Class, Using the TAG, External Style Sheets, Using the TAG. 12 Hours

UNIT – III

Introduction To JavaScript: JavaScript in web pages, The Advantages of JavaScript, WritingJavaScript into HTML, Basic Programming Techniques, Operators and Expressions in JavaScript, JavaScript Programming Constructs, Conditional Checking, Super controlled-endless loops, Functions in Java Script, User Defined Functions ,Placing Text in Browser , Dialog Boxes.

The JavaScript Document Object Model: Introduction, the JavaScript assisted style sheets DOM(JSSS DOM). 12 Hours

UNIT-IV

Understanding Objects in HTML: Browser Objects, Handling (Web page) Events Using

JavaScript. Forms used by a Web Site: The form Object, Other Built-in objects in JavaScript, User Defined Objects.

PHP: Getting Started, The Basics of PHP, Data Types, variables, Constants, Operators, Arrays, Conditional Statements, Iterations. 10 Hours

UNIT –V

Functions: User Defined Functions, Built –in Functions, PHP Server Variables. Working with Forms Introduction to HTML Form Tags and Elements, Form Elements, Adding elements to form, Uploading files to the Web Server using PHP. 10 Hours

Text Book:

1. Web Enable Commercial Application Development Using HTML, Javascript, DHTML and PHP by Ivan Bayross, BPB Publications, 4th revised edition, 2010.
2. Complete Reference HTML by T. A. Powell, 3rd edition, TMH, 2003,
3. The Complete Reference - PHP by Steven Holzner, Tata McGraw Hill, 2008.
4. Web Technology and Design by Xavier, C, New Age International, 2013.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand the basics of HTML for creating structured and multimedia-rich web pages. *(L2: Understand)*
- **CO2**: Apply CSS for styling web pages and designing responsive layouts using various properties like positioning and floats. *(L3: Apply)*
- **CO3**: Write JavaScript code to handle events, validate forms, and manipulate the DOM for dynamic user interactions. *(L4: Analyze)*
- **CO4**: Implement server-side programming using PHP for handling forms, managing data, and working with server variables. *(L5: Evaluate)*
- **CO5**: Integrate HTML, CSS, JavaScript, and PHP to develop dynamic and interactive web applications. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

MCA I SEMESTER

20SCA 707: OPERATING SYSTEMS

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Operating Systems" course provides a comprehensive understanding of how operating systems function, including their structure, processes, memory management, file systems, and I/O systems. This course equips students with the theoretical knowledge and practical skills required to manage and optimize the performance of operating systems in various computing environments.

Course Objectives:

- To introduce the basic concepts of operating systems, including architecture, structure, and operations.
- To explain the process management concepts such as process scheduling, multithreading, and inter-process communication.
- To teach synchronization techniques, including dealing with deadlocks in operating systems.
- To provide an understanding of memory management strategies, including paging, segmentation, and virtual memory.
- To familiarize students with file system management, disk scheduling, and I/O operations in operating systems.

UNIT - I

Introduction: Introduction to Operating Systems, Computer System Architecture, operating System Structure, Operating System Operations, Distributed Systems, Special Purpose Systems, Computing Environments, Open-Source Operating Systems.

System Structures: Operating System Services, User Operating Systems, System Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation Operating System Structure,Virtual Machines. 10 Hours

UNIT - II

Process Concept: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication, Communication in Client Server Systems.

Multithreaded Programming: Overview, Multithreading Models, Thread Libraries, Threading Issues, Operating System Examples.

Process Scheduling: Basic Concepts, Scheduling Criteria and Algorithms, Thread Scheduling, Multiple Processor Scheduling, Real Time CPU Scheduling. 12 Hours

UNIT -III

Synchronization: Background, Critical Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors.

Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock. 12 Hours

UNIT - IV

Memory Management Strategies: Background, Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation.

Virtual Memory Management: Background, Demand Paging, Copy on Write, Page replacement, Allocation of Frames, Thrashing, Other Considerations. 10 Hours

UNIT - V

File System: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing, Protection.

Implementing File Systems: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance.

Mass Storage Structure: Disk Structure, Disk Scheduling, Disk Management.

I/O Systems: I/O Hardware- polling, interrupts, DMA. 10 Hours

Text Book:

1. Operating System Concepts by Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Wiley Publications, 8th edition, 2012.

Reference Books:

- Operating Systems by Achyut S. Godbole, Tata McGraw Hill, 3rd edition,2010.
- Operating Systems: Internals and Design Principles by William Stallings, Pearson Education, 7th edition,2011.
- Operating Systems: A Concept-based Approach by Dhamdhere, D.M., McGraw Hill, 2nd edition, 2006.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand the fundamental concepts of operating systems, including system architecture, structure, and types of system calls. *(L2: Understand)*
- **CO2**: Apply process scheduling techniques, thread management, and inter-process communication mechanisms in operating systems. *(L3: Apply)*
- **CO3**: Analyze synchronization issues and deadlocks and implement solutions using semaphores and monitors. *(L4: Analyze)*
- **CO4**: Evaluate memory management techniques such as paging, segmentation, and virtual memory for efficient memory utilization. *(L5: Evaluate)*
- **CO5**: Understand file systems, disk scheduling, and I/O operations and implement file system management techniques. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

I SEMESTER

VDC 111: VENTURE DISCOVERY

Hours per week: 3 Examination: 100 Marks

Credits: 2

Preamble:

The "Venture Discovery" course is designed to introduce students to the process of starting a venture, from personal discovery to developing a business model and pitching a business idea. The course aims to help students discover their personal values, understand the importance of teamwork, craft innovative solutions, develop business models, and effectively communicate their venture ideas. Through experiential learning and real-world problem-solving, students will be able to apply entrepreneurial principles to develop their own ventures.

Course Objectives:

- To help students discover their personal values and align them with their entrepreneurial goals.
- To guide students through the process of crafting a mission statement and designing innovative solutions.
- To teach students how to develop a business model, define revenue models, and understand different industry dynamics.
- To provide students with the skills to validate business models and understand the impact of their ventures.
- To equip students with storytelling and pitching techniques for effectively communicating their venture ideas.

Course Outline

Unit I (6 sessions)

Personal Values: Defining your personal values, Excite & Excel, Build a Team, Define purpose for a venture. Four stages: Personal Discovery, Solution Discovery, Business Model Discovery, Discovery Integration.

Unit II (6 sessions)

Solution Discovery: Craft and mission statement, Experience design, Gaining user insight, Concept design and positioning, Product line strategy, Ideation & Impact.

Unit III (6 sessions)

Business Model Discovery: Prototyping solutions, Reality Checks, Understand your industry, Types of business models, Define Revenue Models, Define Operating Models

Unit IV (6 sessions)

Discovery Integration: Illustrate business models, Validate business models, Define company impact

Unit V (6 sessions)

Tell a Story: Can you make money, Tell your venture story.

Assessment methods

Task	Task type	Task mode	Weightage (%)
A1. Assignments	Individual	Report/Presentation	20

A2. Case / Project/Assignment	Groups* or Individual	Presentations/Report/Assignment	40
A3. Project	Individual/Group	Report/Pitch	40

Transferrable and Employability Skills

	Outcomes	Assessment
1	Know how to use online learning resources: G-Learn, online journals, etc.	A1 & A2
1	**Communicate effectively using a range of media**	A1 & A2
2	**Apply teamwork and leadership skills**	A2
3	**Find, evaluate, synthesize & use information Analyze real world situation critically**	A1 & A2
5	Analyze real world situation critically	A3
4	**Reflect on their own professional development**	A3
4	**Demonstrate professionalism & ethical awareness**	A2
5	**Apply multidisciplinary approach to the context**	A2

Learning and teaching activities

Mixed pedagogy approach is adopted throughout the course. Classroom based face to face teaching, directed study, independent study via G-Learn, case studies, projects and practical activities (individual & group)

Teaching and learning resources

Soft copies of teaching notes/cases etc. will be uploaded onto the G-learn. Wherever necessary, printouts, handouts etc. will be distributed in the class. Prescribed text book will be provided to all. However you should not limit yourself to this book and should explore other sources on your own. You need to read different books and journal papers to master certain relevant concepts to analyze cases and evaluate projects. Some of these reference books given below will be available in our library.

Prescribed Modules:

Access to NU-IDEA online modules will be provided.

Referential text books and journal papers:

Personal Discovery Through Entrepreneurship, Marc H. Meyer and Chaewon Lee, TheInstitute of Enterprise Growth, LLC Boston, MA.

Suggested journals:

Vikalpa, Indian Institute of Management, Ahmedabad
Journal of General Management, Mercury House Business Publications, Limited
Harvard Business Review, Harvard Business School Publishing Co. USA

Course Outcomes:

At the end of the course, the student is able to

- CO1: Define personal values and understand how they contribute to building a successful venture. *(L2: Understand)*
- CO2: Design innovative solutions by gaining user insights and positioning products effectively. *(L3: Apply)*
- CO3: Develop and illustrate a business model by defining revenue and operating models. *(L4: Analyze)*

- CO4: Validate business models and assess the potential impact of a company. *(L5: Evaluate)*
- CO5: Create and present a compelling story for a venture, including financial viability. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

MCA I SEMESTER

20SCA 721: OBJECT ORIENTED PROGRAMMING WITH C++ LAB

Hours per week: 3 Examination: 100 Marks

Credits: 2

Preamble:

The "Object-Oriented Programming with C++ Lab" course introduces students to the fundamental principles of object-oriented programming using C++. This lab provides hands-on experience in implementing object-oriented concepts such as classes, inheritance, polymorphism, and exception handling. Students will learn to write efficient and reusable code through the practice of function overloading, operator overloading, generic programming, and exception handling.

Course Objectives:

- To teach students the basic concepts of object-oriented programming and how to apply them using C++.
- To provide hands-on experience in using classes, objects, and constructors to structure programs.
- To familiarize students with the concepts of function and operator overloading in C++.
- To teach students how to implement inheritance, polymorphism, and virtual functions.
- To enable students to handle errors and exceptions in C++ programs, and use generic programming for code reusability.

Objective: To make the student learn object oriented concepts, programming with C++. To practice functions, classes and work with file and exception handling.

1. Write program to demonstrate classes and objects
2. Write a program to demonstrate different types of Constructors
3. Write program to demonstrate for friend class and friend functions
4. Write program for inline function, Static function, Static Classes
5. Write program to demonstrate for passing objects to a function, returning objects to function
6, Write a program to demonstrate Function Overloading
7. Write a program to demonstrate Copy Constructor
8. Write a program to demonstrate Default Arguments to a Function
8. Write a program to demonstrate different types of operator overloading
9 Write a program to demonstrate single inheritance
10 Write a program to demonstrate Multiple and Multilevel Inheritance
11. Write a program to demonstrate virtual base class
12. Write a program to demonstrate virtual functions
13. Write a program to demonstrate Generic Functions
14. Write a program to demonstrate Generic Classes
15. Write a program to demonstrate Try and catch block
16. Write a program to demonstrate Multiple catch Statements

Text Book :

1. C++ : The Complete Reference by Herbert Schildt 4th Edition, 2017

Reference Books:

1. Object Oriented Programming in C++ by E. Balaguruswamy, 4rd Edition, Tata McGraw Hill Publication.
2. Let Us C++ by YashavantP.Kanetkar, 2nd Edition, BPB Publications.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand and implement basic object-oriented concepts such as classes, objects, and constructors in C++. *(L2: Understand)*
- **CO2**: Apply function overloading, operator overloading, and inheritance in C++ programs to achieve code reusability. *(L3: Apply)*
- **CO3**: Implement polymorphism using virtual functions and virtual base classes to solve real-world problems. *(L4: Analyze)*
- **CO4**: Write generic functions and classes to implement type-independent code in C++. *(L5: Evaluate)*
- **CO5**: Handle runtime errors using exception handling techniques in C++. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

MCA I SEMESTER

20SCA 723: WEB PROGRAMMING LAB

Hours per week: 3 Examination: 100 Marks

Credits: 2

Preamble:

The "Web Programming Lab" course is designed to introduce students to the fundamentals of web development, focusing on HTML, CSS, JavaScript, and PHP. This lab course provides hands-on experience in creating web pages, designing layouts, and adding interactivity using web technologies. By the end of the course, students will be able to build dynamic and interactive websites.

Course Objectives:

- To teach students the basic concepts of web development using HTML for structuring web pages.
- To provide hands-on experience in styling web pages using CSS for layout design and visual presentation.
- To introduce JavaScript for adding interactivity and dynamic features to web pages.
- To enable students to implement server-side programming using PHP for building dynamic web applications.
- To equip students with the ability to integrate different web technologies to develop full-fledged web applications.

Objective: To learn the basics in web designing using HTML, CSS, and Java Script, PHP

1. Write a HTML document to demonstrate Basic Formatting Tags Formatting tags.
2. HTML document to demonstrate Ordered lists, unordered Lists, definition Lists.
3. Write HTML document to demonstrate Graphics and its attributes
4. Write an HTML document to create table header rows, data rows, caption and attributes of the table tag.
5. Write an HTML document to cell padding and cell spacing, Bgcolor, Colspan and Row span attribute.
6. Write an HTML document using frameset and the targeting named frames.
7. Write HTML document to demonstrate Linking Documents
8. Write CSS Program to demonstrate Create Style Sheet Background, Text Format, Controlling Fonts.
9. Write CSS Program to demonstrate Create Margin Attributes and List Attributes .
10. Write CSS Program to demonstrate various Box Model properties
11. Write a JavaScript to demonstrate different data types.
12. Write a JavaScript to demonstrate different operators.
13. Write a JavaScript to demonstrate for loop and while loop.
14. Write a JavaScript to demonstrate arrays.
15. Write a JavaScript to demonstrate dialog boxes.
16. Write a JavaScript to demonstrate user defined functions.
17. Write a JavaScript to demonstrate built-in functions.
18. Write a JavaScript to demonstrate various controls like Text Field, Button, Password Elements.
19. Write a JavaScript to demonstrate various controls Choice, Multiple Choice Options Radio Button, Check Box.
20. Write PHP Program to demonstrate variables, constants in PHP
21. Write PHP Program to demonstrate control statements.
22. Write PHP Program to demonstrate Arrays in PHP.

Text Book:

1. Web Enable Commercial Application Development Using HTML, Java script, DHTML and PHP by Ivan Bayross, BPB Publications, 4th revised edition, 2010
.

Reference Books:

1. Complete Reference HTML by T. A. Powell, 3rd edition, TMH, 2003.
2. HTML, XHTML, and CSS Bible by Steven M. Schafer, Wiley India, 5th Edition.
3. Beginning CSS: Ca20SCAding Style Sheets for Web Design by Ian Pouncey, Richard York, Wiley India.
4. Web Technology and Design by Xavier, C, New Age International, 2013.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand and use basic HTML tags for structuring and formatting web pages. *(L2: Understand)*
- **CO2**: Apply CSS for styling web pages and controlling the layout using margin, list, and box model properties. *(L3: Apply)*
- **CO3**: Write JavaScript programs to add interactivity to web pages through loops, arrays, and dialog boxes. *(L4: Analyze)*
- **CO4**: Implement form controls and event handling using JavaScript. *(L5: Evaluate)*
- **CO5**: Develop dynamic web applications using PHP for server-side scripting and database interaction. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

MCA I Semester

20SCA 725: Operating Systems Lab

Hours per week: 3 Examination: 100 Marks

Credits: 2

Preamble:

The "Operating Systems Lab" course aims to introduce students to the Unix environment and provide practical experience in using various Unix utilities, shell programming, and process management techniques. Students will gain hands-on experience in writing shell scripts and implementing key operating system concepts such as process creation, process scheduling, and file handling.

Course Objectives:

- To familiarize students with Unix commands and file system utilities for file and process management.
- To teach students how to write and execute shell scripts using various control structures and utilities.
- To provide practical exposure to process creation and management in Unix, including working with chains and fans of processes.
- To implement and simulate process scheduling algorithms such as FCFS, Round Robin, Priority, and Shortest Job First.
- To enhance students' problem-solving skills by applying operating system concepts in real-time Unix environments.

Unix Utilities – Introduction to Unix file system, vi editor, file handling utilities, security by file permissions, process utilities, disk utilities, networking commands, cp, mv ln, rm, unlink, mkdir, rmdir, du, df, mount, umount find, unmask, ulimit, ps, who, finger, arp, ftp, telnet, rlogin, text processing utilities and backup utilities, detailed commands to be covered are cat, tail, head, sort, nl, uniq, grep, egrep, fgrep, cut, paste, join, tee, pg, comm., cmp, diff, tr, awk, tar,cpio.

1. Study about the General Purpose Utilities.
   a) Banner b) cal c) date d) calendar e) tty f) bc g) spell &fspell
2. Write a shell program using if, nested if
3. Write a shell program using switch case
4. Write a shell program to find the Sum of digits in a 3 digit number using while loop.
5. Write a shell program to print first 'n' terms of Fibonacci series using for loop

Programs on Processes:

1. Chain of processes.
2. Fan of processes.

Programs on Process Scheduling:

1. FCFS scheduling algorithm
2. Round Robin scheduling algorithm
3. Priority scheduling algorithm (Pre-emptive, Non Pre-emptive)
4. Shortest job First scheduling algorithm (Pre-emptive, Non Pre-emptive)

Reference Books:

1. Unix Concepts and Applications by Sumitabha Das, Tata McGraw Hill, 4th edition,2006.
2. Unix networking program by Stevens W. Richard,2005.
3. Advanced Unix programming by H.J.Rechkind, Pearson Education, 2nd edition,2004.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand and use various Unix file handling, text processing, and networking utilities in a Unix environment. *(L2: Understand)*
- **CO2**: Write shell scripts using control structures like if, switch case, while, and for loops to solve simple computational problems. *(L3: Apply)*
- **CO3**: Manage processes in Unix by creating and controlling process chains and fans, utilizing process utilities. *(L4: Analyze)*
- **CO4**: Implement and simulate process scheduling algorithms such as FCFS, Round Robin, Priority, and Shortest Job First. *(L5: Evaluate)*
- **CO5**: Demonstrate the ability to handle complex operating system tasks such as file system navigation, backup operations, and process scheduling using Unix commands. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

II SEMESTER

20SCA 702: PROBABILITY AND STATISTICS

Hours per week: 4 End Examination: 60Marks

Credits: 4 Sessional: 40 Marks

Preamble:

The "Probability and Statistics" course provides a comprehensive introduction to the fundamental principles of probability theory and statistical analysis. It covers key topics such as random variables, probability distributions, correlation and regression, sampling techniques, and hypothesis testing. This course equips students with the necessary tools to analyze and interpret data in various fields, especially in computer science and data analysis.

Course Objectives:

- To introduce students to the basic concepts of probability, including sample spaces, events, and conditional probability.
- To teach students about discrete and continuous random variables and their distributions.
- To develop an understanding of correlation, regression, and curve fitting techniques.
- To familiarize students with different sampling methods and estimation techniques.
- To provide students with knowledge of hypothesis testing and its applications in data analysis.

UNIT - I

Probability: Sample Space, Events, Axiomatic Approach to Probability, Conditional Probability Independent Events, Baye's Formula with Applications.

Random Variables: Continuous and Discrete Random Variables, Distribution Function of a random variable, Expectation, Variance, Coefficient of Variation, Moment Generation Function. 10 Hours

UNIT - II

Probability Distribution: Discrete Distributions, Binomial, Poisson and Geometric Distributions, Continuous Distributions, Uniform, Normal, Exponential. 10 Hours

UNIT - III

Correlation and Regression: Correlation Coefficient, Rank Correlation Coefficient of Determination, Linear Regression, Methods of Least Squares, Fitting of the Curve of the Form **ax + b, $ax^2 + bx + c$, ab^x , ax^b and ae^{bx}.** 10 Hours

UNIT - IV

Sampling Theory: Concepts of Sampling, Methods of Sampling, Simple Random Sampling, Systematic Sampling and Stratified Random Sampling (Descriptions Only), Concepts of Sampling Distributions and Standard Error, Point Estimation (Concepts only), Interval Estimation of Mean and Proportion. 10 Hours

UNIT - V

Test of Hypotheses: Critical Region, Two Types of Errors, Level of Significance, Large Sample Tests for Mean & Proportion, Exact Tests Based on t, F and Chi – Square Distributions. 10 Hours

Text Book:

1. Fundamentals of Mathematical Statistics by S.C. Gupta & V.K. Kapoor, Sultan Chand & Sons, 2002.

Reference Books:

1. Probability and Statistics for Engineers by Irwin Millor and John E.Freund, PHI.

2. Probability and Statistics, Spiegel, TMH.
3.P. G. Hoel, S. C. Port and C. J. Stone, Introduction to Probability Theory, Universal Book Stall, 2003 (Reprint).

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand the basic concepts of probability, including sample spaces, events, and conditional probability, and apply Bayes' theorem to solve problems. *(L2: Understand)*
- **CO2**: Analyze and compute probabilities for discrete and continuous random variables and their associated distributions, such as binomial, Poisson, and normal distributions. *(L4: Analyze)*
- **CO3**: Apply correlation and regression techniques to study the relationship between variables and fit curves using the method of least squares. *(L3: Apply)*
- **CO4**: Understand and apply various sampling methods, including simple random sampling and stratified sampling, to real-world data. *(L2: Understand)*
- **CO5**: Conduct hypothesis testing for large and small samples, using t-tests, F-tests, and chi-square tests to make data-driven decisions. *(L5: Evaluate)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

II SEMESTER

20SCA 704: DATA STRUCTURES USING C++

Hours per week: 4 End Examination: 60 Marks

Credits:4 Sessionals: 40 Marks

Preamble:

The "Data Structures Using C++" course introduces students to the fundamental concepts of data structures and their implementation using C++. The course covers essential data structures such as arrays, linked lists, stacks, queues, trees, heaps, and graphs, along with sorting and searching algorithms. It emphasizes both theoretical understanding and practical application of data structures for efficient problem-solving.

Course Objectives:

- To introduce students to the types of data structures and their role in algorithmic problem-solving.
- To teach students how to implement and manipulate linear data structures such as arrays, stacks, and queues.
- To familiarize students with dynamic memory management and linked data structures.
- To provide an understanding of non-linear data structures like trees and graphs, and their applications.
- To teach students various sorting and searching techniques and analyze their efficiency.

UNIT -I

Fundamental Concepts: Introduction to Data Structures, Types of Data Structures, Relationship among data, data structures and algorithms, Implementation of data structures, Analysis of Algorithms, Complexity of algorithms: Space complexity, Time complexity Definition.

Linear Data Structure using Arrays: Sequential Organization, Linear Data Structure Using Sequential Organization: Array ADT, Memory Representation, Class Array, Multidimensional Arrays, Concept of Ordered List, Single Variable Polynomial: Representation, evaluation, Addition, Multiplication, Sparse Matrix: Representation, Addition, Transpose, String Manipulation Using Array, Pros and Cons of Arrays.

Searching: Search Techniques: Sequential search, Binary search, Hashed search.

Sorting: Types of sorting, General sort concepts, Bubble sort, Insertion sort, Selection sort, Quick sort, Merge sort, Heap sort. 15 Hours

UNIT - II

Stacks: Primitive operations, Stack Abstract Data Type, Representation of Stacks, Using Sequential Organization, Multiple Stacks, Applications of Stacks – Expression Evaluation and Conversion, Processing of Function Calls.

Queues: Concept of Queues, Queue as Abstract Data Type, Realization of Queues Using Arrays, Circular Queue, Multi queues, Deque, Priority Queue, Applications of Queues: Job scheduling. 10 Hours

UNIT -III

Linked Lists: Introduction, Linked List: Comparison of sequential and Linked Organizations, Terminology, Primitive operations, Realization of Linked Lists using arrays and dynamic memory management, Dynamic memory management in C++, Linked List Abstract Data Type, Linked List Variants, Doubly Linked List: Creation, Deletion, Insertion, Traversal, Circular Linked List, Linked Stack, Linked Queue. 10 Hours

UNIT - IV

Trees: Introduction, Basic terminology, General trees, Representation of a general tree, Types of Trees, Binary Tree, Properties, Binary Tree Abstract Data Type, Realization of a Binary

Tree, Insertion of a Node in Binary Tree, Binary Tree Traversal (recursive traversals), Formation of binary tree from its traversals, Binary Search Tree: Inserting a node, Searching for a key, Deleting a node, Binary Tree and Binary Search Tree, Applications of Binary Trees: Expression tree, Decision tree, Huffman's coding.

Heaps: Basic Concepts, Implementation of Heap, Heap as Abstract Data Type. 8 Hours

UNIT - V

More on Linked Lists: Copying a linked list, Computing the length of a linked list, reversing singly linked list without temporary storage, concatenating two linked lists, Erasing the linked list.

Graphs: Introduction, Graph Abstract Data Type, Graph Representation, Graph t raversals, Spanning Trees: Prim's,Krushkal's Algorithm. (7)

Text Books:

1. Data Structures using C++ by Varsha H.Patil, Oxford University Press,2012.

Reference Books:

1. Data Structures Algorithms and Applications in C++ by SartajSahani, University Press, 2nd Edition, 2011.
2. Data Structures Using C and C++ by YedidyahLangsam, Moshe J Augenstein and Aaron M Tenenbaum, PHI, 2nd Edition, 2009.
3. Data Structures and Algorithm Analysisin C++by Mark Allen Weiss, Pearson Education, 3rd edition, 2007.
4. Data Structures and Algorithms in C++ by Adam Drozdek, Cengage Learning, 4th Edition, 2013.

Course Outcomes:

At the end of the course, the student is able to

1. **CO1**: Understand the basic concepts of data structures, including types, algorithms, and complexity analysis. *(L2: Understand)*
2. **CO2**: Apply linear data structures such as arrays, stacks, and queues in problem-solving, and implement sorting and searching algorithms. *(L3: Apply)*
3. **CO3**: Implement linked data structures, including singly linked lists, doubly linked lists, and circular linked lists. *(L4: Analyze)*
4. **CO4**: Apply tree data structures, such as binary trees and binary search trees, for efficient data organization and manipulation. *(L5: Evaluate)*
5. **CO5**: Implement graph-based algorithms such as graph traversal, spanning trees, and apply linked list-based operations. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

II SEMESTER

20SCA 706: DATABASE MANAGEMENT SYSTEMS

Hours per week: 4 End Examination: 60 Marks

Credits:4 Sessionals: 40 Marks

Preamble:

The "Database Management Systems" course provides an in-depth understanding of database systems, focusing on conceptual modeling, relational data models, SQL, database design, and transaction management. This course equips students with the skills needed to model, design, and implement databases, while addressing issues related to concurrency control, normalization, and recovery in database systems.

Course Objectives:

- To introduce the fundamental concepts of database systems, including database architectures, data models, and database management.
- To teach data modeling techniques using ER models and relational databases.
- To provide students with hands-on experience in querying databases using SQL.
- To explain relational database design principles, including normalization and the use of functional dependencies.
- To familiarize students with transaction management, concurrency control, and recovery techniques in database systems.

UNIT - I

Introduction and Conceptual Modelling, Databases and Database Users: Introduction, Characteristics of Database Approach, Actors on the Scene, Workers behind the Scene, Advantages of using DBMS Approach.

Database System, Concepts and Architecture: Data Models, Schemas and Instances, Three Schema Architecture and Data Independence, Database Language and Interfaces, The Database System Environment, Centralized and Client/Server Architecture of Database Management Systems, Classification of Database Management Systems. 10 Hours

UNIT - II

Data Modeling Using The ER Model: High Level Conceptual Data Models for Database Design, Entity Types, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Roles and Structural Constraint, Weak Entity Types, ER Diagrams, Naming Conventions and Design Issues. The Enhanced Entity Relationship model, UML Class Diagrams, Relationship Types of Degree Higher Than Two.

The Relational Data Model and Relational Database Constraints: Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update Operations, Transactions and Dealing with Constraints Violations. 12 Hours

UNIT – III

The Relational Algebra: Unary Relational Operations - SELECT and PROJECT, Relational Algebra Operations from Set Theory, Binary Relational Operations - JOIN and DIVISION; Additional Relational Operations, Examples of Queries in Relational Algebra.

Relational Database Design By ER And EER To Relational Mapping: Relational Database Design using ER to Relational Mapping, Mapping EER Model Constructs to Relations. 8 Hours

UNIT – IV

SQL-Schema Definition, Constraints, Queries and Views: SQL Data Definition and Data types, Specifying Constraints in SQL, Schema Change Statements in SQL, Basic Queries in SQL, More Complex SQL Queries, INSERT, DELETE, UPDATE Statements in SQL, Additional Features of SQL, Specifying Constraints As Assertions and triggers, Views, Additional features of SQL.

Functional Dependencies and Normalization for Relational Databases: Informal Design Guidelines for Relational Schemas, Functional Dependencies, Normal Forms Based on Primary Keys, General Definitions of 2nd and 3rd Normal Forms, Boyce-Codd Normal Form

Relational Database Design Algorithms and Further Dependencies: Properties of relational Decomposition, Algorithms for Relational Database Schema design, Multi valued Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form. 12 Hours

UNIT – V

Transaction Processing Concepts: Introduction to Transaction Processing, Transaction and System Concepts, Desirable properties of transactions, Characterizing Schedules based on recoverability, Characterizing Schedules based on serializability

Concurrency Control Techniques: Two Phase Locking, Time Stamp Ordering, Multi version concurrency control, Optimistic Concurrency control

Database Recovery Techniques: Recovery concepts, Recovery Techniques based on Deferred and Immediate Update, Aries Recovery algorithm 10 Hours

Text Books:

1. Fundamentals of Database Systems by Ramez Elmasri and Shamkant B. Navathe, Pearson education, 5th edition, 2009.

Reference Books:

1. Database Concepts by Abraham Silberschatz, Henry F Korth, S.Sudarshan, TMH, 6th edition,2014**.**

2. An Introduction to Database Systems by C.J. Date, Addison Wesley, 8thedition, 2008.

3. Database Management Systems by Raghu Ramakrishnan, Johannes Gehrke, TMH, 2nd edition, 2000.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand the concepts, architecture, and advantages of database management systems. *(L2: Understand)*
- **CO2**: Apply data modeling techniques using the ER and EER models to design relational databases. *(L3: Apply)*
- **CO3**: Write complex queries using SQL, including schema definition, constraints, and views. *(L4: Analyze)*
- **CO4**: Implement database normalization techniques to ensure the integrity and efficiency of relational databases. *(L5: Evaluate)*
- **CO5**: Apply transaction processing concepts, concurrency control, and recovery techniques in database systems. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

II SEMESTER

20SCA 708: OBJECT ORIENTED SOFTWARE ENGINEERING

Hours per week: 4 End Examination: 60 Marks
Credits: 4 Sessionals: 40 Marks

Preamble:
The "Object Oriented Software Engineering" course introduces students to the principles and practices of software engineering with an emphasis on object-oriented methodologies. It covers the software development lifecycle, including modeling with UML, project management, system design, object design, and testing. The course aims to equip students with the necessary skills to develop and manage software projects using object-oriented techniques.

Course Objectives:

- To introduce students to software engineering concepts and the challenges of software development.
- To teach students how to model software systems using UML and manage project communication and organization.
- To provide a comprehensive understanding of system design, including decomposing systems and addressing design goals.
- To enable students to apply object-oriented design principles, including the reuse of design patterns and specifying interfaces.
- To familiarize students with software testing techniques and the essentials of project management.

UNIT-I

Software Engineering: Software related problems, software engineering, concepts, development activities.
Modelling: Concepts, Modelling with UML. 10 Hours

UNIT-II

Project Organization & Communication: Project Organization & communication concepts and their activities.
Requirements: Requirements elicitation & its activities and managing requirements elicitation.
Analysis: Analysis overview, concepts, activities and managing analysis. 10 Hours

UNIT-III

System Design: Decomposing the System: System Design overview, System design concepts, and System design Activities, and managing System Design
System design: Addressing design goals: An overview of system design activities and concepts UML Development diagram, System design goals, Managing system design. 10 Hours

UNIT-IV

Object Design: Reusing Pattern Solutions: An overview of object design Reuse Concepts, Solution objects, inheritance and design patterns.
An Object Design: Specifying Interfaces: An overview of interface specification, interface specifications concepts & its activities and Managing object design. 10 Hours

UNIT-V

Testing: Testing concepts, activities and managing testing.
Project Management -Introduction, An overview of project management, Project Management Concepts, Project Management Activities. 10 Hours

Text Book:

1. Object-Oriented Software Engineering: Using UML, Patterns and Java, Bernd Bruegge and Allen H. Dutoit, 2nd Edition, Pearson Education Asia.

Reference Books:

1. Object-Oriented Software Engineering: Practical software development using UML and Java Timothy C. Lethbridge and Robert Laganiere , McGraw-Hill Higher education
2. An Introduction to Object Oriented Systems Analysis and Design with UML and the Unified Process, Stephen R Schach, Tata McGraw-Hill

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand the key concepts of software engineering and apply UML modeling techniques. *(L2: Understand)*
- **CO2**: Organize and manage project communication and requirements elicitation during software development. *(L3: Apply)*
- **CO3**: Implement system design by decomposing systems and addressing design goals. *(L4: Analyze)*
- **CO4**: Apply object-oriented design principles, reuse design patterns, and specify interfaces for software systems. *(L5: Evaluate)*
- **CO5**: Develop testing strategies and manage software projects effectively using project management tools. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	3
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

II SEMESTER

20SAE 702: TECHNICAL COMMUNICATION SKILLS

Hours per week: 4 End Examination: 60 Marks
Credits: 3 Sessionals: 40 Marks

Preamble:
The "Technical Communication Skills" course aims to develop the communication skills required for professional and technical environments. This course focuses on enhancing both verbal and non-verbal communication, addressing barriers to effective communication, and equipping students with the tools for clear and impactful technical communication. Through topics such as group discussions, interviews, and technical presentations, students will develop the skills needed for effective professional interaction.

Course Objectives:

- To help students identify and correct common language errors in Indian English.
- To differentiate between informal conversation and formal communication in various professional settings.
- To teach students the principles of effective communication, including verbal, non-verbal, and written forms.
- To provide students with the skills required to participate in interviews, group discussions, and technical presentations.
- To enhance students' ability to prepare and write professional business letters and emails.

UNIT - I

Features of Indian English: Correction of sentences, Structures, Tenses, ambiguity, idiomatic distortion and Misappropriation. 8 Hours

UNIT - II

Informal conversation Vs Formal expression: Features of good communication, Different flows of communication, Verbal and Nonverbal communication, Barriers to effective communication –ways to overcome the barriers. 10 Hours

UNIT - III

Types of Communication: Oral, aural, Writing and reading, Word-Power, Vocabulary, Jargon rate of speech, pitch, tone, Clarity of voice, Group discussion, Personality traits, types of group discussion, Team player, Leadership qualities. 12 Hours

UNIT - IV

Formal and informal interviews: Ambiance and polemics, interviewing in different settings and for different purposes e.g., Eliciting and giving information, Preparation for a job interview, Personality traits assessment, Recruiting, Performance appraisal. 12 Hours

UNIT - V

Technical presentations: Types of presentation, Video conferencing, Participation in meetings, chairing sessions. Letter-writing, business letters, Proforma, Format, Style, Effectiveness, Promptness, Analysis of sample letters collected from industry, email. 12 Hours

Text Book:

1. Essentials of Business Communication by Rajendra Pal & J S KorlahaHi, S.Chand Sons.
2. Advanced Communication Skills by V. Prasad, Atma Ram Publications.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Identify and correct common errors in Indian English and understand the nuances of sentence structure, tenses, and idiomatic expressions. *(L2: Understand)*

- **CO2**: Demonstrate an understanding of the differences between informal conversation and formal communication, and apply strategies to overcome communication barriers. *(L3: Apply)*
- **CO3**: Apply verbal and non-verbal communication skills in group discussions and develop leadership and teamwork qualities. *(L4: Analyze)*
- **CO4**: Prepare and participate in interviews, understanding the dynamics of various interview settings and performance assessments. *(L5: Evaluate)*
- **CO5**: Develop proficiency in writing professional business letters, emails, and making technical presentations. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	1	1	1	1	2
CO2	3	3	3	2	1	1	1	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

II SEMESTER

20SCA 722: DATA STRUCTURES USING C++ LAB

Hours per week: 3 Examination: 100 Marks
Credits: 2

Preamble:
The "Data Structures Using C++ Lab" course is designed to provide students with hands-on experience in implementing and applying fundamental data structures such as arrays, linked lists, stacks, queues, trees, and graphs. Through this lab, students will also develop proficiency in implementing sorting algorithms and expression evaluators, using C++. The course aims to strengthen students' understanding of both linear and nonlinear data structures and how they can be applied to solve real-world computational problems.

Course Objectives:
- To introduce students to fundamental data structures such as arrays, linked lists, stacks, queues, trees, and graphs.
- To provide hands-on experience in implementing sorting algorithms and understanding their computational efficiency.
- To familiarize students with linear and nonlinear data structures and how they can be used in solving practical problems.
- To enhance the students' problem-solving skills by applying data structures in algorithms for expression evaluation, searching, and traversals.
- To teach students how to manage memory efficiently while working with dynamic data structures like linked lists and trees.

Objective: The aim of this lab to make the student understand linear and nonlinear data structures, to apply the data structures and algorithms in real time applications.

1. Implementation of Linear Search, Binary Search.
2. Implementing the following sorting methods.
   a.Bubble sort , b. Insertion sort, c. Selection Sort, d. Quick Sort, e. Merge Sort
3. Implementation of Polynomial Addition using Arrays.
4. Implementation of Sparse Matrix addition and Multiplication using Arrays
5. Array implementation of stack.
6. Array implementation of Queue.
7. Implementation to convert infix expressions to post fix notation, prefix notation.
8. Simple expression evaluator that can handle +,-, /,*.
9. Singly Linked List operations – insertion, deletion, display, reversal
10. Implementation of Linked Stack Operations.
11. Implementation of Linked Queue Operations.
12. Implementation of circular queue ADT using an array.
13. Implementation of Doubly LinkedList.
14. Implementation of Circular LinkedList.
15. Implementation of Binary Tree, Binary Search Tree creation, traversals.
16. Implementation of Binary Search Tree operations, insertion all cases and deletion all cases.
17. Implementation of Graph Traversals.

Reference Books:

1. Data Structures through C++ by Varsha H Patil, Oxford University Press, New Edition, 2011.
2. Data Structures through C in depth by S.K. Srivastva and DeepaliSrivastva,BPB publications,2004.

3. Data Structures and Algorithms in C++ by Adam Drozdek, Cengage Learning, 4th edition, 2013.

Course Outcomes:

At the end of the course, the student is able to

- CO1: Implement searching algorithms such as linear and binary search, and analyze their time complexity. *(L3: Apply)*
- CO2: Apply sorting algorithms including bubble sort, insertion sort, selection sort, quick sort, and merge sort, and compare their performance. *(L4: Analyze)*
- CO3: Develop and implement linear data structures such as stacks, queues, and linked lists, using both array and pointer-based implementations. *(L3: Apply)*
- CO4: Implement tree and graph data structures and perform traversals, insertions, deletions, and other related operations. *(L5: Evaluate)*
- CO5: Use data structures to solve real-world problems, such as expression conversion and evaluation, polynomial addition, and sparse matrix operations. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	1	2	2
CO3	3	3	3	3	2	2	2	3
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

II SEMESTER

20SCA 724 : DATABASE MANAGEMENT SYSTEMS LAB

Hours per week: 3 Examination: 100 Marks

Credits: 2

Preamble:

The "Database Management Systems Lab" course aims to provide students with practical experience in working with relational databases, SQL, and PL/SQL. The course focuses on data manipulation, defining and managing database schema, writing queries, and implementing database operations through SQL commands. Students will also gain knowledge in advanced database concepts such as views, joins, subqueries, and procedural SQL.

Course Objectives:

- To introduce students to the basic concepts of SQL and the ability to create, modify, and manipulate relational databases.
- To teach the implementation of integrity constraints and various DML operations for database management.
- To provide hands-on experience in querying databases using built-in functions, set operators, and joins.
- To enable students to create schema objects such as views, sequences, and indexes, and use DCL commands for transaction control.
- To develop proficiency in writing PL/SQL programs, including loops, conditional statements, and subqueries.

Objective: The aim of this lab is to make the student learn the concepts of SQL. Demonstrates on various DDL,DML and DCL statements. Student also will learn PL/SQL

1. Creation, altering and dropping of tables using SQL.
2. Implementing integrity Constraints on tables.
3. Implementing DML Operations using SQL- Insert, Delete, Update.
4. Simple Queries to access data from Tables using Select Statement and where condition using Distinct, And, Or, Not and Order By Operators.
5. Queries Using Built in Functions:
    i. Arithmetic Functions: Sign, Abs, Ceil, Floor, Exp, Power, Log, Sqrt,
    ii. String Functions: Concat, Lpad, Rpad, Ltrim, Rtrim, Lower, Upper, Initcap, Length, Substr and Instr.
    iii. Date Functions: Sysdate, Next_Day, Add_Months, Last_Day, Months_Between, Least, Greatest, Trunc, Round
    iv. Aggregate Functions: Count, Sum, Avg, Max And Min, Group by, Having,
    v. Queries Using Conversion Functions: To_Char, To_NumberandTo_Date
    vi. Queries Using Set Operators: Union, Intersect, Minus
    vii. Queries Using Joins, Natural Join, Innerjoin, OuterJoin.
6. Queries Along with Sub Queries and Correlated Queries using Any, All, In, Exists, Notexists.
7. Creating Other Schema Objects**:** Defining Views, Creating Views, using Views to Change Data, Dropping Views, Creating Indexes andSequences.
8. Using DCL Commands: Commit andRollback.
9. Creation of Simple PL/SQL Program which includes Declaration Section, Executable Section, Select ... intoClause

10. Develop Programs that include Features of Nested If andCase.
11. Develop Programs using While Loop, For Loop, Nested Loops

Text Books:

1. SQL, PL/SQL The programming languageofORACLE by Ivan bayross, BPB publications, 4th edition, 2009.
2. Programming Oracle triggers and Stored Procedures by Kevin Owens, PHI, 3rd Edition, 2003.

Course Outcomes:

At the end of the course, the student is able to

- CO1: Create, alter, and drop tables using SQL and implement integrity constraints on tables. *(L3: Apply)*
- CO2: Perform DML operations such as Insert, Delete, and Update using SQL and manage data effectively. *(L3: Apply)*
- CO3: Write SQL queries using built-in functions, aggregate functions, set operators, and joins to retrieve data efficiently. *(L4: Analyze)*
- CO4: Create views, indexes, and sequences, and utilize DCL commands for transaction control in databases. *(L5: Evaluate)*
- CO5: Develop and execute PL/SQL programs involving loops, conditional statements, and nested structures for procedural operations. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	1	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

III SEMESTER

20SCA 801: Object Oriented Programming with JAVA

Hours per week: 4 End Examination: 60 Marks
Credits: 4 Sessionals: 40 Marks

Preamble:

The "Object-Oriented Programming with JAVA" course introduces students to the principles of object-oriented programming (OOP) through Java, a widely-used and versatile programming language. This course covers fundamental concepts such as inheritance, polymorphism, exception handling, multithreading, collections framework, and GUI programming using Swing. By the end of the course, students will develop the ability to design, implement, and deploy robust Java applications that incorporate OOP principles and GUI elements.

Course Objectives:

- To introduce students to the fundamental object-oriented programming concepts and their implementation in Java.
- To teach students how to apply inheritance, polymorphism, and interfaces in object-oriented software development.
- To familiarize students with exception handling, multithreading, and concurrent programming techniques in Java.
- To provide hands-on experience with the Java Collections Framework for managing and manipulating groups of objects.
- To develop the ability to create GUI-based applications using Java Swing, incorporating event handling and layout management.

UNIT – I

Introduction to Object-oriented concepts- Object-Oriented concepts, An Overview of Java, Data types, Variables and Arrays, operators, expressions, control statements, Classes, and Methods

UNIT – II

Inheritance and Polymorphism: Inheritance, Constructors, Creating Multilevel hierarchy, super uses, using final with inheritance, Polymorphism, method overriding, abstract classes, Object class, forms of inheritance- benefits of inheritance, costs of inheritance. 10 Hours

Packages- Defining a Package, Class path, Access protection, importing packages. Interfaces- defining an interface, implementing interfaces, Nested interfaces, applying interfaces, variables in interfaces and extending interfaces. 10 Hours

UNIT - III

Exception handling - Fundamentals of exception handling, Exception types, Termination or resumptive exception models, Uncaught exceptions, using try and catch, multiple catch clauses, nested try statements, throw, throws and finally, built- in exceptions, creating own exception sub classes. **Multithreading**- Differences between thread-based multitasking and process-based multitasking, Java thread model, creating threads, thread priorities, synchronizing threads, inter thread communication. 12 Hours

UNIT – IV

The Collections Framework (java.util)- Collections overview, Collection Interfaces, The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Accessing a Collection via an Iterator, Using an Iterator, The For-Each alternative, Map Interfaces and Classes, Comparators, Collection algorithms, Arrays, The Legacy Classes and Interfaces- Dictionary, Hashtable, Properties, Stack, Vector More Utility classes, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter, Scanner 10 Hours

UNIT - V

GUI Programming with Swing– Introduction, limitations of AWT, MVC architecture, components, containers. Understanding Layout Managers, Flow Layout, Border Layout, Grid Layout, Card Layout, Grid Bag Layout. Event Handling- The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes. A Simple Swing Application,

Applets – Applets and HTML, Security Issues, Applets and Applications, passing parameters to applets. Creating a Swing Applet, Painting in Swing, A Paint example, Exploring Swing Controls- JLabel and Image Icon, JText Field, The Swing Buttons- JButton, JToggle Button, JCheck Box, JRadio Button, JTabbed Pane, JScroll Pane, JList, JCombo Box, Swing Menus, Dialogs. 12 Hours

Text Books:

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.

Reference Books:

1. An Introduction to programming and OO design using Java, J. Nino and F.A. Hosch, John Wiley & sons.
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
3. Object Oriented Programming through Java, P. Radha Krishna, Universities Press.
4. Programming in Java, S. Malhotra, S. Chudhary, 2nd edition, Oxford Univ. Press.
5. Java Programming and Object oriented Application Development, R. A. Johnson, Cengage Learning.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand object-oriented programming concepts, Java basics, and apply them to design simple Java applications. *(L2: Understand)*
- **CO2**: Apply inheritance, polymorphism, and interfaces to design reusable and modular Java programs. *(L3: Apply)*
- **CO3**: Implement exception handling and multithreading techniques in Java applications. *(L4: Analyze)*
- **CO4**: Utilize the Java Collections Framework to manage and manipulate data in Java programs. *(L5: Evaluate)*
- **CO5**: Develop GUI-based applications using Java Swing, with effective event handling and layout management. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	3
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

III SEMESTER

20SCA 803: DESIGN AND ANALYSIS OF ALGORITHMS

Hours per week: 4 End Examination: 60 Marks
Credits: 4 Sessionals: 40 Marks

Preamble:

The "Design and Analysis of Algorithms" course provides a deep understanding of the principles and techniques involved in designing efficient algorithms for solving computational problems. It covers fundamental algorithmic strategies such as divide and conquer, greedy methods, dynamic programming, and backtracking, along with graph traversal techniques. The course also explores advanced topics like NP-completeness, branch and bound methods, and randomized algorithms.

Course Objectives:

- To introduce the basic principles of algorithm design and performance analysis.
- To teach algorithmic strategies such as divide and conquer, greedy methods, and dynamic programming for solving computational problems.
- To familiarize students with various graph traversal techniques and their applications in solving problems.
- To explain advanced problem-solving methods like backtracking and branch and bound.
- To provide an understanding of NP-hard and NP-complete problems and the significance of these classes in algorithm design.

UNIT – I

Introduction To Algorithms: Algorithm Specification, Performance Analysis, Introduction To Randomized Algorithms. Divide And Conquer: The General Method, Binary Search, Finding Maximum & Minimum, Quick Sort, Selection, Strassen's Matrix Multiplication. 10 Hours

UNIT – II

Greedy Method: General Method, Knapsack Problem, Tree Vertex Splitting, Job Sequencing With Deadlines, Minimum Cost Spanning Trees, Single Source Shortest Paths. 10 Hours

UNIT – III

Dynamic Programming: The General Method, Multistage Graphs, All Pairs Shortest Paths, Optimal Binary Search Trees, String Editing, Reliability Design, the Traveling Sales Person Problem. 10 Hours

UNIT – IV

Basic Traversal And Search Techniques: Techniques For Graphs, Connected Components And Spanning Trees, Bi-Connected Components and DFs. **Back Tracking**: General Method, Eight Queens Problem, Sum Of Subsets, Graph Coloring, Hamiltonian cycles. 10 Hours

UNIT – V

Branch and Bound: The method, 0/1 knapsack problem, traveling salesperson problem algebraic problems: the general method, evaluation and interpolation. np hard and np complete problems: basic concepts. 10 Hours

Text Book:

1. Fundamentals of Computer Algorithms By Ellis Horowitz, SartajSahni, Sanguthevar Rajasekaran, University Press, 2nd Edition, 2008.

Reference Books:

1. Fundamentals of AlgorithmicsBy G. Brassard AndP.Bratley, Phi, 2011.
2. Introduction to Algorithms By T.H. Cormen, C.E. Leiserson, R.L.Rivest, 3rd Edition, PHI,2010.
3. Introduction to Design Analysis of Algorithms By AnanyLevitin, 2nd Edition, Pearson

Publications, 2009.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand the fundamentals of algorithm design, including performance analysis and the divide and conquer technique. *(L2: Understand)*
- **CO2**: Apply greedy methods to solve optimization problems such as the knapsack problem and minimum spanning trees. *(L3: Apply)*
- **CO3**: Analyze dynamic programming approaches for problems like the traveling salesperson problem and all-pairs shortest paths. *(L4: Analyze)*
- **CO4**: Implement graph traversal techniques and backtracking methods for solving problems such as the eight queens problem and graph coloring. *(L5: Evaluate)*
- **CO5**: Evaluate the complexity of NP-hard and NP-complete problems and apply branch and bound techniques to solve them. *(L5: Evaluate)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	3
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

III SEMESTER

20SCA 805: DATA MINING

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Data Mining" course offers an introduction to the concepts, techniques, and applications of data mining, focusing on the process of extracting useful patterns from large datasets. The course covers key topics such as data preprocessing, frequent pattern mining, classification, clustering, and outlier detection. By the end of the course, students will be equipped with the knowledge and skills required to apply data mining techniques to real-world data analysis tasks.

Course Objectives:

- To introduce the fundamental concepts of data mining, including data types, patterns, and technologies.
- To provide an understanding of data preprocessing techniques and their importance in data mining.
- To familiarize students with frequent pattern mining methods and the evaluation of interesting patterns.
- To explain classification techniques, including decision trees, Bayesian classification, and methods for improving classification accuracy.
- To introduce clustering methods, outlier detection techniques, and their applications in data mining.

UNIT – I

Introduction: What is Data Mining? Kind of data on which mining is done, Kinds of patterns can be mined, and Technologies used, Kinds of Applications targeted, Major Issues of Data mining.

Getting to Know Your Data: Data Objects and Attribute, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity. 15 Hours

UNIT - II

Data Pre-processing: An Overview, Data Integration, Data Reduction, Data Transformation and Data Discretization.

Data Warehouse and OLAP Technology for Data Mining: Data Warehouse basic concepts, Data warehouse modeling, Data warehouse design and usage, Implementation, Generalization by Attribute oriented Induction.15 Hours

UNIT - III

Mining Frequent patterns: Associations and Correlations: Basic Concepts, Frequent Item set Mining Methods, Interesting Patterns and Pattern evaluation methods. 8 Hours

UNIT - IV

Classification:Basic Concepts, Decision Tree Induction, Bayesian Classification, Rule based Classification, Model Evaluation & Selection, Techniques to improve classification Accuracy. 10 Hours

UNIT - V

Cluster Analysis: Requirements, Overview, Partitioning Methods, Hierarchical Methods, Density based Methods, Grid-based Methods, Evaluation of Clustering.

Outlier Detection: Types of Outliers & Challenges in outlier detection, Outlier detection methods, Statistical Approaches, Proximity based approaches, Cluster & Classification based approach. 15 Hours

Text Books:

1. Data Mining Concepts and Techniques by Jiawei Han, Michelin Kamber, and Jian Pei, Elsevier Publications,3rd Edition, 2013.

Reference Books:

1. Introduction to Data Mining by Pang- Ning Tan, Michael Steinbach and Vipin Kumar, Pearson Education, Low Price Edition.

2. Insight to Data Mining Theory and Practice by K.P.Soman, ShyamDiwakar and V. Ajay, Prentice Hall of India, 2006.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand the basic concepts of data mining, including the types of data, patterns, and applications. *(L2: Understand)*
- **CO2**: Apply data preprocessing techniques such as data integration, reduction, and transformation to prepare data for mining. *(L3: Apply)*
- **CO3**: Analyze frequent pattern mining methods and evaluate the interestingness of discovered patterns. *(L4: Analyze)*
- **CO4**: Implement classification techniques such as decision trees, Bayesian classification, and rule-based methods to solve classification problems. *(L5: Evaluate)*
- **CO5**: Apply clustering methods and outlier detection techniques to group data and identify anomalies. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	3
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

III SEMESTER

GENERIC ELECTIVE I

20SCA 841:ARTIFICIAL INTELLIGENCE

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Artificial Intelligence" course provides students with a foundational understanding of the core concepts, techniques, and applications of artificial intelligence. The course covers problem-solving techniques, knowledge representation, reasoning under uncertainty, expert systems, and natural language processing. Students will gain the necessary skills to develop AI systems that address real-world challenges through the application of search algorithms, logic programming, and AI frameworks.

Course Objectives:

- To introduce the fundamental concepts of Artificial Intelligence, including problem-solving techniques and search algorithms.
- To provide students with an understanding of knowledge representation methods and logic programming.
- To teach reasoning under uncertainty using probabilistic models, including Bayesian networks and fuzzy logic.
- To explain the structure and functionality of expert systems, their types, and their knowledge acquisition techniques.
- To familiarize students with the role of knowledge in natural language processing and the approaches to language understanding.

UNIT-1

Introduction to Artificial Intelligence: Artificial Intelligence, AI Problems, AI Techniques, The Level of the Model, Criteria For Success. Defining the Problem as a State Space Search, Problem Characteristics, Production Systems, Production System Characteristics, search Issues in the Design of Search Programs, Un-Informed Search, BFS, DFS; Heuristic.

Search Techniques: Generate-And- Test, Hill Climbing, Best-First Search, A * Algorithm, Problem Reduction, AO*Algorithm, Constraint Satisfaction, Means-Ends Analysis. 12 Hours

UNIT-II

Knowledge Representation: Procedural Vs Declarative Knowledge, Representations and Mappings, Approaches to Knowledge Representation, Issues in Knowledge Representation, Logic Programming Forward Vs Backward Reasoning,

Symbolic Logic: Propositional Logic, First Order Predicate Logic: Representing Instance and is a Relationships, Computable Functions and Predicates, Syntax & Semantics of FOPL, Normal Forms, Unification &Resolution, Representation Using Rules, and Natural Deduction. 12 Hours

UNIT-III

Reasoning under Uncertainty: Introduction to Non-Monotonic Reasoning, Truth Maintenance Systems, Statistical Reasoning: Bayes Theorem, Certainty Factors & Rule-Based Systems, Bayesian Probabilistic Inference, Bayesian Networks, Dempster-Shafer Theory, Fuzzy Logic & Fuzzy Systems. 10 Hours

UNIT-IV

Experts Systems: Overview of an Expert System, Structure of an Expert Systems, Different Types of Expert Systems- Rule Based, Model Based, Case Based and Hybrid Expert Systems, Knowledge Acquisition and Validation Techniques, Black Board Architecture, Knowledge Building System Tools, Expert System Shells. 10 Hours

UNIT-V

Natural Language Processing: Role of Knowledge in Language Understanding, Approaches Natural Language Understanding, Steps in The Natural Language Processing, Syntactic Processing and Augmented Transition Nets, Semantic Analysis

Hours

Text Book:

1. Artificial Intelligence by Elaine Rich, McGraw-Hill Publications,3rd edition,2017

References:

1. Artificial Intelligence : A Modern Approach by Stuart Russell, Peter Norwig Pearson Education Publications, 3rdedition , 2018.
2. Artificial Intelligence by George F Luger, Pearson Education Publications, 5th edition, 2008.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand the fundamental concepts of Artificial Intelligence, problem-solving techniques, and search algorithms such as BFS, DFS, and A*. *(L2: Understand)*
- **CO2**: Apply knowledge representation techniques, including propositional logic and first-order predicate logic, for problem-solving. *(L3: Apply)*
- **CO3**: Analyze reasoning techniques under uncertainty, such as Bayesian networks, fuzzy logic, and Dempster-Shafer theory. *(L4: Analyze)*
- **CO4**: Evaluate the structure and types of expert systems, including rule-based, model-based, and hybrid systems, for solving complex problems. *(L5: Evaluate)*
- **CO5**: Apply natural language processing techniques for syntactic and semantic analysis of language. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	3
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

III SEMESTER

20SCA 843: CLOUD COMPUTING

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Cloud Computing" course offers a comprehensive understanding of cloud-based technologies and their applications in modern computing environments. It covers key concepts such as cloud computing architecture, virtualization, cloud infrastructure models (IaaS, PaaS, SaaS), and practical exposure to AWS services. By the end of this course, students will have the knowledge and skills required to design, implement, and manage cloud-based applications and infrastructure effectively.

Course Objectives:

- To introduce the fundamental concepts of cloud computing, its vision, and the benefits and challenges it brings to modern computing.
- To provide students with an understanding of virtualization technologies and their role in cloud computing.
- To familiarize students with the architecture and infrastructure of cloud computing, including public, private, hybrid, and community clouds.
- To offer hands-on experience with AWS services, including EC2, Elastic Block Store (EBS), and Elastic Beanstalk, for developing cloud-based applications.
- To enable students to perform basic cloud development tasks, such as configuring cloud services, managing security, and monitoring applications using AWS tools.

UNIT – I

Introduction: Cloud Computing at a Glance, The Vision of Cloud Computing, defining a Cloud, A Closer Look, Cloud Computing Reference Model, Characteristics and Benefits, Challenges Ahead, Historical Developments, Distributed Systems, Virtualization, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing, Building Cloud Computing Environments, Application Development, Infrastructure and System Development, Computing Platforms a Technologies, Amazon Web Services (AWS), Google App Engine, Microsoft Azure, Hadoop, Force.com and Salesforce.com, Manjra soft Aneka. 8 Hours

UNIT - II

Virtualization: Introduction, Characteristics of Virtualized Environments, Taxonomy of Virtualization Techniques, Execution Virtualization, Other Types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples, Xen: Paravirtualization, VMware: Full Virtualization, Microsoft Hyper-V. 8 Hours

UNIT – III

Cloud Computing Architecture: Introduction Cloud Reference Model, Architecture Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds, Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges, Cloud Dentition, Cloud Interoperability and Standards, lability and Fault Tolerance, Security, Trust, and Privacy, Organizational Aspect. 8 Hours

UNIT - IV

Discovering the AWS Development Environment: Starting Your AWS Adventure, Defining the AWS Cloud, Discovering IaaS, Determining Why You Should Use AWS, Considering the AWS-Supported Platforms. Obtaining Development Access to Amazon Web Services: Discovering the Limits of Free Services, Considering the Hardware Requirements, Getting Signed Up, Testing Your Setup, Choosing the Right Services, Getting a Quick Overview of Free-Tier Services, Matching AWS Services to Your Application, Considering AWS Security Issues. 10 Hours

UNIT - V

Starting the Development Process: Considering AWS Communication Strategies, Defining the Major Communication Standards, Understanding How REST Works, Creating a Development Environment, Choosing a Platform, Obtaining and Installing Python, Working with the Identity and Access Management Console, Installing the Command Line Interface Software, Configuring Using CLI, Configuring Using Node.js, Configuring Using a Desktop Application, Creating a Virtual Server Using EC2, Getting to Know the Elastic Compute Cloud (EC2), Working with Elastic Block Store (EBS) Volumes, Discovering Images and Instances.

Performing Basic Development Tasks : Understanding AWS Input/Output, Considering the Input /Output Options, Working with JSON, Working with XML, Working with Amazon API Gateway, Developing Web Apps Using Elastic Beanstalk, Considering Elastic Beanstalk (EB) Features, Deploying an EB Application, Updating an EB Application, Removing Unneeded Applications, Monitoring Your Application Using Amazon CloudWatch. 12 Hours

Text Books:

1. Mastering Cloud Computing by Rajkumar Buyya, Christian Vecchiola, S ThamaraiSelvi, Tata McGraw Hill Education Private Limited, 2013.
2. AWS for Developers- Dummies by John Paul Mueller, John Wiley & Sons Inc.publications, 2017.

Reference Books:

1. Cloud Computing Concepts Technology Architecture by Thomas Erl, Pearson Education, 2014.
2. Cloud Computing Explained by John Rhoton , Recursive Press, 2009.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand the fundamental concepts, benefits, and challenges of cloud computing and its historical developments. *(L2: Understand)*
- **CO2**: Analyze various virtualization techniques and their role in enabling cloud environments. *(L4: Analyze)*
- **CO3**: Understand the cloud computing architecture, including different types of clouds and cloud service models (IaaS, PaaS, SaaS). *(L3: Apply)*
- **CO4**: Develop and configure cloud-based applications using AWS services like EC2 and Elastic Beanstalk. *(L5: Evaluate)*
- **CO5**: Perform basic cloud development tasks, such as configuring security, managing I/O operations, and monitoring applications using AWS tools. *(L6: Create)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	3
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

III SEMESTER

20SCA 845: NETWORK SECURITY

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Network Security" course provides an in-depth understanding of the principles, mechanisms, and technologies used to secure modern networks and information systems. It covers the fundamental concepts of computer and network security, including cryptography, secure communication protocols, wireless network security, and protection against malicious software. By the end of the course, students will be equipped with the knowledge to analyze and implement various security mechanisms that protect data and networks from potential threats.

Course Objectives:

- To introduce the basic concepts of network security, including security architecture, attacks, services, and mechanisms.
- To familiarize students with symmetric and asymmetric encryption methods, and their use in ensuring message confidentiality and authentication.
- To provide an understanding of transport-level and wireless network security protocols, such as SSL, TLS, and IEEE 802.11.
- To explain the principles of electronic mail and IP security, including the use of PGP, S/MIME, and IPsec.
- To educate students on different types of malicious software, their propagation methods, and defense mechanisms against attacks like viruses, worms, and DDoS.

UNIT- I

Introduction: Computer Security Concepts, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms, A Model for Network Security, Standards. Symmetric Encryption and Message Confidentiality: Symmetric Encryption Principles, Symmetric Block Encryption Algorithms, DES, AES, Stream. 10 Hours

UNIT – II

Public-Key Cryptography and Message Authentication: Approaches to Message Authentication, Secure Hash Functions, Message Authentication Codes, Public Key Cryptography Principle, PublicKey Cryptography Algorithms, RSA, Diffie Hellman key Exchange, Digital Signatures. Key Distribution and User Authentication: Symmetric Key Distribution Using Symmetric Encryption, Kerberos, Key Distribution Using Asymmetric Encryption. 10 Hours

UNIT - III

Transport-Level Security: Web Security Issues, Secure Sockets Layer (SSL), Transport Layer Security (TLS), HTTPS, Secure Shell (SSH).

Wireless Network Security: Wireless Security, Mobile Device Security, IEEE 802.11 Wireless LAN Overview, IEEE 802.11i Wireless LAN Security. 10 Hours

UNIT –IV

Electronic Mail Security: Pretty Good Privacy (PGP), S/MIME.IP Security: IP Security Overview, IP Security Policy, Encapsulating Security Payload, Combining Security Associations, Internet Key Exchange and Cryptographic Suites.

UNIT - V

Malicious Software: Types of Malicious Software, Propagation, Infected Content, Viruses Propagation, Vulnerability Exploit, Worms, Propagation, Social Engineering, SPAM, Trojans, Payload-System Corruption, Payload, Attack Agent, Zombie, Bots, Payload, Information Theft,

Keyloggers, Phishing, Spyware, Payload, Stealthing, Backdoors, Rootkits Distributed Denial of Service Attacks. 10 Hours

Text Books:

1. Network Security Essentials, Applications and Standards, 5th edition, William Stallings, Pearson Education, 2013.

Reference Books:

1. Cryptography & Network Security, Behrouz A. Forouzan, Tata McGraw-Hill, New Delhi, 2007.
2. Network Security: Private Communication in a Public World, Kaufman, Pearson Education Asia,New Delhi, 2002.
3. Cryptography and Network Security: AtulKahate, Tata McGraw-Hill Publishing Company Limited, New Delhi, 2003.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Understand the fundamental concepts of computer security, security attacks, services, and mechanisms. *(L2: Understand)*
- **CO2**: Apply symmetric and public-key encryption techniques to ensure message confidentiality and secure key distribution. *(L3: Apply)*
- **CO3**: Implement transport-level security protocols such as SSL, TLS, and SSH, and understand wireless network security principles. *(L4: Analyze)*
- **CO4**: Analyze and implement electronic mail security solutions and IP security mechanisms for secure data transmission. *(L5: Evaluate)*
- **CO5**: Identify and counteract different types of malicious software, including viruses, worms, and Distributed Denial of Service (DDoS) attacks. *(L5: Evaluate)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	3
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	3	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

III SEMESTER

20SSE 801: DIGITAL MARKETING

Hours per week: 4 Examination: 100 Marks

Credits: 3

Preamble:

The "Digital Marketing" course provides an in-depth understanding of the evolving digital marketing landscape, focusing on the strategies and tools used to engage customers in the digital era. Students will explore various digital channels, learn how to create digital marketing plans, and analyze the performance of digital media to optimize marketing campaigns.

Course Objectives:

- To introduce the components and strategies of digital marketing and its comparison with traditional marketing.
- To provide an understanding of the various channels of digital marketing, including website marketing, social media marketing, and online advertising.
- To teach the importance of audience segmentation and customer acquisition in the digital marketing context.
- To guide students in creating and implementing a comprehensive digital marketing plan.
- To explore search engine marketing (SEM), social media marketing (SMM), and techniques for measuring the performance of digital media.

Understanding Digital Marketing: Components of Digital Marketing, Need and Scope of Digital Marketing, Benefits of Digital Marketing, Digital Marketing Platforms and Strategies, Comparison of Marketing and Digital Marketing, Digital Marketing Trends.

Channels of Digital Marketing: Digital Marketing, Website Marketing, Search Engine Marketing, Online Advertising, Email Marketing, Blog Marketing, Social Media Marketing, Audio, Video and Interactive Marketing, Online Public Relations, Mobile Marketing, Migrating from Traditional Channels to Digital Channels.

Marketing in the Digital Era: Segmentation – Importance of Audience Segmentation, How different segments use Digital Media – Organizational Characteristics, Purchasing Characteristics, Using Digital Media to Reach, Acquisition and Retention of new customers, Digital Media for Customer Loyalty.

Digital Marketing Plan: Need of a Digital Marketing Plan, Elements of a Digital Marketing Plan – Marketing Plan, Executive Summary, Mission, Situational Analysis, Opportunities and Issues, Goals and Objectives, Marketing Strategy, Action Plan, Budget, Writing the Marketing Plan and Implementing the Plan. .

Search Engine Marketing and Online Advertising: Importance of SEM, understanding Web Search – keywords, HTML tags, Inbound Links, Online Advertising vs. Traditional Advertising, Payment Methods of Online Advertising – CPM (Cost-per-Thousand) and CPC (Cost-per-click), Display Ads - choosing a Display Ad Format, Landing Page and its importance.

Social Media Marketing: Understanding Social Media, Social Networking with Facebook, LinkedIn, Blogging as a social medium, Microblogging with Twitter, Social Sharing with YouTube, Social Media for Customer Reach, Acquisition and Retention. Measurement of Digital Media: Analyzing Digital Media Performance, Analyzing Website Performance, Analyzing Advertising Performance.

Text Books:

1. B2B Digital Marketing by Michael Miller, 1e, Pearson, 2014.

2. Digital marketing by Vandana Ahuja,Oxford University Press 2015
3. Social Media Marketing by Michael R Solomon, Tracy Tuten, Pearson, 1e, 2015.
4. E-Marketing by Judy Strauss & Raymond Frost, Pearson, 2016
5. Online marketing – A customer led approach by Richard Gay, Alan Charles worth, Rita Esen,

Course Outcomes:

At the end of the course, the student is able to:

- **CO1**: Identify the applications of digital marketing in the globalized market. *(L3: Apply)*
- **CO2**: Identify the various channels of digital marketing. *(L3: Apply)*
- **CO3**: Discuss and create a comprehensive digital marketing plan. *(L6: Create)*
- **CO4**: Explain the concepts and strategies of search engine marketing. *(L5: Evaluate)*
- **CO5**: Conclude and assess the effectiveness of online advertising techniques. *(L5: Evaluate)*

CO-PO Mapping table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	3	2	2	2	1	2	1
CO2	2	3	2	2	2	1	2	2
CO3	1	2	1	2	3	1	1	2
CO4	2	3	2	2	1	2	2	2
CO5	2	3	2	2	1	3	1	2

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

III SEMESTER

SSE 803:MANAGEMENT INFORMATION SYSTEMS

Hours per week: 4 Examination: 100 Marks

Credits: 3

Preamble:

The "Management Information Systems" course provides students with an understanding of the role of MIS in the digital age and how it supports business processes, decision-making, and strategic management. The course also covers the development and management of MIS, as well as its integration into global enterprises, including ERP, SCM, and CRM systems.

Course Objectives:

- To introduce the role and impact of Management Information Systems (MIS) in the digital era.
- To explain the components of e-business enterprises and their applications in real-time communication, collaboration, and commerce.
- To provide insights into strategic management and the use of MIS for improving business performance.
- To teach decision-making concepts and how MIS supports decision-making processes in organizations.
- To familiarize students with the development, implementation, and management of enterprise systems such as ERP, SCM, and CRM.

Management Information System in Digital form: Definition, Role, impact, MIS and User, Management as a Control System, Management Effectiveness and MIS, Organization as a system.

E-Business Enterprise: Introduction, E-business, E-Commerce, E- Communicaiton, E- Collaboration, Real Time Enterprise.

Strategic Management of Business Performance: Essentiality of Strategic Planning, Tools of Planning, Strategic Management, Three approaches to development of Strategy, Classes and Types of Strategy.

Decision Making: Decision Making Concepts, Process, Decision Analysis by Analytical Modeling, Behavioural Concepts in Decision Making, Organizational Decision Making, MIS and Decision Making .

Development Process of MIS: Development of Long Range Plans of the MIS, Determining the Information Requirement, Development and Implementation of the MIS, Management of Information Quality in the MIS, Organization for Development of MIS, Development Process Model.

Management of Global Enterprise: Enterprise Management System, Enterprise Resource Planning System, ERP Model and Modules, Benefits of the ERP, ERP product Evaluation, ERP Implementation, Supply Chain Management , Information Management in SCM, Customer Relationship Management, EMS and MIS.

Text Books:

1. Management Information Systems: A Global Digital Enterprise Perspective, Waman S Jawadekar, Fifth Edition, TMH,2017.

2. Management Information System- Kenneth C. Laudon, Jane P Laudon, Pearson, 14th Edition, 2016.

3. Information Systems for Modern Management, Murdick, Robert G, PHI, 3rd Edition.\

Course Outcomes:

At the end of the course, the student is able to:

- **CO1**: Explain the fundamental concepts of Management Information Systems (MIS) and their role in enhancing management effectiveness and organizational control. *(L2: Understand)*
- **CO2**: Analyze the components and benefits of an e-business enterprise, including e-commerce, e-

communication, and real-time collaboration, and their impact on modern business practices. *(L4: Analyze)*

- **CO3**: Evaluate strategic management tools and approaches to improve business performance and formulate strategies for organizational growth. *(L5: Evaluate)*
- **CO4**: Apply decision-making models and analytical techniques to solve complex business problems and enhance organizational decision-making processes. *(L3: Apply)*
- **CO5**: Develop a comprehensive understanding of enterprise management systems, including ERP, SCM, and CRM, and evaluate their implementation and impact on global business operations. *(L6: Create, L5: Evaluate)*

CO-PO Mapping table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	1	1	1	2
CO2	3	3	3	3	2	2	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

III SEMESTER

20SCA 821: OBJECT ORIENTED PROGRAMMING WITH JAVA LAB

Hours per week: 3 Examination: 100 Marks

Credits: 2

Preamble:

The "Object Oriented Programming with Java Lab" course is designed to provide students with hands-on experience in applying the principles of object-oriented programming (OOP) using Java. The course covers the essential concepts of OOP, including inheritance, interfaces, threads, file handling, and graphical user interface (GUI) components such as applets and Swing. Through practical implementation, students will learn to develop Java applications that incorporate real-world problem-solving techniques, preparing them for professional software development.

Course Objectives:

- To introduce students to the basic syntax and structure of Java programming through practical exercises.
- To enable students to implement key object-oriented principles such as inheritance, polymorphism, and interfaces.
- To teach students how to handle exceptions, manage input/output operations, and work with files in Java.
- To provide hands-on experience with multithreading, concurrency, and GUI-based applications using Java Swing and AWT.
- To equip students with the ability to develop Java applications that use event handling, collections, and Java packages.

Objective: The aim of this lab is to make the students learn the basic concepts of Java programming. This course covers preliminaries and makes the students learn how to program in java using Basic Concepts, Inheritance, Interfaces, Packages, Threads, I/Os, Applets, Swings, Event Handling, Collections and allow the students to implement effectively.

1. To find the average and sum of the N numbers Using Command line argument.
2. To Demonstrate Type Casting.
3. To find the number of arguments to provide at runtime.
4. To Test the Prime number.
5. To calculate the Simple Interest and Input by the user.
6. To create a simple class to find out the Area and Perimeter of rectangle and box using super and this keyword.
7. To design a class account using the inheritance and static that show all function of bank (withdrawal,deposite).
8. To design a class using abstract methods and classes.
9. To create a package that access the member of external class as well as same package.
10. Import the user define package and access the Member variable of classes that contained by Package.
11. To show the partial implementation of Interface.
12. To create a thread that implement the Runnable interface.
13. To create a file and write data into it using the methods OutputStream class
14. To accept specified number of characters as input and converts them into uppercase characters
15. To illustrate creation of threads using runnable class.(start method start each of the newly created thread. Inside the run method there is sleep() for suspend the thread for 500 milliseconds).
16. To create a class MyThread in this class a constructor, call the base class

constructor, using super and starts the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently

17. To get the reference to the current thread by calling currentThread() method
18. Write a program for example of try and catch block. In this check whether the given array size is negative or not.
19. Write a program for example of multiple catch statements occurring in a program.
20. Write a program to describe usage of throws clause.
21. To create a new array list, add some colors (string) and print out the collection
22. To iterate through all elements in a linked list
23. To append the specified element to the end of a hash set.
24. To demonstrate Banner Applet.
25. To demonstrate different control on applets.
26. To demonstrate the Mouse Event Handlers.
27. To create an AWT based application
28. Implement the flow layout And Border Layout.
29. Implement the GridLayout,CardLayout
30. Handle an event in a Swing program
31. Perform a simple Swing-based applet
32. Demonstrate an icon-based JButton

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Students will be able to write Java programs using fundamental concepts such as command-line arguments, type casting, and input/output operations. *(L3: Apply)*
- **CO2**: Students will develop object-oriented programs that demonstrate the use of inheritance, interfaces, and abstract classes. *(L4: Analyze)*
- **CO3**: Students will implement multithreading in Java by creating threads and handling concurrency issues. *(L5: Evaluate)*
- **CO4**: Students will be able to handle exceptions using try-catch blocks and implement file operations using Java I/O streams. *(L3: Apply)*
- **CO5**: Students will create graphical user interfaces using Java AWT and Swing components, and handle events in Java-based applications. *(L6: Create)*

Text Books:

1. Java The complete reference by Herbert Schildt, McGraw Hill Education Pvt. Ltd, 9th edition.
2. Understanding Object-Oriented Programming with Java by T. Budd, Pearson Education.

CO-PO Mapping Table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	3	1	1	1	2
CO2	3	3	3	3	2	1	2	3
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	2
CO5	3	3	3	3	2	2	2	2

Note: 1 - Low Correlation, 2 - Medium Correlation, 3 - High Correlation

III SEMESTER

20SCA 824: DATA ANALYSIS USING R LAB

Hours per week: 3 Examination: 100 Marks

Credits: 2

Preamble:

The "Data Analysis Using R Lab" course is designed to provide students with hands-on experience in R programming, enabling them to work effectively with data analysis, manipulation, and visualization. The course covers a range of R programming techniques, from basic syntax to advanced data manipulation, text processing, and visualization using packages like ggplot2 and grid. Students will gain practical knowledge in handling data, working with large datasets, and building dynamic data visualizations, preparing them for real-world data analysis challenges.

Course Objectives:

- To introduce students to the R programming environment and its basic syntax, enabling them to perform data analysis tasks.
- To teach students advanced data manipulation techniques, including summarizing, filtering, and merging data in R.
- To equip students with the skills to process text data, work with regular expressions, and optimize physical memory usage in R.
- To develop students' abilities to create and customize data visualizations using the ggplot2 package.
- To familiarize students with advanced visualization techniques such as mapping, interactive plots, and the use of the grid package for enhanced visualizations.

Objective: The aim of this course is to have gain knowledge in latest and advanced features of R programming. It explains in detail how to perform various data analysis functions using R Programming.

1. The R Programming Environment: Basic R Language, Basics of R, including syntax, some tidy data principles and processes, and how to read data into R.

2. Data Manipulation: Summarize, filter, merge, and otherwise manipulate data in R, including working through the challenges of dates and times.

3. Text Processing, Regular Expression, & Physical Memory: Using R tools and packages to deal with text and regular expressions. Learn how to manage and get the most from your computer's physical memory when working in R.

Building Data Visualization Tools:

1. Welcome to Building Data Visualization Tools:
2. Plotting with ggplot2: creating and customizing ggplot2 plots.
3. Mapping and interactive plots:Creating simple and dynamic maps with ggplot2 and ggmap, how to overlay data, and how to create chloropleth maps of US counties.
4. The grid Package: The grid package in R implements the primitive graphical functions that underlay the ggplot2 plotting system.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Students will be able to understand and navigate the R programming environment, including basic syntax and data manipulation techniques. *(L2: Understand)*
- **CO2**: Students will demonstrate the ability to manipulate and process data using R, including summarizing, filtering, and handling dates and times. *(L3: Apply)*
- **CO3**: Students will be able to work with text data and apply regular expressions, as well as manage physical memory when working with large datasets in R. *(L4: Analyze)*
- **CO4**: Students will develop skills in creating and customizing data visualizations using the ggplot2 package in R. *(L5: Evaluate)*
- **CO5**: Students will apply advanced visualization techniques, such as creating interactive plots, mapping, and using the grid package to enhance data visualizations. *(L6: Create)*

CO-PO Mapping Table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	3	2	1	1	2
CO2	3	3	3	3	2	1	2	2
CO3	3	3	3	3	2	2	2	2
CO4	3	3	3	3	2	2	2	2
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation, 2 - Medium Correlation, 3 - High Correlation

MCA IV SEMESTER

20SCA 842 : MACHINE LEARNING

Hours per week: 4 Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Machine Learning" course is designed to introduce students to the fundamental concepts and methodologies of machine learning, covering both theoretical and practical aspects. The course encompasses a wide range of machine learning techniques including supervised, unsupervised, and neural networks, as well as advanced topics such as ensemble learning and regularization. Through hands-on practice, students will develop skills in building, evaluating, and improving machine learning models using real-world datasets. This course prepares students to tackle modern data-driven challenges in various fields such as AI, data science, and robotics.

Course Objectives:

- To provide a foundational understanding of machine learning, its applications, and the tools used in the field.
- To explore different machine learning models, their selection, training, and evaluation techniques.
- To introduce supervised learning algorithms, including KNN, decision trees, support vector machines, and regression models.
- To familiarize students with unsupervised learning methods such as clustering and association rules, along with neural networks.
- To introduce advanced machine learning topics such as representation learning, active learning, ensemble learning, and regularization.

UNIT - I

Introduction: What is human Learning , Types of Human Learning, Problems not to be solved using machine Learning, Applications of Machine Learning, Tools in Machine Learning, Issues in Machine Learning.

Preparing to Model: Machine Learning Activities, Basis types of Data in Machine Learning, Exploring structured data, Data Quality and Remediation, Data Preprocessing. 10 Hours

UNIT-II

Modelling & Evaluation: Introduction, selecting a model, training a model, Model representation and Interpretability, Evaluating Model performance, Improving model performance.

Feature Engineering: Introduction, Feature Transformation, Subset selection. 8 Hours

UNIT-III

Introduction, Importance of Bayes Theorem, Bayes theorem and concept learning, Bayesian Belief Network.

Supervised Learning: KNN, Decision Tree, Random forest model, Support vector Machines, Regression. 10 Hours

UNIT-IV

Unsupervised Learning: Supervised Vs Unsupervised Learning, Applications of Unsupervised Learning, Clustering, Association Rule.

Basic Neural Networks: Neural Network, Understanding Biological Neuron, Exploring Artificial Neuron, Types of activation function, Early implementation of ANN, Architecture of Neural Networks, Learning process in Artificial Neural Networks, Back Propagation, Deep Learning.12 Hours

UNIT-V

Other Types of Learning: Introduction, Representation Learning, Active Learning, Instance Based learning, Association Rule Learning, Ensemble Learning Algorithm, Regularization algorithm. 8 Hours

Text Book:

1. Machine Learning by Subramanian, Chandra Mouli, Amit Kumar Das , SaikantDutt, Pearson Publications, I edition, 2018.

Reference Book:

1. Machine Learning by Tom Mitchell, McGraw Hill, 2007

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Students will be able to understand the basics of machine learning, data preprocessing, and the challenges associated with real-world data. *(L2: Understand)*
- **CO2**: Students will demonstrate proficiency in selecting, training, and evaluating machine learning models. *(L3: Apply)*
- **CO3**: Students will apply supervised learning techniques to solve classification and regression problems. *(L3: Apply)*
- **CO4**: Students will understand and implement unsupervised learning algorithms and basic neural networks for clustering and pattern recognition. *(L4: Analyze)*
- **CO5**: Students will gain knowledge of advanced machine learning techniques such as ensemble learning, regularization, and representation learning. *(L5: Evaluate)*

CO-PO Mapping Table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	2	1	1	2
CO2	3	3	3	3	2	1	2	2
CO3	3	3	3	3	2	2	2	2
CO4	2	3	3	3	2	2	2	3
CO5	3	2	3	3	2	2	2	3

Note: 1 - Low Correlation, 2 - Medium Correlation, 3 - High Correlation

MCA IV SEMESTER

20SCA 844 : BLOCK CHAIN TECHNOLOGIES

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Blockchain Technologies" course provides a comprehensive introduction to the fundamental concepts of blockchain and its applications. It covers the history, architecture, and evolution of blockchain from cryptocurrencies to more advanced versions like Blockchain 2.0. The course dives into cryptographic techniques, distributed consensus mechanisms, smart contracts, and decentralized applications. Students will also explore various real-world use cases such as blockchain for voting and managing land registry records, along with cutting-edge research topics in the field of blockchain.

Course Objectives:

- To introduce the fundamental concepts of blockchain technology and its evolution from cryptocurrency to smart contracts.
- To provide an understanding of cryptographic techniques used in blockchain, such as hash functions, digital signatures, and public key cryptography.
- To explain the mechanisms of distributed consensus, including proof of work, proof of stake, and mining in the context of blockchain networks.
- To introduce smart contracts, Solidity programming, and decentralized applications (DApps) for blockchain-based solutions.
- To explore real-world applications of blockchain in various sectors and discuss the research challenges and future advancements in blockchain technology.

UNIT – I

Introduction: What is Blockchain (BC), public ledgers, BC as public ledgers; BC history - Bitcoin and Cryptocurrency, BC 2.0, Smart contracts; BC architecture – Blocks in BC, transactions and distributed consensus; BC conceptualization - The Chain and the Longest Chain, Cryptocurrency to Blockchain 2.0, Permissioned Model of Blockchain.12 Hours

UNIT -II

Cryptographic Primitives: Cryptographic Hash Function, Properties of a hash function, Hash pointer and Merkle tree, Digital Signature, Public Key Cryptography, A basic cryptocurrency.10 Hours

UNIT - III

Distributed consensus : Distributed consensus in open environments, Consensus in a Bitcoin network; Bitcoin Consensus - Proof of Work (PoW) – basic introduction, HashcashPoW, Beyond Consensus in Bitcoin - BitcoinPoW, Attacks on PoW and the monopoly problem, Proof of Stake, Proof of Burn and Proof of Elapsed Time; Consensus in Bitcoin (The Miners) - The life of a Bitcoin Miner, Mining Difficulty, Mining Pool.12 Hours

UNIT - IV

Smart contracts - I : Smart contracts, Solidity, REMIX IDE, EthereumBlockchain, Ethereum Virtual Machine.

Smart contracts –II : Decentralized applications (Dapps), Truffle development, Design improvements, Application models and standards. 12 Hours

UNIT - V

Use Cases: Blockchain for Voting, Government Use-cases – Public distribution system, Blockchain for Tax Payments, Blockchain for Managing Land Registry Records, Other Block Chain Frame Works: IBM Hyperledge fabric

Research Aspets in Block Chain: Consensus protocols, Identity management, Strong and weak synchronization, avoiding forks, Mining improvements. 10 Hours

Text Books:

1. Drescher, Daniel. "Blockchain basics",A Non-Technical Introduction in 25 Steps Apress, 2017.
2. Mougayar, William. "The business blockchain: promise, practice, and application of the next Internet technology", John Wiley & Sons, 2016.
3. Dannen, Chris. "Introducing Ethereum and Solidity", Berkeley: Apress, 2017.
4. Prusty, Narayan. "Building Blockchain Projects", Packt Publishing Ltd, 2017.
5. Pilkington, Marc. "Blockchain technology: principles and applications" Research handbook on digital transformations, 2016.
6. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.
7. Swan, Melanie, "Blockchain: Blueprint for a new economy", O'Reilly Media, Inc., 2015.
8. Antonopoulos, Andreas M. "Mastering Bitcoin: unlocking digital crypto currencies", O'Reilly Media, Inc., 2014.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Students will be able to explain the architecture and functioning of blockchain technology and its use in cryptocurrencies. *(L2: Understand)*
- **CO2**: Students will demonstrate knowledge of cryptographic primitives and their role in securing blockchain networks. *(L3: Apply)*
- **CO3**: Students will understand and apply consensus mechanisms such as proof of work and proof of stake in blockchain systems. *(L4: Analyze, L3: Apply)*
- **CO4**: Students will develop and deploy smart contracts and decentralized applications (DApps) using Solidity and related blockchain tools. *(L6: Create)*
- **CO5**: Students will evaluate various use cases of blockchain technology and understand the research challenges in advancing blockchain protocols. *(L5: Evaluate)*

CO-PO Mapping Table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	2	1	3	2
CO2	3	3	2	3	2	1	2	2
CO3	3	3	3	3	2	2	2	2
CO4	2	3	3	3	2	2	2	3
CO5	3	2	3	3	2	2	2	3

Note: 1 - Low Correlation, 2 - Medium Correlation, 3 - High Correlation

MCA IV SEMESTER

20SCA 846: CYBER SECURITY

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Cyber Security" course is designed to provide students with a comprehensive understanding of the evolving landscape of cyber threats, the various forms of cybercrimes, and the tools and techniques used by attackers. This course covers the fundamentals of cyber security, including cyber-crime, cyber offenses, mobile device security, cyber laws, and forensics. By exploring both technical and legal aspects, students will gain the knowledge and skills needed to secure systems and networks, as well as understand the legal implications of cybercrimes and digital forensics.

Course Objectives:

- To introduce the concept of cybercrime, its evolution, and the legal perspectives in India and globally.
- To explore the security challenges posed by mobile devices and wireless technologies in the modern digital era.
- To familiarize students with the tools, methods, and techniques used in cybercrime, including malware, phishing, and attacks on networks.
- To provide an understanding of cyber laws, focusing on the Indian IT Act and its implications for digital security.
- To introduce the principles of digital forensics and the process of forensic investigation in cybercrime.

UNIT-I

Introduction to cyber-crime: Definition & Origin of the Word, Cyber Crime and Information Security, Who are Cyber Criminals, Classification of cyber Crimes, Cyber Crime – Legal and Indian perspective, Cyber Crime and Indian ITA 2000, Global Perspective on cyber-crimes.

Cyber offences: Introduction, How criminals plan the attacks, Social engineering, Cyber talking, Cyber Café and Cyber Crimes.

Botnets: The fuel for Cyber Crimes, Attack Vector, Cloud Computing. 10 Hours

UNIT-II

Cybercrime-Mobile and Wireless Devices: Proliferation of mobile and wireless devices, Trends in mobility, Credit card Frauds in Mobile and Wireless computing Era, Security Challenges posted by Mobile Devices, Registry settings for mobile devices, Authentication Service Security, Attacks on mobile/cell phones

Mobile Devices: Security Implications for Organizations, Devices – Related Security Issues, Organizational Security Policies & Measures in mobile computing era, Laptops.10 Hours

UNIT-III

Tools and Methods used in Cyber Crime: Proxy servers and Anonymigers, Phishing, Password Cracking, Key Loggers, and Spywares, Virus and Worms, Trojan Horses & Backdoors Steganography, DOS & DDOS Attacks, SQL Injection, Buffer Overflow, Attacks on wireless networks.12 Hours

UNIT-IV

Cyber Crimes and Cyber Security Legal Perspectives: Cybercrime and Legal Landscape around the World, Cyber Laws Indian Context, Indian IT Act challenges to Indian Law & Cyber Crime Scenario in India Consequences of Not Addressing the Weakness in IT Act, Digital Signature &

Indian IT Act, Cyber Crime & Punishment. 12 Hours

UNIT-V

Understanding Cyber Forensics: Historical Background, Digital Forensics Science, Need For Computer Forensics, Cyber Forensics & Digital Evidence, Forensic Analysis of Email, Digital Forensics Life Cycle, Chain of Custody Concept, Network Forensics, Computer Forensics & Steganography, Relevance of OSI 7 Layer Model to Computer Forensics, Forensics & Social Networking Sites, Challenges in Computer Forensics, Special Tools & Techniques, Forensics Auditing, Anti Forensics. 12 Hours

Text Book:

- Understanding Cyber Crimes, Computer Forensics & Legal Perspective by Sunit Belapure, Nina Godbole ,Wiely India , 2011

Reference Book:

- Cyber Security Essentials by Charles J.Brooks, Christopher Grow, Philip Craig , MC Grawhill, 2018

Course Outcomes:

At the end of the course, the student is able to

- CO1: Students will be able to identify different types of cyber-crimes and understand the legal frameworks governing cyber security. *(L2: Understand)*
- CO2: Students will demonstrate knowledge of the security challenges related to mobile devices and wireless networks. *(L3: Apply)*
- CO3: Students will be able to use and understand tools for detecting cyber-attacks, including phishing, malware, and SQL injections. *(L3: Apply)*
- CO4: Students will gain knowledge of cyber laws and analyze the legal consequences of cybercrimes, particularly in the Indian context. *(L4: Analyze)*
- CO5: Students will understand the principles of digital forensics, including forensic analysis, the life cycle of a forensic investigation, and the tools used in computer forensics. *(L5: Evaluate)*

CO-PO Mapping Table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	2	1	3	2
CO2	3	3	2	3	2	1	2	2
CO3	3	3	3	3	2	2	2	2
CO4	2	2	3	2	2	3	3	2
CO5	3	2	3	3	2	2	2	3

Note: 1 - Low Correlation, 2 - Medium Correlation, 3 - High Correlation

MCA IV SEMESTER

20SCA 848: BIG DATA AND ANALYTICS

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Big Data and Analytics" course provides an in-depth understanding of the key concepts, tools, and technologies used in big data processing and analytics. It introduces students to the complexities of managing vast amounts of data and how it differs from traditional data processing methods. This course focuses on big data technologies such as Hadoop, NoSQL databases, MongoDB, Cassandra, Hive, and Pig, which are critical for managing and analyzing large datasets efficiently. The course equips students with both theoretical knowledge and practical skills necessary to tackle real-world big data challenges.

Course Objectives:

- To provide an understanding of the characteristics, evolution, and challenges of big data and its importance in business intelligence and analytics.
- To introduce NoSQL databases and compare their usage with traditional SQL for managing large-scale data.
- To familiarize students with the Hadoop ecosystem and its role in big data storage, processing, and resource management.
- To provide hands-on experience with MongoDB and Cassandra for managing unstructured data in distributed systems.
- To teach the fundamentals of Hive and Pig for querying and analyzing large datasets in a Hadoop environment.

UNIT-I

Classification of digital data, Characteristics of Data, Evolution of Big Data, Definition of Big Data, Challenges of big data, Definition of big data, other characteristics of big data, Business Intelligence Vs Big Data, Warehouse environment and Hadoop Environment, Classification of Analytics, Challenges facing big data, Importance of big data Analytics, Data Science, Terminology used in Big data environment, Tools for Analytics. 10 Hours

UNIT-II

NoSQL: Types of No SQL data bases, Advantages, Use of NoSQL, vendors of NoSQL, SQL Vs NoSQL, Comparisons of SQL, NoSQL& New SQL

Hadoop: Features, Advantages, Overview of Hadoop Ecosystem, Hadoop distribution, HadoopVs SQL, Integrated Hadoop Systems, Cloud based Hadoop Systems. 10 Hours

UNIT – III

Introduction to Hadoop: History, Overview, RDBMS Vs Hadoop, Distributed Computing Challenges, Use Case of Hadoop, Hadoop Distributors, HDFS, Processing Data with Hadoop, Managing resources and Applications with Hadoop YARN, Interacting with Hadoop Eco System. 10 Hours

UNIT – IV

Introduction to Mongo DB: What is Mongo DB, Why Mongo DB, Terms Used, Data Types in Mongo DB, Mongo DB Query Language.

Introduction to Cassandra: Apache Cassandra, Features of Cassandra CQL Data types, CQLSH, Key Spaces, CRUD, Collections, Using a counter, Time to Live, Alter Commands, Import- Export, Querying System Tables. 10 Hours

UNIT – V

Introduction to HIVE: History, Features, Workflows, Data Units, Architecture, Data types, File Format, HQL, RC File implementation, User defined function.

Introduction to Pig: Anatomy, Pig on Hadoop, Use case of Pig, Pig Latin Overview, Data types, Running Pig, Execution modes in Pig, HDFS Commands, Relational Operators, Eval Function, Complex Data Types. 10 Hours

Text Book:

1. Big data and Analytics by SemaAcharya ,SubhashiniChellappan , Wiely Publications, 2019.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Students will be able to classify and understand the characteristics of big data, its challenges, and its significance in modern analytics. *(L2: Understand)*
- **CO2**: Students will gain the ability to compare and utilize NoSQL databases for efficient management of large-scale unstructured data. *(L4: Analyze)*
- **CO3**: Students will understand and effectively use Hadoop and its ecosystem components for distributed data storage and processing. *(L3: Apply)*
- **CO4**: Students will be able to apply MongoDB and Cassandra for storing and querying data in distributed and scalable environments. *(L3: Apply)*
- **CO5**: Students will develop proficiency in using Hive and Pig for querying, analyzing, and transforming big data. *(L5: Evaluate)*

CO-PO Mapping Table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	3	2	1	1	2
CO2	3	3	3	3	2	1	1	2
CO3	3	3	3	3	2	2	2	3
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation, 2 - Medium Correlation, 3 - High Correlation

MCA IV SEMESTER

20SCA 850 : PYTHON PROGRAMMING

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Python Programming" course introduces students to the fundamental and advanced concepts of Python, a widely-used and versatile programming language. This course is designed to equip students with the necessary skills to build, debug, and manage Python programs. Starting from basic Python syntax, data structures, and functions, the course moves on to advanced topics like object-oriented programming, regular expressions, file handling, and exception management. By the end of the course, students will be able to develop robust and efficient Python applications for a variety of use cases, preparing them for real-world software development.

Course Objectives:

- To introduce the fundamental concepts of Python programming and its various data types.
- To develop problem-solving skills through the application of control flow mechanisms and functions in Python.
- To familiarize students with Python's built-in data structures such as strings, lists, dictionaries, and tuples.
- To provide an understanding of object-oriented programming (OOP) principles, inheritance, and namespaces in Python.
- To enable students to handle file operations, exceptions, and implement small scripts for data processing.

UNIT – I

Python Basics: Introduction, Data Types in Python, Mutable versus Immutable, Type Casting (also called Type Conversion) in Python, Input to a Python Program. Operators in Python: Introduction Assignment (and reassignment), Overview of Operators

Functions: Introduction, Need of Functions, Basics of Functions, defining your own functions and function syntax, Passing variables in Function Call, Function Arguments, some special functions

Flow control: Introduction, using "if", while loop, for loop, range function, Common Errors in Flow Control. 12 Hours

UNIT – II

Strings: Introduction, Creating, Initializing and Accessing Elements of a string, Traversing a String, String Operations, Difference between Functions, Methods and Attributes, String Functions versus String Methods, A Short Note on String Module.

Lists: Introduction, Some basic concepts of Lists, Creating, Traversing, and Slicing Lists, List Functions and Methods, Nested Lists and Using them as matrix.12 Hours

UNIT – III

Directories: Introduction, Basics of Dictonary-1, Basic concepts-2, Dictionary Functions and Methods, Dictionary Methods,

Tuples: Introduction, Some basic concepts regarding Tuples, some Additional Topics,

Regular Expression: Introduction, basic concepts of Expressions, Special Characters, Groups of Characters and Anchors, Understanding Re Module, Some Important methods of the Re Module.12 Hours

UNIT - IV

Object-Oriented Programming with Python: Introduction, basic concepts of Object-Oriented Programming, OOP concepts related specifically to Python, some common "Built in" Attributes and Methods of a Python Modules and classes.

Inheritance and Namespace: Introduction, Basics of Inheritance of Python, Single Inheritance, Multiple Inheritance, Concept of Namespace. 12 Hours

UNIT-V

File Operations in Python: Introduction, basics of file Operations in Python, Reading and Writing a File, Some more Advanced concepts in File Operations, some useful Methods of the OS Module, Writing small scripts for Inserting Data in a File.
Python Exceptions: Introduction, basic concepts of Exceptions in Python, User-defined Exceptions, Built-in Exceptions. 10 Hours

Text Book:

1. Python Programming by Anurag Gupta , G.P.BIswas, McGraw Hill, 2019.

Course Outcomes:

At the end of the course, the student is able to

- **CO1**: Students will be able to write Python programs using basic data types, operators, and control flow structures. *(L3: Apply)*
- **CO2**: Students will gain proficiency in string manipulation, list operations, and other data structures in Python. *(L3: Apply)*
- **CO3**: Students will understand how to use dictionaries, tuples, and regular expressions to solve complex problems efficiently. *(L4: Analyze)*
- **CO4**: Students will be able to implement object-oriented programming concepts like inheritance and namespaces in Python applications. *(L5: Evaluate)*
- **CO5**: Students will demonstrate the ability to handle file operations, manage exceptions, and write scripts for real-world data processing tasks. *(L6: Create)*

CO-PO Mapping Table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	3	1	1	1	2
CO2	3	3	2	3	1	1	1	2
CO3	3	3	3	3	2	1	1	2
CO4	2	3	3	3	2	2	2	3
CO5	3	3	3	3	2	2	2	3

Note: 1 - Low Correlation, 2 - Medium Correlation, 3 - High Correlation

MCA IV SEMESTER

20SCA 852 : ADVANCED JAVA PROGRAMMING

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

The "Advanced Java Programming" course is designed to deepen students' understanding of core and advanced Java programming concepts, focusing on building robust, scalable, and efficient enterprise-level applications. The course introduces advanced features of Java, such as JDBC for database connectivity, Servlet API for web-based applications, JSP for dynamic content generation, Hibernate for ORM solutions, and the Spring MVC framework for building Java web applications. Through hands-on practice, students will gain essential skills to build, deploy, and manage full-stack Java-based applications, preparing them for real-world software development.

Course Objectives:

- To equip students with advanced skills in Java for enterprise-level application development.
- To teach students how to implement JDBC for seamless database connectivity and management within Java applications.
- To introduce Servlet API and JSP for building dynamic and interactive web applications.
- To enable students to design and implement object-relational mapping (ORM) using Hibernate for efficient data management.
- To develop proficiency in using the Spring MVC framework for creating scalable, maintainable, and secure web applications.

.

UNIT- I

JDBC Programming: The JDBC Connectivity Model, Database Programming: Connecting to the Database, Creating a SQL Query, Getting the Results, Updating Database Data, Error Checking and the SQL Exception Class, The SQL Warning Class, The Statement Interface, Prepared Statement, Callable Statement The Result Set Interface, Updatable Result Sets, JDBC Types, Executing SQL Queries, Result Set Meta Data, Executing SQL Updates, Transaction Management. 10 Hours

UNIT – II

Servlet API and Overview: Servlet Model: Overview of Servlet, Servlet Life Cycle, HTTP Methods Structure and Deployment descriptor.

Servlet Context and Servlet Config interface, Attributes in Servlet, Request Dispatcher interface

The Filter API: Filter, Filter Chain, Filter Config,

Cookies and Session Management: Understanding state and session, Understanding Session Timeout and Session Tracking, URL Rewriting.12 Hours

UNIT - III

Java Server Pages: JSP Overview: The Problem with Servlets, Life Cycle of JSP Page,

JSP Processing, JSP Application Design with MVC, Setting Up the JSP.

Environment, JSP Directives, JSP Action, JSP Implicit Objects, JSP Form Processing, JSP Session and Cookies Handling, JSP Session Tracking, JSP Database Access, JSP Standard Tag Libraries,

JSPCustom Tag, JSP Expression Language, JSP Exception Handling, JSP XML Processing. 12 Hours

UNIT – IV

Hibernate 4.0: Overview of Hibernate, Hibernate Architecture, Hibernate Mapping, Types, Hibernate O/R Mapping, Hibernate Annotation, Hibernate Query Language.10 Hours

UNIT – V

Java Web Frameworks: Spring MVC: Overview of Spring, Spring Architecture, bean life cycle, XML. Configuration on Spring, Aspect – oriented Spring, Managing Database, Managing Transaction.10 Hours

Text Books:

1. Black Book "Java Server Programming" J2EE, 1st ed., Dream Tech Publishers, 2008.
2. Complete Reference J2EE by James Keogh, McGraw publication, 2017

Reference Books:

1. Professional Java Server Programming by Subrahmanyam Allamaraju, Cedric Buest, 1.3 edition, Apress Publication.
2. Core Java, Volume II: Advanced Features by Cay Horstmann and Gary Cornell, Ninth edition, 2013, Pearson Publication,
3. Java Persistence with Hibernate by Christian Bauer, Gavin King, 2nd edition, 2015, Manning publication
4. Spring in Action Craig walls, 5thedition, 2018, Manning Publication,
5. Hibernate Recipes, Joeseph etingar, Srinivas Guruju, Gray Mak, 2nd edition, 2015 Aprèss publication
6. Java Server Faces in Action, Kito D. Mann, Manning Publication, 2005

Course Outcomes:

At the end of the course, the student is able to:

- **CO1**: Students will be able to design and implement JDBC-based applications for effective database interaction and management. *(L6: Create)*
- **CO2**: Students will gain the ability to build and manage dynamic web applications using Servlet API and JSP. *(L3: Apply)*
- **CO3**: Students will develop expertise in Hibernate ORM to manage database entities within Java applications effectively. *(L4: Analyze)*
- **CO4**: Students will be able to implement full-stack Java applications using the Spring MVC framework. *(L5: Evaluate)*
- **CO5**: Students will demonstrate the ability to manage transactions, session tracking, and state in web applications using advanced Java frameworks. *(L5: Evaluate)*

CO-PO Mapping Table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	3	3	3	3	3
CO2	2	3	2	3	2	1	1	2
CO3	3	2	3	3	2	1	2	3
CO4	2	3	3	3	2	2	2	3
CO5	2	3	2	3	2	3	2	2

Note: 1 - Low Correlation, 2 - Medium Correlation, 3 - High Correlation,

IV SEMESTER

20SCA 868: BIG DATA ANALYTCS LAB

Hours per week: 4 Examination: 100 Marks

Credits: 2

Preamble:

With the exponential growth of data, organizations need efficient tools and frameworks to store, manage, and analyze vast datasets. Big Data technologies like Hadoop, Hive, and Pig provide scalable and distributed solutions to handle these challenges. This lab-based course equips students with hands-on experience in working with Hadoop Distributed File System (HDFS), MapReduce programming, modern file formats, and big data tools like Hive, Pig, and Sqoop. The course aims to foster practical skills for managing, processing, and analyzing large datasets in distributed computing environments.

Course Objectives:

- **Understand** the architecture and functioning of the Hadoop Distributed File System (HDFS) and its API.
- **Develop** skills in running and managing MapReduce programs for distributed data processing tasks.
- **Explore** the use of modern file formats (Parquet, ORC, JSON, Avro) and compression techniques in Hive and Pig for efficient data storage and retrieval.
- **Implement** data ingestion techniques using Sqoop to transfer data from relational databases to Hadoop ecosystems, and prepare Hive tables using HCatalog.
- **Understand** High Availability (HA) and launch big data processing jobs on YARN clusters for better resource management.

1.Exploring Hadoop Distributed File System (HDFS).

2.Implementation of file system commands using Hadoop file system API.

1. Implementation of HDFS file watchers to monitor the events on specific directory path.
2. MapReduce: Running the Word Count Program, calculating the size of each word, no. of alphabets in the entire dataset.
3. Modern file formats like Parquet, ORC, JSON and avro along with compressions like snappy, gz, bzip2, lzo etc. both with Pig and Hive using partitioning, bucketing, map joins, vectorizations in hive.
4. Preparing hive tables with above said file formats using sqoop & Hcatalog from RDBMS like sql-server.
5. Understanding HA and launching jobs on YARN cluster mode.

Text Books:

1. Big Data Black Book by Dt Editorial Services, Dreamtech Publications,2016.
2. Hadoop The Definitive Guide by Tom White, O'reilly ,4th Edition,2016.
3. Programming Hive- Jason Rutherglen, Dean Wampler, Edward Capriolo, O'reilly Publisher, 1st edition,2012.

Course Outcomes:

At the end of the course, the student is able to

- CO1: Navigate and perform operations on the Hadoop Distributed File System (HDFS) using file system commands and APIs. *(L3: Apply)*

- CO2: Implement MapReduce programs for tasks such as word counting and analyzing dataset characteristics. *(L4: Analyze)*
- CO3: Utilize modern file formats and compression techniques in Hive and Pig to optimize data storage and queries. *(L5: Evaluate)*
- CO4: Transfer data from RDBMS to Hadoop ecosystems using Sqoop and prepare Hive tables with various file formats for analysis. *(L3: Apply)*
- CO5: Manage and run big data jobs on YARN cluster mode, understanding the principles of high availability (HA) in Hadoop environments. *(L5: Evaluate)*

CO-PO Mapping Table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	3	1	-	-	2
CO2	3	3	3	3	1	-	-	2
CO3	3	2	3	3	1	-	-	2
CO4	3	3	3	3	1	-	-	2
CO5	3	2	3	3	2	-	-	2

Note: 1 - Low Correlation, 2 - Medium Correlation, 3 - High Correlation, - No correlation

IV SEMESTER

0SCA 870: PYTHON PROGRAMMING LAB

Hours per week: 4 Examination: 100 Marks

Credits: 2

Preamble:

In today's data-driven world, programming skills are fundamental to analyzing complex problems and developing intelligent systems. This course offers hands-on experience in Python programming and the implementation of various machine learning algorithms, including K-Nearest Neighbors (KNN), Linear Regression, Clustering, and more. Through a mix of coding challenges and the application of these techniques using tools like Weka and Scikit-learn, students will gain a strong foundation in computational problem-solving and data analysis. The course is designed to empower students to understand and implement key algorithms, preparing them for real-world applications in machine learning and data science.

Course Objectives:

- **Develop** proficiency in Python programming by solving practical problems using loops, recursion, file handling, and custom functions.
- **Understand** and implement key machine learning algorithms, such as K-Nearest Neighbors (KNN) and Linear Regression, using Python and Weka tools.
- **Explore** clustering techniques like K-Means and Hierarchical Clustering to analyze data and evaluate performance using validation techniques like RMSE.
- **Apply** classification models such as Logistic Regression and Support Vector Machines (SVM) on real-world datasets, evaluating their performance with cross-validation.
- **Learn** to use Scikit-learn for implementing advanced algorithms like PCA, K-Means, and matrix decomposition to solve machine learning tasks.

1. Write a python program that displays the sum of all digits for a user entered number.
2. Write a python function leap year that prints all the leap years between ranges. The user will enter lower and upper year boundary inside the function.
3. Write a program that outputs all possible strings formed by using the characters a, c, t, o, and g. a particular character can appear only once and all the characters should be used in the formation of string.
4. Write a python script that takes input from file representing a paragraph, and writes to a file named out.txt with all the stop words (a, an, the) removed.
5. Write a recursive function in python to print a Fibonacci series. The Fibonacci sequence is the series of numbers: 0,1,1,2,3,5,8,13,21,34,…etc
6. Write a program for sorting the integer data by using quick sort.
7. Implement the KNN (K Nearest Neighbours) algorithm in python. Your program should have different functions as follows:

   i) Handle Data: Open the dataset from CSV and split into test/train (datasets). A ratio of 67/33 for train/test is a standard ratio used for splitting data.

   ii) Similarity: Calculate the distance between two data instances. The Euclidean distance is used for calculating the difference. It is defined as the square root of the sum of the squared differences between the two arrays of numbers. Only first 4 attributes are used for calculating the distance.

iii) Neighbours: Locate k most similar data instances.

iv) Response: Generate a response from a set of data instances. It is a function for getting the majority voted response from a number of neighbours. It devises a predicted response based on those neighbours.

v) Accuracy: Summarize the accuracy of predictions. An easy way to evaluate the accuracy of the model is to calculate a ratio of the total correct predictions out of all predictions made, called the classification accuracy.

vi) Main: Take split = 0.67, k=3.

- Apply the KNN algorithm in Weka tool on the iris dataset. Compare the results of your implemented algorithm with algorithm of Weka tool. 2. Implement the linear Regression. The data will be taken as input from the file. Select the appropriate dataset from the website https://archive.ics.uci.edu/ml/index.php". Justify the reason why the dataset has been selected. b) Apply the Linear regression in Weka tool on the same dataset. Compare the results of your implemented algorithm with algorithm of Weka tool.Clustering: Remove the label column of the Parkinson_dataset.csv dataset and implement the following: a) Perform K-Means clustering and Hierarchical clustering. b) Use Manhattan distance c) Use Average merging Strategy in Hierarchical clustering. d) Use three different K values in K-Mean clustering. e) Validate using RMSE and compare both the techniques

- Logistic regression and SVM : Divide the Parkinson_dataset.csv dataset in training and testing dataset randomly and implement the following:
  a. Classify the disease using Logistic regression and SVM
  b. Find out the accuracy of classification Model.

c. Perform 5-fold cross- validation.

d. Compare the result of both techniques using matplotlib

- Sci-kit learn tool Kit :

Implementation of the following algorithms in scikit-learn

a. Principal components analysis (PCA)

b. Decomposing signals in components (matrix CO5 factorization problems)

c. K-means.

Course Outcomes:

At the end of the course, the student is able to

- CO1: Develop Python programs to solve algorithmic problems, such as finding leap years, generating Fibonacci sequences, and sorting data using Quick Sort. *(L3: Apply)*
- CO2: Implement and evaluate the KNN algorithm from scratch using Python, and compare results with Weka tool implementation for practical data analysis. *(L5: Evaluate)*
- CO3: Perform data clustering using K-Means and Hierarchical clustering methods, applying various distance measures and validation strategies to compare their effectiveness. *(L4: Analyze)*
- CO4: Classify data using Logistic Regression and SVM algorithms, evaluate model accuracy, and apply cross-validation techniques for improved performance. *(L5: Evaluate)*
- CO5: Utilize the Scikit-learn toolkit to implement advanced machine learning techniques like PCA, matrix decomposition, and K-Means clustering, visualizing the results with Matplotlib. *(L6: Create)*

CO-PO Mapping Table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	1	-	-	1
CO2	3	3	2	3	1	-	-	1
CO3	3	3	3	3	1	-	-	2
CO4	3	3	3	3	1	-	-	2
CO5	3	2	3	3	2	-	-	2

Note: 1 - Low Correlation, 2 - Medium Correlation, 3 - High Correlation, - No correlation

IV SEMESTER

20SCA 872: ADVANCED JAVA PROGRAMMING LAB

Hours per week: 4 Examination: 100 Marks

Credits: 2

Preamble:

In modern software development, Java continues to be one of the most widely used programming languages, particularly in enterprise and web applications. This course, "Advanced Java Programming Lab," aims to deepen students' understanding of advanced Java technologies such as socket programming, Hibernate, Spring Framework, and Java Database Connectivity (JDBC). Through hands-on lab exercises, students will gain practical experience in developing robust applications that demonstrate effective client-server communication, database integration, and web services. The course encourages students to apply theoretical knowledge to real-world problems, fostering creativity and technical mastery in Java programming.

Course Objectives:

- Understand and implement socket programming using TCP/UDP protocols for client-server communication.
- Develop distributed applications using Remote Method Invocation (RMI) with custom security policies.
- Perform database operations using JDBC and Servlets for efficient data handling.
- Design dynamic web applications using JSP, Servlets, and state management techniques.
- Apply the Model-View-Controller (MVC) architecture using the Spring Framework to build scalable applications.

Socket Programming(TCP/UPD)

- Create chat application using either TCP or UDP protocol.
- Implement TCP Server for transferring files using Socket and Server Socket
- Implement any one sorting algorithm using TCP/UDP on Server application and Give
- Input On Client side and client should sorted output from server and display sorted on input side.
- Implement Concurrent TCP Server programming in which more than one client can connect and communicate with Server for sending the string and server returns the reverse of string to each of client
- Write RMI application where client supplies two numbers and server response by summing it. Provide your custom security policy for this application.
- Implement Student information system using JDBC and RMI.

JDBC/Servlet

1. Create Servlet file which contains following functions:

i) Connect ii) Create Database iii) Create Table iv) Insert Records into respective table

v) Update records of particular table of database vi) Delete Records from table.

vii) Delete table and also database.

2. User can create a new database and also create new table under that database. Once database has been created then user can perform database operation by calling above

functions. Use following Java Statement interface to implement program:

a) Statement b) Prepared statement c) Callable statement

3. Create Servlet file and study web descriptor file.

4. Create login form and perform state management using Cookies, Http Session and URL Rewriting.

5.Implement Authentication filter using filter API.

6. Create database of student subject-wise data and retrieve all data using JSP and generate xml structure along with DTD and XML Schema definition

7. Apply XSLT (Style) to generated xml document and print your result. Create web service which provides student information.

8. Create Web Service client which consume above service and display student databy entering studentid.

9. Study and implement Hibernate

10. Study and Implement MVC using Spring Framework.

(C) Design based Problems (DP)/Open Ended Problem:

11.Using J2EE JSP/Servlet API develop student's management system required to manage student's academic activity such as student's profile, student's day to day assignment.

Course Outcomes:

At the end of the course, the student is able to:

- CO1: Develop client-server applications using TCP/UDP socket programming. *(L3: Apply)*
- CO2: Create distributed Java applications with RMI and implement security features. *(L6: Create)*
- CO3: Implement database-driven applications using JDBC and Servlets for CRUD operations. *(L3: Apply)*
- CO4: Design web applications with effective session management and authentication mechanisms. *(L5: Evaluate)*
- CO5: Apply MVC architecture using the Spring Framework to build robust web applications. *(L4: Apply)*

CO-PO Mapping Table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	3	2	1	-	-	1
CO2	3	2	3	2	1	-	2	1
CO3	3	3	3	2	1	-	-	1
CO4	3	2	3	2	1	-	2	1
CO5	3	3	3	3	2	2	-	2

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

IV SEMESTER

20SCA 891 : PROJECT

Hours per week: 4 End Examination: 50 Marks

Credits: 8

Sessionals: 150 Marks

Preamble:

In today's rapidly evolving technological landscape, Machine Learning and Data Mining are key pillars driving innovation across industries. With vast amounts of data available, developing intelligent systems that can analyze and derive actionable insights is paramount. This course, "PROJECT," provides students with the opportunity to apply their knowledge of machine learning, data mining, and software development automation. Through in-depth research, problem identification, algorithmic improvements, and hands-on project development, students will gain the practical skills needed to tackle real-world challenges. The project encourages critical thinking, creativity, and technical competence, preparing students for successful careers in data-driven and automated software environments.

Course Objectives:

Upon completion of this course, students will be able to:

- Explore and specify a research project topic within the domains of Machine Learning and Data Mining.
- Conduct a literature review by studying high-quality research papers, understanding existing methodologies, and identifying gaps.
- Analyze selected research papers through SWOT analysis to identify opportunities for improvement.
- Formulate a research problem based on the literature review and propose novel contributions or improvements in algorithm design or feature enhancement.
- Develop the system architecture for the proposed solution and design a comprehensive test bed.
- Implement the proposed solution using appropriate libraries and frameworks, focusing on Python for experimentation.
- Perform extensive testing and evaluation of the proposed solution, analyzing experimental results to validate the effectiveness.
- Understand the software development automation process and apply relevant tools to streamline project implementation.
- Design and implement automation software using frameworks, APIs, and libraries, while evaluating the developed project with respect to software automation standards.
- Prepare a detailed technical report, documenting the problem statement, methodology, design, test plan, and final implementation of the project.

1. Specify the broad topic of the project based on the Machine Learning and Data mining.
2. Study minimum 6 quality research papers based on the selected topic.
3. Do the SWOT analysis of selected research papers/reports.
4. Identify the research problem.
5. Propose your novelty/improvement in terms of algorithm/new feature.
6. Design the architecture for the proposed problem.
7. Design the test bed.
8. Design a set of experiments to be carried out for the proposed problem.
9. Perform the experimental analysis (in Python language only).
10. Prepare your report.

Write a short research paper based on your contribution.

1. Understand the Software Development Automation processes and work to develop a project on software development automation. Understanding Level (Level II)
2. Conduct preliminary literature Review, study different automation tools and find vulnerabilities in the studied literature/tools. Understanding Level (Level II)
3. Analyze and identify the various frameworks, APIs , libraries and tools used for project/ software implementation. Analyzing Level (Level III)
4. Design Software Development Automation software using required frameworks, APIs and libraries. Applying Level (Level IV)
5. Evaluate and validate developed project with respect to various software automation frameworks. Evaluating Level (Level V)
6 .Prepare technical detailed report detailing the problem statement, proposed methodology, software specification, design, test plan, and implementation details. Creating Level (Level VI)

Course Outcomes:

At the end of the course, the student is able to:

- CO1: Understand the principles of Machine Learning, Data Mining, and Software Development Automation. *(L2: Understand)*
- CO2: Critically analyze existing literature, frameworks, and tools to identify research gaps and vulnerabilities. *(L2: Understand, L3: Analyze)*
- CO3: Design and propose novel solutions or enhancements to existing algorithms and frameworks. *(L4: Apply)*
- CO4: Develop a project that incorporates their proposed solution using Python, alongside relevant software automation frameworks. *(L4: Apply, L5: Evaluate)*
- CO5: Evaluate the developed solution against industry-standard frameworks and methodologies, ensuring its validity. *(L5: Evaluate)*
- CO6: Create a detailed technical report summarizing their research, solution, and experimental findings, contributing to their professional portfolio. *(L6: Create)*