

SYLLABUS



BACHLOR OF COMPUTER APPLICATION

(BCA – CKUG02A03)

(Effective from Academic Year 2023-24)

2023-24



DR. C.V. RAMAN UNIVERSITY

AN ISO 9001:2015 CERTIFIED FOR QMS

// Madhya Pradesh, Khandwa AN AISECT GROUP UNIVERSITY

Approved by : MP. Govt., AICTE, NCTE, MP PARAMEDICAL COUNCIL, PCI Recognized by : UGC Member of : AIU

Statuary Warning: This is a classified document for the internal circulation strictly, no part thereof can be changed, altered, revised omitted or deleted without approval of the Academic Council of Dr. C.V. Raman University, Khandwa.

By the order of honorable Vice Chancellor.

This syllabus content Page number- 135

Preamble

The objective of any program at Higher Education Institute (HEI) is to prepare future professionals well equipped with attitude – skills and knowledge demanded by the growing society. The CVRUK envisions all its programs in the best interest of its students and in this endeavor; it offers a new vision to all its under graduate courses. It imbibes Learning Outcome-Based Curriculum Framework (LOCF) for all its under graduate programs. The LOCF approach is envisioned to provide focused, outcome – based syllabus at the undergraduate level with an agenda to structure the teaching – learning experiences in a more student-centric manner.

The LOCF approach has been adopted to strengthen student’s experience as they engage themselves in the program of their choice. The under graduate programs will prepare the students both for academic pursuit as well as enhance her/his employability.

Each program vividly elaborates its nature and promises the outcomes that are to be accomplished by studying the courses. The programs also state the attributes that it offers to inculcate at the graduation level. The graduate attributes encompass values related to well-being, emotional stability, critical thinking, social justice also skills for employability. In short, each program prepares students for sustainability and life-long learning.

The new curriculum Bachelor of Computer Application offers the students to gain the requisite knowledge, skills, and aptitude for the field of Computer Science and Information Technology. The efforts are made to measure cognitive as well as applied learning. Students are not only trained on the core components but also in areas which are need based, innovative, and relevant keeping in pace with the fast-growing industry. The course is internationally competitive.

The CVRUK hopes the LOCF approach of the program Bachelor of Computer Application will help students in making an informed decision regarding the goals that they wish to pursue in further education and life, at large.

1. Introduction to Bachelor of Computer Application–

Bachelor in Computer Application is an Undergraduate degree course in Computer Application with the rapid growth of IT industry which has created and creating lots of opportunities for the computer graduates. Bachelor of Computer Application is one of the popular fast moving technical fields. The duration of the course is 3 years divided into 6 semesters. This course provides a lot of opportunities to the students who are interested in the Information Technology field.

The objectives of the program shall be to provide sound academic base from which an advanced career in Computer Applications can be developed conceptual grounding in computer usage as well as its practical business application will be provided.

The Bachelor of Computer Application course in Choice Based Credit System is of 3 - year duration which comprises of 6 semesters, divided into 11 Major Core papers, 6 Minor Core Papers, 3 Inter Disciplinary Course (IDC), 4 Skill Enhancement Elective Courses (SEC), 4 Ability Enhancement Courses (AEC) and 2 Value added Courses (VAC). Each year consists of 2 semesters. This course has been prepared keeping in view, the unique requirements of Bachelor of Computer Application students.

The objectives of the program are:

- To acquaint the students with the policy domain and related legislations applicable in setting up and operating businesses in the Computer Application-related industry.
- To impart knowledge in areas related to Computer Applications, including programming languages, software development, database management, and information technology.
- To enable the students to understand the Computer Application sector/industry along with its economic, social and ethical aspects.
- To acquaint the students with the essential skills/qualities required in the field of Computer Applications and foster their professional development.
- To emphasize the importance of course-specific knowledge and quality management, adhering to national and international industry standards, regulations, and best practices in the field of Computer Applications.

The Bachelor of Computer Application (BCA) curriculum is thoughtfully crafted to align with the dynamic advancements in Computer Science and Information Technology. Given that Computer Science is an interdisciplinary field, it is highly recommended for students to opt for subjects such as AEC, SEC and other related courses as their Interdisciplinary (ID) choices, as they complement and enhance the overall curriculum. These subjects have a synergistic effect and enable students to gain a comprehensive understanding of the diverse aspects of the Computer Application domain. Nevertheless, students also have the flexibility to explore and select Interdisciplinary Courses from other departments based on their individual interests and academic inclinations. This approach encourages a well-rounded education and prepares students to adapt to the ever-evolving landscape of the IT industry.

2. Learning outcome-based curriculum framework

2.1 Nature and Extent of The Program in Bachelor of Computer Application

The learning outcomes-based curriculum framework is based on the premise that every student and graduate is unique. Each student or graduate has his/her own characteristics in terms of previous learning levels and experiences, life experiences, learning styles and approaches to future career related actions. The quality, depth and breadth of the learning experiences made available to the students while at the higher education institutions help develop their characteristic attributes.

2.2 Aims of Bachelor Degree Program in Bachelor of Computer Application

The key objectives that underpin curriculum planning and development at the undergraduate level include Program Learning Outcomes, and Course Learning Outcomes. For the Bachelor of Computer Application course, it includes:

- To make students and aspirants aware of the policy domain and related legislations in the upcoming IT industry not only in regional or national perspective but global perspective as well;
- To demonstrate comprehensive knowledge and understanding of the IT Technology curriculum;
- To apply the principles of Computer Science and Information Technology to assure the quality and safety of software/products;
- To understand that the real-world problems in the IT industry require continuous acquisition of knowledge and its application to improve the safety and quality of a given system or process;
- To analyse, interpret and draw conclusions from quantitative/qualitative data; and critically evaluate ideas, evidence, and experiences from an open-minded and reasoned perspective;
- To acquire knowledge and skills, including “learning how to learn”, that are necessary for participating in learning activities throughout life, through self-paced and self-directed learning aimed at personal development, and adapting to changing trades and demands of work place through knowledge/skill development/reskilling;
- To use ICT in a variety of learning situations, demonstrate ability to access, evaluate, and use a variety of relevant information sources;
- To acquire professional competency and entrepreneurial skills for economic empowerment.
- To demonstrate the ability to acquire, analyse, interpret, and appropriately present laboratory data;

3. Graduate Attributes in Bachelor of Computer Application

Disciplinary Knowledge

Students pursuing a Bachelor of Computer Application degree will acquire in-depth knowledge and understanding of various core courses, including Computer Programming, Data Structures, algorithms, database management, operating systems, and Computer Networks. They will also gain expertise in software development, Computer Hardware, Artificial Intelligence, Machine Learning, and Cyber Security. Throughout the course, students will be exposed to a wide range of programming languages and software tools used in the industry. Additionally, they will develop problem-solving and analytical skills, which are crucial for tackling real-world challenges in the field of computer science. The program aims to prepare students for successful careers as Computer scientists, Software Engineers, Data analysts, and Technology consultants, among other rewarding opportunities in the rapidly evolving technological landscape.

Communication Skills

Development of student's communication skills is planned through an AEC paper (English) which is compulsory for each student. Besides that, the students do various assignments that enable them to develop skills in public speaking writing and effective's interpersonal skills. Presentations in each paper enhances their confidence, ability to express themselves and presentation skills.

Research-Related Skills

Students develop a scientific temper and a sense of enquiry through various Information Technology papers. They have capabilities in asking relevant questions relating to current issues and themes and state hypothesis and rationale for inquiry. Students can use appropriate research methodology especially for understanding issues in Computer Science and Information Technology and reporting the results in different formats.

Cooperation/Team Work

Students are capable of effective working in diverse contexts and teams in class rooms laboratories, student societies, industry, and the community. They have basic management skills for independently organizing events, resource mobilization and leading community-based projects, initiatives; cultural shows.

Self-Directed Learning

Students can work independently and are able to apply the concepts of Computer Science and Information Technology in an original creative manner to solve and manage real life issues for the customers and industry. Students develop customized processes and or products as per the requirements of society.

Multicultural Competence

Students are confident of working in diverse socio-cultural contexts. They can effectively engage with multicultural groups and teams. They have sensitivities of cross cultural and ethnic diversity which they can apply to different settings. College through a student and faculty exchange program with foreign University helps them to acquire multicultural competency. They are competent to seek higher education in foreign Universities.

Moral and Ethical Awareness/Reasoning

Student has awareness of ethical conduct in different situations (academic and personal). They have skills in understanding and avoiding unethical behavior such as misrepresentation, plagiarism and environmental misuse and violence. They are formally taught ethics of research and human interventions.

Leadership Readiness/Qualities

Students have leadership qualities in organizing teams and their mobilization for effective problem solving in different Computer Science and Information Technology aspects.

Students apply creative leadership for realization of various goals. As a leader, they are trained to have greater customer Sensitivity and connect. They can organize events and for a Computer Science and Information Technology and design business plans.

Lifelong Learning

Students acquire ability to gain knowledge and skills which are necessary in life for the holistic development for meeting their professional and personal needs in varying environment and changing contexts.

4. Qualification Descriptors for Bachelor of Computer Application

The following descriptors indicate the expectations from

- They will understand the science and technologies of information processing.
- They will understand computational and algorithmic aspects.
- They will possess knowledge of computer science applications, both domestically and globally.
- They will be versant with key principles of software development.

5. Programme Learning Outcome in Bachelor of Computer Application

The learning outcome of the program are-

- Knowledge of various areas related to computer applications,
- Understanding of the technological aspects,
- Know-how of programming languages and software tools,
- Relevance and significance of information technology in the modern world.

6. Structure of Bachelor of Computer Application

The Bachelor of Computer Application program will be of three years duration. Each year will be called an academic year and will be divided into two semesters, thus there will be a total of six semesters. Each semester will consist of sixteen weeks.

The Bachelor of Computer Application course in Choice Based Credit System is of 3 - year duration which comprises of 6 semesters, divided into 11 Major Core papers, 6 Minor Core Papers, 3 Interdisciplinary Course (IDC), 4 Skill Enhancement Elective Courses (SEC) and 4 Ability Enhancement Courses (AEC) and 2 Value added courses (VAC). Each year consists of 2 semesters. This course has been prepared keeping in view, the unique requirements of Bachelor of Computer Application students.

BACHELOR OF COMPUTER APPLICATION
Duration: 36 Months (3 Years) Eligibility: 12th Pass with any Discipline

COURSE STRUCTURE OF BCA I st SEMESTER											
Course Code		Course Details		External Assessment		Internal Assessment		Credit Distribution		Allotted Credits	
Course Code	Course Type	Course Title	Total Marks	Max Marks	Min Marks	Max Marks	Min Marks	L	T	P	Course wise Distribution
Theory Group											
3HHLA101	Ability Enhancement Course	Hindi language	100	60	20	40	14	2	-	-	2
3IST1102/3ICN1102	Interdisciplinary Course	Statistics/Computer Network	100	60	20	40	14	3	-	-	3
3SEEV106/3IFAV106	Value Added Course	Environmental Education / Fundamentals of AI	100	60	20	40	14	3	-	-	3
3IPCC103	Major Core	Programming in C	100	60	20	40	14	4	-	-	4
3IFCM104	Minor Core	Fundamentals of Computers & Information Technology	100	60	20	40	14	4	-	-	4
Practical Group											
				Term End Practical Exam		Sessional					
3IPCC103	Major Core	Programming in C	100	60	20	40	14			2	2
3IWDS105	Skill Enhancement course	Web Development-I (HTML & CSS)	100	60	20	40	14	-	1	1	2
Grand Total			700	-	-	-	-	16	1	3	20

Minimum Passing Marks are equivalent to Grade DL- Lectures T - Tutorials P - Practical

External Theory & Practical will carry 60 marks.

Internal Assessment (Theory & Practical) will carry total of 40 marks.

Internal Assessment – Attendance 75%, Pre-University Test (PUT)/Assignments

Note-List of AEC, VAC, SEC, IDC, MAJOR and MINOR subjects are enclosed after the scheme.

BACHELOR OF COMPUTER APPLICATION
Duration: 36 Months (3 Years) Eligibility: 12th Pass with any Discipline
COURSE STRUCTURE OF BCA IInd SEMESTER

Course Code		Course Details		External Assessment		Internal Assessment		Credit Distribution			Allocated Credits
Course Code	Course Type	Course Title	Total Marks	Max Marks	Min Marks	Max Marks	Min Marks	L	T	P	Course wise Distribution
				Max Marks	Min Marks	Max Marks	Min Marks	Sessional			
Theory Group											
3HELA201	Ability Enhancement Course	English Language	100	60	20	40	14	2	-	-	2
3IOSI202	Interdisciplinary Course	Operating System	100	60	20	40	14	3	-	-	3
3ICSV206/ 3HYEV206/ 3HCIV206	Value Added Course	Cyber Security / *Yoga Education/ Contemporary India	100	60	20	40	14	3	-	-	3
3IOPC203	Major Core	Object Oriented Programming with C++	100	60	20	40	14	4	-	-	4
3IWNM204	Minor Core	Windows & MS Office	100	60	20	40	14	2	-	-	2
Practical Group											
3IOPC203	Major Core	Object Oriented Programming with C++	100	60	20	40	14	-	-	2	2
3IWNM204	Minor Core	Windows & MS Office	100	60	20	40	14	-	-	2	2
3IWDS205	Skill Enhancement course	Web Development-II(Java Script)	100	60	20	40	14	-	1	1	2
Grand Total			800					14	1	5	20

*For value added course Yoga Education credit distribution will be

Practical Group		Total Marks	Max Marks	Min Marks	Max Marks	Min Marks	L	T	P	Total Credit
3HYEV206	Value Added Course Yoga Education	100	60	20	40	14	-	1	2	3

Minimum Passing Marks are equivalent to Grade DL- Lectures T- Tutorials P- Practical External Theory & Practical will carry 60 marks.

Internal Assessment (Theory & Practical) will carry total of 40 marks.

Internal Assessment – Attendance 75%, Pre-University Test (PUT)/Assignments

Note-List of AEC, VAC, SEC, IDC, MAJOR and MINOR subjects are enclosed after the scheme.

After Second Sem

Students existing the programme after securing 40 credits will be awarded UG Certificate in the relevant Discipline /Subject provided they secure 4 credits in work based vocational courses offered during summer term or internship / Apprenticeship in addition to 6 credits from skill-based courses earned during first and second semester.



BACHELOR OF COMPUTER APPLICATION
Duration: 36 Months (3 Years) Eligibility: 12th Pass with any Discipline

COURSE STRUCTURE OF BCA III rd SEMESTER													
Course Details			External Assessment			Internal Assessment			Credit Distribution		Allotted Credits		
Course Code	Course Type	Course Title	Total Marks	Max Marks	Min Marks	Max Marks	Min Marks	L	T	P			
Theory Group													
3HCSA301	Ability Enhancement Course	Communication Skill	100	60	20	40	14	2	-	-	2		
3IDMI302/ 3IOAI302	Interdisciplinary Course	Discrete Math's/ *Office Automation	100	60	20	40	14	3	-	-	3		
3IPJC303	Major Core	Programming with JAVA	100	60	20	40	14	4	-	-	4		
3IDMM304	Minor Core	Database Management System	100	60	20	40	14	4	-	-	4		
Practical Group													
			Term End Practical Exam			Sessional							
3IPJC303	Major Core	Programming with JAVA	100	60	20	40	14	-	-	2	2		
3IDMM304	Minor Core	Database Management System	100	60	20	40	14	-	-	2	2		
3IWDSS305	Skill Enhancement course	Web Development-III (React.js)	100	60	20	40	14	-	1	2	3		
Grand Total			700							13	1	6	20

***For Interdisciplinary CourseOffice Automation credit distribution will be**

310AI302	Practical Group		Total Marks	Max Marks	Min Marks	Max Marks	Min Marks	L	T	P	Total Credit
	Interdisciplinary Course	Office Automation									
			100	60	20	40	14	-	1	2	3

Minimum Passing Marks are equivalent to Grade DL- Lectures T- Tutorials P- Practical External Theory & Practical will carry 60 marks.

Internal Assessment (Theory & Practical) will carry total of 40 marks.

Internal Assessment – Attendance 75%, Pre-University Test (PUT)/Assignments

Note-List of AEC, VAC, SEC, IDC, MAJOR and MINOR subjects are enclosed after the scheme.

After IV Sem

Students existing the programme after securing 80 credits will be awarded UG Diploma in the relevant Discipline /Subject provided they secure additional 4 credit in skill based vocational courses offered during first year or second year summer term.



BACHELOR OF COMPUTER APPLICATION
Duration: 36 Months (3 Years) Eligibility: 12th Pass with any Discipline

+

COURSE STRUCTURE OF BCA IV th SEMESTER											
Course Details			External Assessment			Internal Assessment		Credit Distribution			Allocated Credits
Course Code	Course Type	Course Title	Total Marks	Max Marks	Min Marks	Max Marks	Min Marks	L	T	P	Course wise Distribution
Theory Group											
3HCHA401	Ability Enhancement Course	Cultural Heritage	100	60	20	40	14	2	-	-	2
3IPPC403	Major Core - I	Python Programming -I	100	60	20	40	14	4	-	-	4
3ISEC403	Major Core - II	Software Engineering & UML	100	60	20	40	14	5	1	-	6
3IDS404	Minor Core	Data Structure	100	60	20	40	14	4	-	-	4
Practical Group											
				Term End Practical Exam		Sessional					
3IPPC403	Major Core - I	Python Programming -I	100	60	20	40	14	-	-	2	2
3IDS404	Minor Core	Data Structure	100	60	20	40	14	-	-	2	2
Grand Total			600					15	1	4	20

Minimum Passing Marks are equivalent to Grade DL- Lectures T- Tutorials P- Practical

External Theory & Practical will carry 60 marks.

Internal Assessment (Theory & Practical) will carry total of 40 marks.

Internal Assessment – Attendance 75%, Pre-University Test (PUT)/Assignments

Note-List of AEC, VAC, SEC, IDC, MAJOR and MINOR subjects are enclosed after the scheme.

BACHELOR OF COMPUTER APPLICATION
Duration: 36 Months (3 Years) Eligibility: 12th Pass with any Discipline

COURSE STRUCTURE OF BCA Vth SEMESTER												
Course Details			External Assessment			Internal Assessment			Credit Distribution		Allotted Credits	
Course Code	Course Type	Course Title	Total Marks	Max Marks	Min Marks	Max Marks	Min Marks	L	T	P	Course wise Distribution	
Theory Group												
3IPPC503	Major Core - I	Python Programming -II	100	60	20	40	14	4	-	-	4	
3IDSC503	Major Core - II	Distributed System	100	60	20	40	14	4	-	-	4	
3IITC503	Major Core -III	Internet of Things	100	60	20	40	14	4	-	-	4	
3IFDC504	Minor Core	Fundamentals of Datascience	100	60	20	40	14	4	-	-	4	
Practical Group												
3IPPC503	Major Core - I	Python Programming -II	100	60	20	40	14	-	-	2	2	
3IIP1505	Internship	Internship/ Project	100	60	20	40	14	-	-	2	2	
Grand Total			600				Sessional			16	4	20

Minimum Passing Marks are equivalent to Grade DL- Lectures T - Tutorials P- Practical
 External Theory & Practical will carry 60 marks.

Internal Assessment (Theory & Practical) will carry total of 40 marks.

Internal Assessment – Attendance 75%, Pre-University Test (PUT)/Assignments

Note-List of AEC, VAC, SEC, IDC, MAJOR and MINOR subjects are enclosed after the scheme.

BACHELOR OF COMPUTER APPLICATION
Duration: 36 Months (3 Years) Eligibility: 12th Pass with any Discipline

COURSE STRUCTURE OF BCA VI th SEMESTER											
Course Details		Course Type	Course Title	Total Marks	External Assessment		Internal Assessment		Credit Distribution		Allotted Credits
Course Code					Max Marks	Min Marks	Max Marks	Min Marks	L	T	
Theory Group											
3IWPC603	Major Core – I	Web Programming with ASP.NET Using C#	100	60	20	40	14	4	-	-	4
3IROM603	Major Core – II	RDBMS With Oracle	100	60	20	40	14	4	-	-	4
3IDWC603	Major Core – III	Data Warehousing & Mining	100	60	20	40	14	4	-	-	4
3ICCM604	Minor Core	Computer Communication Network	100	60	20	40	14	4	-	-	4
Practical Group											
				Term End Exam	End Practical	Sessional					
3IWPC603	Major Core – I	Web Programming with ASP.NET Using C#	100	60	20	40	14	-	-	2	2
3IROM603	Major Core – II	RDBMS With Oracle	100	60	20	40	14	-	-	2	2
	Grand Total		600					16	-	4	20

Minimum Passing Marks are equivalent to Grade DL- Lectures T- Tutorials P- Practical
 External Theory & Practical will carry 60 marks.
 Internal Assessment (Theory & Practical) will carry total of 40 marks.
 Internal **Assessment** – Attendance 75%, Pre-University Test (PUT)/Assignments
 Note-List of AEC, VAC, SEC, IDC, MAJOR and MINOR subjects are enclosed after the scheme.

List of Major

S.N.	Course Type	Semester	Course Code	Paper Name	Credit (L-T-P)
1	MajorCore	I st Sem.	3IPCC103	Programming in C	6(4-0-2)
2	MajorCore	II nd Sem.	3IOPC203	Object Oriented Programming with C++	6(4-0-2)
3	MajorCore	III rd Sem.	3IPJC303	Programming with JAVA	6(4-0-2)
4	MajorCore-I	IV th Sem.	3IPPC403	Python Programming -I	6(4-0-2)
	MajorCore-II		3ISEC403	Software Engineering & UML	6(5-1-0)
5	MajorCore-I	V th Sem.	3IPPC503	Python Programming -II	6(4-0-2)
	MajorCore-II		3IDSC503	Distributed System	4(4-0-0)
	MajorCore-III		3ITTC503	Internet of Things	4(4-0-0)
6	MajorCore-I	VI th Sem.	3IWPC603	Web Programming with ASP.NET Using C#	6(4-0-2)
	MajorCore-II		3IROM603	RDBMS with Oracle	6(4-0-2)
	MajorCore-III		3IDWC603	Data Warehousing & Mining	4(4-0-0)

Total Credit - 60

List of Minor

S.N.	Course Type	Semester	Course Code	Paper Name	Credit (L-T-P)
1	Minor Core	I Sem.	3IFCM104	Fundamentals of Computers and Information Technology	4(4-0-0)
2	Minor Core	II Sem.	3IWMM204	Windows and MS office	4(2-0-2)
3	Minor Core	III Sem.	3IDMM304	Database Management System	6(4-0-2)
4	Minor Core	IV Sem.	3IDSM404	Data Structure	6(4-0-2)
5	Minor Core	V Sem.	3IFDC504	Fundamentals of Data science	4(4-0-0)
6	Minor Core	VI Sem.	3ICCM604	Computer Communication Network	4(4-0-0)

Total Credit -28



BACHLOR OF COMPUTER APPLICATION

I Semester

(BCA – CKUG02A03)

(Effective from Academic Year 2023-24)

DR. C.V.RAMAN UNIVERSITY

MAJOR CORE

Programming in C - 3IPCC103

(Credit: Theory -4 Practical - 2)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical			Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work	Total		
3IPCC103	Programming in C	6(4-0-2)	60	30	10	60	40	200	3 hr	2 hr

Course objective

- To understand the basic knowledge of programming concepts.
- To understand the C language and its concepts.

Course outcomes

- Illustrate the flowchart and design an algorithm for a given problem and to develop IC programs using operators
- Develop conditional and iterative statements to write C programs
- Exercise user defined functions to solve real time problems
- Exercise files concept to show input and output of files in C.

Syllabus**Theory**

Unit-I Principles of Programming: Introduction to Programming, Program Concept, Characteristics of Programming, Stages in Program Development, Tips for Program Designing, Programming Aids, Algorithms, Notations, Design, Flowcharts, Symbols, Rules

Programming Techniques and Logic: Introduction, Introduction to programming techniques, Top-down approach or technique, Bottom-up approach or technique, Unstructured technique of programming, Structured technique of programming, Modular technique of programming, Comparative study of programming techniques, Cohesion, Coupling, Debugging, Syntax Errors, Logical Errors, Data Entry Errors, Linker Errors, Runtime Errors, Program Testing

Unit-II Turbo C IDE: Turbo C IDE (Integrated Development Environment), Main Menu Bar, File Options, edit option, run option, compile option, Project option, Options option, Debug option, Break/watch option, Edit Window, Message Window, Status bar, Editing, Compiling and Running a C Program, Features of C language, C language standards, Standardization, Successors of C language

Introduction to 'C': Introduction, Structure of a C program, 'C' Tokens, Keywords, Identifiers, 'C' Constants, Variables in C, Data Types, Derived Data Types: Operators, Precedence and Associativity of operators, Hierarchy of operators at a glance, Expression and its Evolution, Type conversion in expressions, (Implicit and Explicit type conversion).

Unit-III Decision Making and Branching: Introduction, Sequential statements, Unformatted I/O functions, Formatted input using scanf () function, Formatted output using print (), Branching statements, The if-else statement, The nested if-statement, The switch statement, Additional programs

Looping Statements: Introduction, for-statement, while-statement, do-while statement, Difference between while-loop and do-while loop, Nested loops, Jumps in loops, Programming examples

Unit-IV Arrays: Introduction, Single-dimensional arrays, Reading and writing single dimensional arrays, Examples of Complex Programs, Searching, Sorting, Two-dimensional arrays (multi-dimensional arrays), Reading-writing two-dimensional arrays, Manipulation in two-dimensional arrays, Programming Examples

Strings, Concepts of string, Strings in C language, String variable, Initializing strings, String input/output functions, Arrays of strings, String handling functions, Memory formatting

User Defined Functions: Introduction, Elements of user-defined functions, Categories of functions, Passing parameters to functions, Programming Examples, Arrays in functions, Nesting of Functions, Recursion, Command Line Arguments, Storage Classes.

Unit-V: Structure and Union: Introduction to structures, Structure and its definition, Structure declaration, Tagged Structure, structure variables, Type-Defined Structure, Structure initialization, accessing structures, Nested structures, Array of structures, Structures and functions, sending individual members, Sending the whole structure, passing structures through pointers, Uses of structures, Union and its definition

Debugging: Common Programming Errors, Program Testing and Debugging, Types of Errors, Debugging C program

Pointers: Introduction, Pointer concepts, Pointer variable, accessing variables through pointers, Pointer declaration and Definition, Initializing a pointer variable, Pointers to Pointers, Compatibility, Pointer applications, Pointers and other operators, Memory allocation functions, Memory map of C program, Memory management functions

File Handling: Introduction to file handling, File system basics, Standard streams in C, File structure, FILE pointer, Opening and closing a file, File handling functions, File types, Text and Binary, Input / Output operations on file, reading a character using `getc()`, Writing a character using `putc()`, Using `f()`, Working with string using `fputs()` and `fgets()`, Using `fprintf()` and `fscanf()`, Using `fread()` and `fwrite()`, Direct Access file, `fseek()`.

Practical's

1. Write a C Program to add two integer numbers.
2. Write a C Program to Check Whether a Number is Even or Odd.
3. Write a C Program to Check Whether a Number is Positive or Negative or Zero.
4. Write a C Program to Display Fibonacci Series.
5. Write a C Program to Reverse a Number.
6. Write a C Program to Check Whether a Number is Palindrome or Not.
7. Write a C Program to Make a Simple Calculator to Add, Subtract, Multiply or Divide Using `switch...case`.
8. Write a C Program to Calculate Factorial of a Number Using Recursion.
9. Write a C Program to Calculate Average Using Arrays.
10. Write a C Program to Add Two Matrix Using Multi-Dimensional Arrays.
11. Write a C Program to Swap Numbers in Cyclic Order Using Call by Reference.

References books

1. E. BALA GURUSWAMI, PROGRAMMING IN ANSI-C, MCGRAW HILL EDU.
2. ARUN PANDEY, PROGRAMMING IN C, Kamal Prakashan.
3. YASHWANT KANETKAR, LET US C, BPB Publication.

4. DR. R NAGESWARA RAO, CORE C, Dreamtech.
5. MAHAPATRA, Thinking In C, PHI Publications.
6. ANURAG SEETHA, Introduction to Computers and Information Technology, Gottfried, Bhopal.

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
1	<ol style="list-style-type: none"> 1. Understand the principles and concepts of programming. 2. Identify the stages involved in program development. 3. Apply programming techniques and logic to solve problems. 4. Design and develop algorithms using appropriate notations and symbols. 5. Create flowcharts to represent the logical structure of a program. <p>Debug and test programs to identify and fix errors.</p>	<ol style="list-style-type: none"> 1. Lectures: The instructor will deliver lectures on principles of programming, program development stages, programming techniques, and debugging methods. 2. Hands-on Exercises: Students will practice designing programs, writing algorithms, and creating flowcharts. 3. Group Discussions: Students will engage in discussions to compare and analyze different programming techniques. 4. Case Studies: Real-world examples will be discussed to understand the importance of program design and debugging. <p>Programming Assignments: Students will be given programming tasks to apply the learned concepts and techniques.</p>	<ol style="list-style-type: none"> 1. Quizzes: Short quizzes will be conducted to assess understanding of programming principles and techniques. 2. Programming Assignments: Students will submit programming assignments to demonstrate their ability to design and develop programs. 3. Tests: Written tests will be conducted to evaluate knowledge of program development stages, debugging techniques, and programming aids. 4. Project: Students will work on a programming project to apply all the learned concepts and techniques. <p>CCE: A Continuous Course Evaluation exam will be conducted to assess overall understanding of the unit's content, including programming techniques, logic, and debugging.</p>
2	<ol style="list-style-type: none"> 1. Understand the features and functionality of Turbo C IDE (Integrated Development Environment). 2. Familiarize with the main menu bar and various options available in Turbo C IDE. 3. Learn to navigate and utilize the File, Edit, Run, 	<ol style="list-style-type: none"> 1. Lectures and discussions on the concepts and features of Turbo C IDE. 2. Hands-on practice sessions using Turbo C IDE to navigate through the main menu bar and explore different options. 3. Practical exercises to edit, compile, and run C 	<ol style="list-style-type: none"> 1. Quizzes or tests to assess the understanding of Turbo C IDE features and functionality. 2. Practical assignments or projects to assess the ability to navigate and utilize Turbo C IDE options. 3. Coding exercises or programming assignments to

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
	<p>Compile, Project, Options, Debug, and Break/Watch options in Turbo C IDE.</p> <p>4. Gain proficiency in using the Edit Window, Message Window, and Status Bar for editing, compiling, and running C programs.</p> <p>5. Understand the basic structure of a C program and the different components it consists of.</p> <p>6. Identify and utilize different C language tokens, including keywords, identifiers, constants, and variables.</p> <p>7. Understand the concept of data types in C and be able to declare and use variables of different data types.</p> <p>8. Gain knowledge of derived data types in C and their usage in programming.</p> <p>9. Understand the various operators in C and their precedence and associativity.</p> <p>Learn about type conversion in expressions, including implicit and explicit type conversion.</p>	<p>programs using Turbo C IDE.</p> <p>4. Interactive sessions to understand the structure of a C program and the different components it consists of.</p> <p>5. Group activities and exercises to identify and analyze different C language tokens.</p> <p>6. Practical exercises to declare and use variables of different data types in C.</p> <p>7. Interactive sessions and exercises to understand and practice the usage of derived data types in C programming.</p> <p>8. Practical exercises and problem-solving sessions to explore and understand the various operators in C.</p> <p>9. Group discussions and exercises to analyze and evaluate the precedence and associativity of operators.</p> <p>Practical exercises and examples to demonstrate and practice type conversion in expressions, both implicit and explicit.</p>	<p>assess the ability to edit, compile, and run C programs using Turbo C IDE.</p> <p>4. Written assignments or exams to assess the understanding of the structure of a C program and the different components it consists of.</p> <p>5. Practical exercises or group activities to assess the identification and analysis of different C language tokens.</p> <p>6. Coding exercises or programming assignments to assess the ability to declare and use variables of different data types in C.</p> <p>7. Practical exercises or problem-solving tasks to assess the understanding and usage of derived data types in C programming.</p> <p>8. Written assignments or exams to assess the knowledge and application of various operators in C.</p> <p>9. Quizzes or tests to assess the understanding of the precedence and associativity of operators in C.</p> <p>10. Coding exercises or programming assignments to assess the ability to demonstrate and apply type conversion in expressions, both implicit and explicit.</p>
3	<p>1. Understand the concept of decision making and branching in programming</p> <p>2. Demonstrate the ability to write sequential statements and use unformatted I/O functions</p> <p>3. Apply the scanf() function for formatted input and the print() function for formatted output</p>	<p>1. Lecture sessions: The instructor will deliver lectures covering the concepts of decision making, branching, and looping statements. They will explain the syntax and usage of different constructs, provide examples, and clarify doubts.</p> <p>2. Hands-on coding exercises: Students will be given coding exercises</p>	<p>1. Written exams: Students will be assessed through written exams that will include multiple-choice questions, short answer questions, and coding problems related to decision making and looping statements.</p> <p>2. Coding assignments: Students will be given coding assignments to demonstrate</p>

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
	<p>4. Implement branching statements such as if-else, nested if, and switch statements</p> <p>5. Develop programs using additional decision-making constructs</p> <p>6. Explain the concept of looping statements in programming</p> <p>7. Use for, while, and do-while loops effectively in programming</p> <p>8. Differentiate between while-loop and do-while loop and understand their respective use cases</p> <p>9. Write programs with nested loops</p> <p>10. Understand and use jumps in loops for control flow</p> <p>Develop programming examples demonstrating the use of looping statements.</p>	<p>related to decision making, branching, and looping statements. They will practice writing programs using different constructs and ask questions for clarification.</p> <p>3. Group discussions: Students will participate in group discussions to understand the practical applications of decision making, branching, and looping statements. They will share their insights and approaches to solving programming problems.</p> <p>Code review and debugging: The instructor will review and provide feedback on the code written by students. They will help students identify and fix errors or bugs in their programs related to decision making and looping constructs.</p>	<p>their understanding of decision making and looping constructs. They will be evaluated based on the correctness of their code, adherence to coding standards, and the efficiency of their solutions.</p> <p>3. Project work: Students may be assigned a project that requires them to implement decision making and looping constructs to develop a specific program. They will be assessed based on the functionality, code quality, and creativity of their project.</p> <p>Class participation: Students will be evaluated on their active participation in class discussions, asking questions, and engaging in group activities related to decision making and looping statements.</p>
4	<p>1. Understand the concept and importance of arrays in programming.</p> <p>2. Demonstrate the ability to read and write single-dimensional arrays.</p> <p>3. Develop complex programs using arrays for efficient searching and sorting.</p> <p>4. Gain proficiency in working with two-dimensional arrays.</p> <p>5. Apply string handling functions for manipulating and formatting strings.</p>	<p>1. Lectures: The instructor will provide lectures to introduce and explain the concepts of arrays, strings, user-defined functions, recursion, command line arguments, and storage classes. The lectures will include examples and demonstrations to enhance understanding.</p> <p>2. Hands-on Programming Exercises: Students will be given programming exercises to practice reading and writing single-dimensional arrays, working with two-dimensional arrays, manipulating strings, and</p>	<p>1. Quizzes: Regular quizzes will be conducted to assess students' understanding of the concepts taught in the unit. These quizzes may include multiple-choice questions, short-answer questions, and programming problems.</p> <p>2. Programming Projects: Students will be required to complete programming projects that involve implementing various concepts learned in the unit. These projects will be assessed based on functionality, efficiency, and code quality.</p> <p>3. Examinations: Midterm and final examinations will</p>

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
	<p>6. Comprehend the elements and categories of user-defined functions.</p> <p>7. Demonstrate the ability to pass parameters to functions and work with arrays within functions.</p> <p>8. Understand and implement recursion in programming.</p> <p>Gain knowledge of command line arguments and storage classes.</p>	<p>implementing user-defined functions. These exercises will be done individually or in groups.</p> <p>3. Classroom Discussions: Classroom discussions will be held to clarify doubts, share different approaches to solving programming problems, and discuss real-world examples that demonstrate the application of arrays, strings, and functions.</p> <p>Programming Assignments: Students will be assigned programming assignments that involve developing complex programs using arrays, strings, and user-defined functions. These assignments will help students apply their knowledge and develop problem-solving skills.</p>	<p>be conducted to assess students' overall understanding of the unit. These exams may include theoretical questions, code writing questions, and problem-solving questions.</p> <p>4. Class Participation: Active participation in classroom discussions and activities will be considered for assessment. This includes asking questions, contributing to discussions, and engaging in collaborative problem-solving.</p> <p>Assignments and Homework: Regular assignments and homework will be given to assess students' understanding and application of concepts. These may include programming exercises, problem-solving tasks, and written assignments.</p>
5	<p>1. Understand the concept of structures and unions in C programming.</p> <p>2. Demonstrate the ability to declare and define structures and unions.</p> <p>3. Use structures and unions to store and manipulate data efficiently.</p> <p>4. Apply nested structures and arrays of structures in programming tasks.</p> <p>5. Understand the concept of pointers and their applications in C programming.</p> <p>6. Utilize pointers to access and manipulate variables effectively.</p>	<p>1. Lectures: The instructor will deliver lectures to explain the concepts of structures, unions, pointers, and file handling. Examples and demonstrations will be provided to enhance understanding.</p> <p>2. Discussions: Students will participate in discussions to clarify doubts and share their understanding of the topics.</p> <p>3. Hands-on Practice: Students will be given programming exercises to practice implementing structures, unions, pointers, and file handling operations.</p> <p>4. Code Review: Students will review and analyze code examples related to structures, unions, pointers, and file handling to</p>	<p>1. Quizzes: Short quizzes will be conducted to assess the understanding of concepts related to structures, unions, pointers, and file handling.</p> <p>2. Programming Assignments: Students will be given programming assignments to implement structures, unions, pointers, and file handling operations.</p> <p>3. Practical Exams: Practical exams will be conducted to assess the ability to write code and perform file handling operations.</p> <p>4. Project Work: Students will work on a project that involves the implementation of structures, unions, pointers, and file handling</p>

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
	<p>7. Understand the basics of file handling in C programming.</p> <p>8. Demonstrate the ability to perform input/output operations on files.</p> <p>9. Apply various file handling functions to read, write, and manipulate data in files.</p> <p>Understand and utilize direct access files using fseek () function.</p>	<p>understand their practical implementation.</p> <p>5. Assignments: Students will be assigned assignments to apply the concepts learned in real-world programming scenarios.</p> <p>6. Group Activities: Students will work in groups to solve programming problems using structures, unions, pointers, and file handling techniques.</p> <p>Lab Sessions: Students will have lab sessions to practice coding and gain hands-on experience with structures, unions, pointers, and file handling.</p>	<p>techniques to solve a real-world problem.</p> <p>Final Exam: The final exam will cover all the topics discussed in the course, including structures, unions, pointers, and file handling.</p>

MINOR CORE**3IFCM104 - Fundamentals of Computers and Information Technology**

(Credit: Theory -4 Tutorial - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.		
			Theory			Practical			Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work				
3IFCM104	Fundamentals Of Computers and Information Technology	4-0-0	60	20	20	-	-	100	3 hr	-	

Course objective

1. To understand the basic knowledge of computer
2. To understand the assembly-level programming
3. To understand the input output devices, storage media, memory.
4. To understand the concept of MIS, Networking devices.

Course outcomes

After study this student will be able to know about terms and concepts of Fundamentals of Computers and Information Technology (hardware, software, networking, security, Internet/Web, and applications).

Syllabus

Unit 1: Know the Computer -Introduction, what does computer stands for? Strengths of computers, Limitations of computers, Fundamental uses of computers, Development of computers, Types of Computers, Generations of Computers

Personal Computer - Introduction, Personal computer, Uses of personal computers, Components of personal computers, Evolution of PCs, Developments of processors, Architecture of Pentium iv, Configuration of pc

Number System - Introduction, Digital and Analog Operations, Binary Data, Binary Number System, Decimal Number System, Octal Number System, Hexadecimal Number System, Fractional Conversion, Coding System

Data Representation and Binary Arithmetic - Introduction, Bits, Nibbles, Bytes and Words, Data Representation, Coding system, Binary Arithmetic, Binary Addition, Binary Subtraction, Binary Multiplication, Binary Division, Character Representation, Checking the Result of Binary Arithmetic.

Unit 2: Input Devices - Introduction, Input Device, Typing Input Devices, Pointing Input Devices, Scanning Input Devices, Audio Visual Input Devices

Output Devices - Introduction, Output Devices, Soft Copy Vs Hard Copy Output, Monitor, Printers, Plotter, Electrostatic Technique, Special Purpose Output Equipment

Central Processing Unit - Introduction, what is Central Processing Unit, Arithmetic and Logic Unit, Control Unit, Registers, Instruction set, Processor Speed Storage Devices - Introduction, Storage and its needs, Brain Vs Memory, Storage Evaluation Units, Data Access Methods, Primary Storage, Secondary Storage, Hard Disk Operations, Floppy Disk Drives, Winchester Disk, Optical Disk, VCD, CD-R, CD-RW, DVD, Zip Drive, Flash Drives, Blue Ray Disk, Memory Card, Driving Naming Conventions In a PC

Basics of Software- Introduction, What Does Software Stand For? Needs of software, Types of software, Open-Source Software, Integrated Development Environment

Operating System - Introduction, Operating System, Why an Operating System, Functions of Operating System, The Booting Process, Types of Reboots, Booting from Different Operating System, Types of Operating System, Some Prominent Operating Systems

Disk Operating System - Introduction, what is DOS? Functions of DOS, Versions of DOS, DOS Commands, Important Internal Commands of DOS, Important External Commands of dos, Executable Vs Non-Executable Files in Dos.

Unit 3: Programming Languages, Introduction, Data, information And Knowledge, Characteristics of Information, Comparison between human language and, Computer Language, what is a program? What is a Programming language? Programming development cycle, Algorithm, Program Flowcharts, Pseudo code, Programming approaches, Programming Paradigms, Types of Programming Language, Third Generation Language, Fourth Generation Language

Computer Virus - Introduction, Virus, History, Mechanism of virus, How A Virus Spreads, how is virus named, A few Prominent Viruses, Types of Computer Virus, Related Concepts: Anti-Virus Programs, Norton Anti - Virus (NAV), Execution of Norton Anti-Virus

Communication and IT - Introduction, Computer Network, Communication Process, Communication Types, Transmission Media, Wireless Media, Communication Channels/Media, Modem, Characteristics of a Modem, Types of Modems.

Unit 4: Networks - Introduction, Internet Vs Intranet, Types of Networks, Topology, Types of Connectivity, Network Devices

Internet - Introduction, what is Internet actually? Growth of Internet, Owner of the Internet, Internet Service Provider, Anatomy of Internet, ARPANET and Internet history of the World Wide Web, Services Available on Internet (Internet Tools), Basic internet terminologies, net etiquette, Application of internet.

Unit 5: Applications of Computers and Information Technology - Introduction, Business and Computer, E-Mail, E-Commerce, Project management, Computers in Personnel Administration, Accounting, Computers in Cost and Budget Control, Marketing, Manufacturing, Materials management, Banking, Insurance and Stock broking, Purchasing, Computers in warehousing

Reference Books

1. Computer Fundamentals – BPB Publications, Pradeep k Sinha, Priti Sinha.
2. Fundamentals of Computers – PHI Publication, V. Rajaraman, Neeharika Adabala.
3. Fundamentals of Computers – McGraw Hill Education Balagurusamy.

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
1	Understand the fundamentals of computers. Recognize different types and generations of computers. Understand different number systems and data representation.	- Lectures on the introduction to computers, their strengths and limitations. - Discussions on personal computers, their components, and evolution. - Interactive sessions on number systems, binary arithmetic, and data representation.	Computer Fundamentals Quiz: Assess understanding. Personal Computer Analysis. Analyze PC architecture. Number Systems and Data Representation.
2	Learn about various input and output devices. Understand the Central Processing Unit and storage devices. Explore the basics of software and operating systems. Learn about Disk Operating System (DOS) and its functions.	- Lectures on input and output devices, including typing, pointing, and scanning devices. - Discussions on the CPU, registers, storage needs, and data access methods. - Interactive sessions on software types, development, paradigms, and operating systems. - Lectures on DOS, its functions, versions, commands, and file types.	Input and Output Devices Quiz: Assess knowledge. CPU and Storage Devices Analysis: Analyze CPU and storage. Software and OS Examination: Assess understanding. DOS Knowledge Test: Assess DOS understanding.
3	Understand programming languages and their characteristics. Learn about computer viruses, their history, and mechanisms. Explore computer networks, communication, and transmission media.	- Lectures on programming languages, development cycle, and paradigms. - Discussions on computer viruses, how they spread, and anti-virus programs. - Interactive sessions on computer networks, communication types, and media.	Programming Languages Quiz: Assess knowledge. Virus Analysis and Protection: Analyze virus concepts. Network and Communication Exercise: Practice concepts.
4	Understand the basics of networks and internet technology. Learn about the growth and services of the Internet.	- Lectures on network types, topology, connectivity, and internet fundamentals. - Discussions on the Internet's growth, services, and basic terminologies.	Networks and Internet Quiz: Assess understanding. Internet Analysis and Terminology: Analyze Internet concepts.
5	Learn about the applications of computers and IT in business.	- Lectures on the applications of computers in various business sectors.	Business Applications Assessment: Evaluate knowledge.

Ability Enhancement Course (AEC)**3HHLA101: हिन्दी आधार/पाठ्यक्रम, हिन्दी भाषा और संरचना-1**

(Credit: Theory -2 Tutorial - 0)

Scheme of Examination

Course Code	Course Name	Credit	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical		Total	Theory	Practical
			End	Mid	Assign	End	Term Sem			
3HHLA101	हिन्दी भाषा और संरचना	2(2-0-0)	60	30	10	-	-	100	2 hr	-

पाठ्यक्रम के उद्देश्य:

- विद्यार्थियों में राष्ट्र प्रेम की भावना का विकास करना।
- हिन्दी के समृद्ध साहित्य को नयी पीढ़ी तक पहुँचाना।
- पत्र-लेखन, सार लेखन, भाव पल्लवन एवं साक्षात्कार के कौशल का विकास करना।
- डायरी, संस्मरण, लेखन, पारिभाषिक, शब्दावली, तत्सम, तद्भव, देशज, विदेशी शब्दों इत्यादि के ज्ञानका परिमार्जन करना।

अपेक्षित परिणाम:

- विद्यार्थी भारत भूमि से प्रेम व स्नेह के भावों को बढ़ा सकेंगे।
- विद्यार्थियों की हिन्दी की भाष्य संपदा में वृद्धि होगी।
- पत्र-लेखन, सार लेखन, भाव पल्लवन साक्षात्कार के कौशल का विकास होगा।
- डायरी एवं संस्मरण लेखन विद्या का परिमार्जन होगा।
- हिन्दी के समृद्ध साहित्य कोष से लाभान्वित होंगे।

पाठ्यक्रम:

इकाई – 1 भारत वंदना (काव्य) सूर्यकांत त्रिपाठी निराला, जाग तुझको दूर जाना सुश्री महादेवी वर्मा, स्वतंत्रता पुकारती (काव्य) जयशंकर प्रसाद, हम अनिकेतन (काव्य), बालकृष्ण शर्मा नवीन, भाषा की महत्ता और उसके विविध रूप, भाषा-कौशल

इकाई – 2 करुणा (निबंध) आचार्य रामचन्द्र शुक्ल, समन्वय की प्रक्रिया (निबंध) रामधारी सिंह 'दिनकर' बिच्छी बुआ (कहानी) डॉ. लक्ष्मण बिष्ट 'बटरोही', अनुवाद परिभाषा प्रकार, महत्व, विशेषताएं, हिन्दी की शब्द-संपदा, पारिभाषिक शब्दावली

इकाई – 3 विलायत पहुंच ही गया (आत्मकथांश) महात्मा गांधी, अफसर (व्यंग्य) शरद जोषी, तीर्थयात्री (कहानी) डॉ. मिथिलेश कुमार मिश्र, मकड़ी का जाला (व्यंग्य) डॉ. रामप्रकाश सक्सेना वाक्य- संरचना :तत्सम, तद्भव देशज विदेशी

इकाई – 4 अप्प दीपो भव (वक्तृत्व कला) स्वामी श्रद्धानंद, भारत का सामाजिक व्यक्तित्व (प्रस्तावना) जवाहरलाल नेहरू, पत्र मैसूर के महाराजा को (पत्र-लेखन) स्वामी विवेकानंद, बनी रहेंगी किताबें

(आलेख) डॉ. सुनीता रानी घोष, पत्र-लेखन: महत्व और उसके विविध रूप, सड़क पर दौड़ते ईहा मगू (निबंध) डॉ. श्यामसुन्दर दुबे

इकाई – 5 योग की शक्ति (डायरी) डॉ. हरिवंशराय बच्चन, कोष के अखाड़े में कोई पहलवान नहीं उतरता (साक्षात्कार) – भाषाविद् डॉ. हरिदेव, बाहरी से प्रो. – त्रिभुवननाथ शुक्ल, नीग्रो सैनिक से भेंट (यात्री-संस्मरण) डॉ. देवेन्द्र सत्यार्थी, यदि "बा" न होती तो शायद गांधी को यह ऊँचाई न मिलती (साक्षात्कार) कथाकार- गिरिराज किशोर से सत्येन्द्र शर्मा सार –लेखन, भाव-पल्लवन साक्षात्कार और कौशल

संदर्भ पुस्तक:

कथा साहित्य आईसेक्ट ग्रुप प्रकाशन

पाठ्यक्रम सीखने के उद्देश्यों की प्राप्ति को सुगम बनाना

इकाई	पाठ्यक्रम के उद्देश्य	पाठ्यक्रम से प्राप्त लाभ	शिक्षण और सीखने की गतिविधियां	मूल्यांकन कार्य
1	● भाषा के विभिन्न रूपों का अध्ययन और जानकारी प्राप्त करना।	● भाषा के विभिन्न रूपों के साथ परिचित होना।	● भाषाओं के रूपों का प्रदर्शन और उनके विशेषताओं का अध्ययन करना। भाषाओं के रूपों का प्रदर्शन और उनके विशेषताओं का अध्ययन करना।	● भाषाओं के रूपों का प्रदर्शन और उनके विशेषताओं की व्याख्या करना।
2	● तत्सम और तद्भव शब्दों के अंतर को समझना।	● तत्सम और तद्भव शब्दों के बीच का अंतर समझना।	● तत्सम और तद्भव शब्दों के उदाहरण प्रदान करना और समझाना।	● तत्सम और तद्भव शब्दों के अंतर को समझाने के लिए उपाय करना।
3	● नाटक और गद्य साहित्य के बारे में जानकारी प्राप्त करना।	● नाटक और गद्य साहित्य के अध्ययन से साहित्यिक ज्ञान में वृद्धि होना।	● नाटक और गद्य साहित्य के उदाहरण और उनके विशेषताओं का अध्ययन करना।	● नाटक और गद्य साहित्य के उदाहरणों की व्याख्या करना।
4	● रस, अलंकार, दोहा, सोरठा आदि के बारे में जानकारी प्राप्त करना।	● साहित्यिक उपकरणों के ज्ञान से भाषा का सुधार होना।	● विभिन्न रसों, अलंकारों, दोहों, सोरठों इत्यादि के उदाहरण और व्याख्या करना।	● रस, अलंकार, दोहा, सोरठा आदि के उदाहरणों की व्याख्या करना।
5	● काव्यांग विवेचन, रस, छंद, अलंकार, उपमा, रूपक, दोहा, सोरठा, चौपाई आदि	● साहित्यिक उपकरणों के ज्ञान से भाषा का सुधार होना।	● विभिन्न काव्यांग और साहित्यिक उपकरणों के उदाहरण और व्याख्या करना।	● काव्यांग और साहित्यिक उपकरणों के उदाहरणों की व्याख्या करना।

इकाई	पाठ्यक्रम के उद्देश्य	पाठ्यक्रम से प्राप्त लाभ	शिक्षण और सीखने की गतिविधियां	मूल्यांकन कार्य
	के बारे में जानकारी प्राप्त करना।			

Inter Disciplinary Course**3ISTI102 Statistics**

(Credit: Theory -3 Tutorial - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work			
3ISTI102	Statistics	3-0-0	60	30	10	-	-	100	3 hr	-

Course Objective

- To understand the concept of population and sample.
- To use frequency distribution to make decision.
- To understand and to calculate various types of averages and variation.

Course Outcomes

Students will be able to apply statistical methods to analyze and interpret data effectively. They will demonstrate proficiency in calculating measures of central tendency and dispersion, as well as understanding the types and methods of correlation and regression analysis. Students will be equipped to use probability distributions, such as normal, binomial, and Poisson, to solve real-world problems.

Syllabus

Unit 1: Statistical Methods: Definition and scope of Statistics, Concepts of statistical population and sample. Data: Quantitative and Qualitative, Attributes, variables, scales of measurement nominal, ordinal, interval and ratio. Presentation: Tabular and Graphical, including histogram.

Unit 2: Measures of Central Tendency: Mathematical and Positional. Measures of Dispersion Range, Quartile deviation, mean deviation, standard deviation, coefficient of variation, Moments, absolute moments, factorial moments, skewness and kurtosis, Sheppard's corrections.

Unit 3: Correlation- Meaning, Definitions, Types of correlation, Methods of correlation Regression- Meaning, Uses, Difference between Correlation and Regression

Unit 4: Probability Distributions: -Definition, probability distribution, Normal distribution, Binomial distribution, Poisson distribution

Unit 5: Index Number- Meaning, Characteristics, Importance and Uses, Construction of Index number, Fisher's Ideal Index Number.

Reference Books

- S.P. Gupta, Statistical Methods, Sultan Chand and sons New Delhi
- Dr. BS.M. Shukla Principles of Statistics Sahitya Bhawan Publication Agra.
- Fundamentals of Statistics" by S.C. Gupta
- Advanced Statistics by Dr. Aalok Gupta
- Statistical Methods" by S.P. Gupta

Inter Disciplinary Course**3ICNI102 Computer Network**

(Credit: Theory -3 Tutorial - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assign.	End Sem	Term work			
3ICNI102	Computer Network	3-0-0	60	30	10	-	-	100	3 hr	-

Course Objective

- Understand the fundamentals of network applications, hardware, and software.
- Familiarize with reference models like OSI, TCP/IP, and the Internet.
- Learn about connection-oriented networks like X.25 and frame relay.

Course Outcomes

- After Study this student will able to know about the concepts of computer network and internet. concepts of LAN, CAN and MAN and WAN.

Syllabus

Unit 1: INTRODUCTION: Network applications, network hardware, network software, reference models: OSI, TCP/IP, Internet, Connection-oriented network-X.25, frame relay.

THE PHYSICAL LAYER: Theoretical basis for communication, guided transmission media, wireless transmission, the public switched telephone networks, mobile telephone system.

Unit 2: THE DATA LINK LAYER: Design issues, error detection and correction, elementary data link protocols, sliding window protocols, example data link protocols - HDLC, the data link layer in the internet.
THE MEDIUM ACCESS SUB LAYER: Channel allocation problem, multiple access protocols, Ethernet, Data Link Layer switching, Wireless LAN, Broadband Wireless, Bluetooth

Unit 3: THE NETWORK LAYER: Network layer design issues, routing algorithms, Congestion control algorithms, Internetworking, the network layer in the internet (IPv4 and IPv6), Quality of Service.

Unit 4: THE TRANSPORT LAYER: Transport service, elements of transport protocol, Simple Transport Protocol, Internet transport layer protocols: UDP and TCP.

Unit 5: THE APPLICATION LAYER: Domain name system, electronic mail, World Wide Web: architectural overview, dynamic web document and HTTP.

APPLICATION LAYER PROTOCOLS: Simple Network Management Protocol, File Transfer Protocol, Simple Mail Transfer Protocol, Telnet.

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
1	<ul style="list-style-type: none"> • Understand network applications, hardware, and software. • Familiarity with reference models (OSI, TCP/IP, Internet). • Learn about connection-oriented networks (X.25, frame relay). • Comprehend the theoretical basis for communication. • Study public switched telephone networks and mobile systems. 	<ul style="list-style-type: none"> • Lectures on network applications, hardware, and software. • Discussions on OSI and TCP/IP models. • Case studies on X.25 and frame relay networks. • Labs to explore guided and wireless transmission media. • Guest lectures on public switched telephone networks and mobile systems. 	<ul style="list-style-type: none"> • Quiz on UNIT I concepts. • Assignment on reference models. • Exam on UNIT I content.
2	<ul style="list-style-type: none"> • Understand design issues in the data link layer. • Learn about error detection and correction. • Explore elementary data link protocols. • Study sliding window protocols and their applications. • Analyze data link layer in the internet and its components. 	<ul style="list-style-type: none"> • Lectures on data link layer design issues. • Problem solving sessions on error detection and correction techniques. • Lab exercises to implement elementary data link protocols. • Discussions and examples of sliding window protocols. • Guest lectures on data link layer in the internet. 	<ul style="list-style-type: none"> • Quiz on UNIT II concepts. • Assignment on data link protocols. • Exam on UNIT II content.
3	<ul style="list-style-type: none"> • Gain insights into network layer design issues. • Learn about routing algorithms and congestion control. • Understand internetworking and IPv4/IPv6. • Explore Quality of Service (QoS) in networking. 	<ul style="list-style-type: none"> • Lectures on network layer design issues and routing algorithms. • Simulations and practical exercises for routing algorithms. • Workshops on internetworking and IPv4/IPv6. • Case studies and discussions on QoS. 	<ul style="list-style-type: none"> • Quiz on UNIT III concepts. • Assignment on network layer. • Exam on UNIT III content.
4	<ul style="list-style-type: none"> • Study transport service and elements of transport protocols. 	<ul style="list-style-type: none"> • Lectures and practical examples of transport protocols. 	<ul style="list-style-type: none"> • Quiz on UNIT IV concepts.

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
	<ul style="list-style-type: none"> Learn about Simple Transport Protocol, UDP, and TCP. 	<ul style="list-style-type: none"> Handson labs to implement UDP and TCP. 	<ul style="list-style-type: none"> Assignment on transport layer.
5	<ul style="list-style-type: none"> Understand the Domain Name System (DNS) and its role. Explore electronic mail and the World Wide Web architecture. Study application layer protocols (SNMP, FTP, SMTP, Telnet). 	<ul style="list-style-type: none"> Lectures and handson exercises on DNS. Discussions on email systems and HTTP. Handson labs to work with SNMP, FTP, SMTP, and Telnet. 	<ul style="list-style-type: none"> Quiz on UNITV concepts. Assignment on application layer. Final Exam covering UNITV.

Reference Books

- Behrouz A. Forouzan (2006), Data Communication and Networking, 4th Edition, McGraw Hill, India.
- Kurose, Ross (2010), Computer Networking: A TopDown Approach, Pearson Education, India.

Value added course (VAC)
3SEEV106: Environmental Education
 (Credit: Theory -3 Tutorial - 0)
Scheme of Examination

Course Code	Course Name	Credit	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assign	End Sem	Term Sem			
3SEEV106	Environmental Education	3(3+0)	60	30	10	-	-	100	3	-

Course Objective

Upon completion of the course, the student – teacher will be able to:

- Understand the concept, significance, scope and terminologies objectives and program of environmental education.
- Develop awareness about the various types of pollution ecological Imbalances and life and contributions of environmental activities.
- Interpret the environmental legislations in conservation and protection of the environment.
- Understand the role of governmental and non-governmental Agencies in environmental education.
- Apply the methods of teaching and evaluation in environmental education.

Course Outcomes

- Environmental education raises awareness about various environmental issues, such as pollution, climate change, habitat destruction, and resource depletion.
- It provides individuals with a deeper understanding of ecosystems, biodiversity, and the interconnections between living organisms and their environment.
- Environmental education can lead to changes in behavior, such as reducing waste, conserving energy and water, and adopting sustainable consumption patterns.
- Individuals become more inclined to take care of their surroundings, leading to increased community involvement in local environmental projects.
- It fosters critical thinking skills by encouraging individuals to analyze complex environmental problems and develop solutions.

Syllabus:

Unit 1: Introduction to Environmental Education-

Environmental Education Concept, Importance and Scope, Objectives and Principles of Environmental Education. Basic Concepts in Environmental Education, Ecology, Eco-System, Food Chain, Natural Resources, Greenhouse Effect, Bio-Diversity.

Unit 2: Environment and Pollution-

Definition and Types of Environmental pollution, Air Pollution- Definition, Causes and Remedial Measures, Water Pollution: Definition, Causes and Remedial Measures, Soil Pollution: Definition, Causes and Remedial Measures, Sound Pollution: Definition, Causes and Remedial Measures, Ecological Imbalances -Deforestation, Soil Erosion.

Unit 3: Environmental Laws and Organization-

The Air Prevention and Control of Pollution Act 1977, The Water Prevention and Control of Pollution Act 1974, Forest Conservation Act 1980, Environment Protection Act 1986, United Nations Environment Program (UNEP), International Union for Conservation of Nature and Natural Resources (IUCN), Central pollution control board (CPCB)

Unit 4: Environmental Ethics-

Role of Indian and other religions and cultures in environmental conservation. Green Politics, Earth Hour, Green Option Technologies, Environmental communication and public awareness, EIA Formulations, stages, Merits and demerits

Unit 5: Methods of Teaching Environmental Education-

Project Work, Intellectual Meets-Seminars, Symposia, Workshops, Conferences, Group Discussions, Debates, Brain Storming Quiz, Poster Making, Models Making and Exhibitions.

Text Book

- Environmental science by Kamal Kant Joshi & Deepak Kumar, TechSar. 2019
- Basics of Environmental science by Abhijit Mitra & Tanmay Ray Chaudhuri, New central book agency Pvt. Ltd. 2017
- Essentials of Environmental Education by A.B. Saxena & V.V. Anand, Motilal Banarsidass publishing House, 2012
- Environmental Studies by Dr. SM Saxena, Dr. Seema Mohan.

Reference Books

- Ecological Literacy: Educating Our Children for a Sustainable World, Michael K. Stone and Zenobia Barlow, Publication: Published by Sierra Club Books in 2005.
- Place-Based Education: Connecting Classrooms and Communities, David Sobel Publication: Published by The Orion Society in 2005.

The Handbook of Environmental Education, Robert B. Stevenson, Michael Brody, Justin Dillon, and Arjen E.J. Wals, Publication: Published by Routledge in 2019.

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
1	<ul style="list-style-type: none"> ● The students about this particular are intended to gain knowledge about the objectives and importance of environmental education. Enabling them to understand the composition of environment, greenhouse 	<ul style="list-style-type: none"> ● Lectures ● Group discussion ● Sight Seeing ● The teaching will be done through lectures and group discussion 	<ul style="list-style-type: none"> ● Assignment ● Homework

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
	effect. Students will improve their understanding towards the factors governing the life on earth		
2	<ul style="list-style-type: none"> • Students will understand about the impacts of our unusual and destructive use of resources and their harmful effects. • Students will understand about the destruction of environment and its sustainability. Enhance the concern about this depletion among the students. 	<ul style="list-style-type: none"> • Lectures • Group discussion • Visit to any industry or manufacturing site • The teaching will be done through lectures and group discussion. 	<ul style="list-style-type: none"> • Assignment • Poster making
3	<ul style="list-style-type: none"> • Importance of government laws and agencies their interference to regulate pollution and harming the quality of environment. Methods to improve the quality of habitat and natural resources will be necessary for the students to have its knowledge and concern. 	<ul style="list-style-type: none"> • Lectures • Group discussion • Visit to any law governing body • The teaching will be done through lectures and group discussion 	<ul style="list-style-type: none"> • Brain storming Quiz • Assignment
4	<ul style="list-style-type: none"> • Students will know their ethics and responsibilities towards the improvement in quality of environment. Innovation, technologies, awareness through communication and various others measures through which a student can involve these practices in their lifestyle. 	<ul style="list-style-type: none"> • Lectures • Group discussion • The teaching will be done through lectures and group discussion 	<ul style="list-style-type: none"> • Seminar • Conferences
5	<ul style="list-style-type: none"> • The students will enhance the techniques to prepare a project on any of the issues regarding the environment pollution or the remedial measures. Students will find ways to present the issues through seminars, workshops, poster making, model making. 	<ul style="list-style-type: none"> • Lectures • Group discussion • Seminars • Poster making • Model making 	<ul style="list-style-type: none"> • Power point presentation • Project work • Debates • Brain storming Quiz

Value added course (VAC)**3IFAV106 Fundamentals of AI**

(Credit: Theory -3 Tutorial - 0)

Scheme of Examination

Course Code	Course Name	Credit	Maximum marks Allotted					Duration of Exam.		
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assign	End Sem	Term Sem			
3IFAV106	Fundamentals of AI	(3-0-0)	60	30	10	-	-	100	3 hr	-

Course Objective

Student will be able-

- To understanding the importance of AI and puzzle problem.
- To understanding the Search Techniques.
- To understanding the Symbolic and Statistical Reasoning.
- To understanding the frames and Structural Knowledge Representation.
- To understanding the expert system life cycle.

Course Outcomes

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

- Understand the fundamental concepts and scope of Artificial Intelligence.
- Describe the essential tools and techniques used in Machine Learning.
- Describe interface mechanisms and their role in knowledge representation.
- Understand the fundamentals of probability theory and its role in AI.

Syllabus**Theory**

Unit 1: Introduction: 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, 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.

Unit 2: Introduction to Machine Learning: Applications of ML, Difference between Data Mining and Predictive Analysis, Tools and Techniques of Machine Learning. What is Machine Learning, Basic Terminologies of Machine Learning

Unit 3: Knowledge Representations First order predicate calculus, Skolemization, resolution principle and unification, interface mechanisms, horn's clauses, semantic networks, frame systems and value inheritance, scripts, conceptual dependency.

Unit 4: Natural Language processing Parsing techniques, context free grammar, recursive transitions nets (RNT), augmented transition nets (ATN), case and logic grammars, semantic analysis. Game playing Minimax search procedure, alpha-beta cut offs, additional refinements. Planning Overview an example domain the block word, component of planning systems, goal stack planning, nonlinear planning.

Unit 5: Probabilistic Reasoning and Uncertainty Probability theory, bayes theorem and Bayesian networks, certainty factor. Expert Systems Introduction to expert system and application of expert systems, various expert system shells, vidwan frame work, Knowledge acquisition, case studies, MYCIN. Learning Rote learning, learning by induction, explanation-based learning

Reference Books

- Elaine Rich and Kevin Knight, "Artificial Intelligence," Tata McGraw-Hill. "Artificial Intelligence," 4th Edition, Pearson.
- Dan W. Patterson, "Introduction to Artificial Intelligence and Expert Systems," Prentice India.
- Nils J. Nilson, "Principles of Artificial Intelligence," Narosa Publishing House. Clocksin and C.S. Melish, "Programming in PROLOG," Narosa Publishing House.
- M. Sasikumar, S. Raman, etc., "Rule-based Expert System," Narosa Publishing House.

Facilitating the Achievement of Course Learning Objectives

Unit	Course learning outcomes	Teaching and learning activities	Assessment tasks
1	<ul style="list-style-type: none"> • Understand the fundamentals of Artificial Intelligence (AI). Identify AI problems and techniques. Learn about the levels of AI models and criteria for success. Explore state space search and production systems. • Understand search algorithms such as BFS, DFS, and heuristic search techniques. • Learn about problem reduction and constraint satisfaction. Familiarize with means ends analysis. 	<ul style="list-style-type: none"> • Problem solving exercises on state space search. • Group discussions on production systems and search algorithms. • Handson exercises with BFS, DFS, and heuristic search algorithms. Case studies on problem reduction and constraint satisfaction. Practical demonstrations of means end analysis. 	<ul style="list-style-type: none"> • Quizzes on AI fundamentals and problem-solving techniques. Written assignments on state space search. Group presentation on search algorithms.
2	<ul style="list-style-type: none"> • Learn about applications of Machine Learning (ML). Differentiate between Data Mining and Predictive Analysis. Explore tools and techniques of Machine Learning. Understand basic ML terminologies. 	<ul style="list-style-type: none"> • Lectures on ML applications and differences from Data Mining. Discussions on ML tools and techniques. Handson experience with ML terminology. Case studies on real world ML applications. 	<ul style="list-style-type: none"> • Written assignments on ML applications and differences from Data Mining. Quizzes on ML tools and terminology. Case study analysis of ML applications. Final examination on unit II content.
3	<ul style="list-style-type: none"> • Understand knowledge representations in AI. Learn about first order predicate calculus, Skolemization, and resolution principles. Explore 	<ul style="list-style-type: none"> • Practical exercises on resolution principles and interface mechanisms. Group discussions on semantic networks, frame 	<ul style="list-style-type: none"> • Problem solving assignments on predicate calculus and resolution. Quizzes on knowledge representation techniques.

Unit	Course learning outcomes	Teaching and learning activities	Assessment tasks
	interface mechanisms, horn's clauses, semantic networks, frame systems, and value inheritance. Familiarize with scripts and conceptual dependency.	systems, and scripts. Handson sessions with conceptual dependency. Case studies on real world knowledge representation systems.	
4	<ul style="list-style-type: none"> Explore Natural Language Processing (NLP) techniques. Learn about parsing techniques, context free grammar, and semantic analysis. Understand gameplaying strategies and Minimax search procedures. Familiarize with alpha beta cutoffs and planning components. Apply these concepts to an example domain. 	<ul style="list-style-type: none"> Lectures on NLP, parsing techniques, and semantic analysis. Coding practice for parsing and grammar. Problem solving exercises on gameplaying strategies. Practical demonstrations of planning components. Handson sessions with an example domain. 	<ul style="list-style-type: none"> Implementation and testing of parsing and semantic analysis. Problem solving assignments on game playing and planning. Quizzes on NLP and gameplaying concepts. GD, Unit Test and Quizzes.
5	<ul style="list-style-type: none"> Understand probabilistic reasoning and uncertainty in AI. Learn about probability theory, Bayes' theorem, and Bayesian networks. Explore certainty factors and expert systems. Familiarize with expert system shells, knowledge acquisition, and case studies. Learn about various learning techniques, including rote learning, induction, and explanation-based learning. 	<ul style="list-style-type: none"> Bayesian networks, and expert systems. Practical exercises on probability theory and Bayes' theorem. Group discussions on expert systems and knowledge acquisition. Handson sessions with learning techniques. Case studies on expert systems and learning methods. 	<ul style="list-style-type: none"> Problem solving assignments on probability theory and expert systems. Quizzes on probabilistic reasoning and learning techniques. Group presentation on expert system case studies.

SKILL ENHANCEMENT COURSES**3IWDS105: Web Development I– (HTML and CSS)**

(Credit: Theory -0 Tutorial – 1 Practical- 1)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assign.	End Sem	Term work			
3IWDS105	Web Development I – (HTML and CSS)	0-1-1	-	-	-	60	40	100	-	2h

Course Objective

- Understand web fundamentals: Learn about the Internet, Intranet, World Wide Web, web browsers, and search engines.
- Master HTML essentials: Comprehend HTML syntax, elements, basic tags, formatting, lists, images, hyperlinks, tables, iframes, and forms.
- Proficiently use CSS: Gain expertise in CSS syntax, selectors, colors, backgrounds, borders, margins, padding, text, and layout.
- Enhance interactivity: Utilize CSS pseudo classes, pseudo elements, animations, and transitions to create dynamic web pages.
- Create responsive designs: Learn to implement CSS media queries for designing websites that adapt to different devices and screen sizes.
- Build web pages: Apply knowledge of HTML and CSS together to construct visually appealing and interactive web content.

Course Outcomes

- Demonstrate a solid understanding of web technologies and their components.
- Create well-structured and visually appealing web pages using HTML and CSS.
- Implement various CSS properties to style and layout web content effectively.
- Develop interactive web pages using CSS for enhanced user experience.
- Design responsive websites that adjust to different devices and screen resolutions.
- Successfully build basic web pages independently, adhering to web standards and best practices.

Syllabus**Theory**

Unit 1: ABC of Web Introduction, what is Internet Actually? Internet Vs Intranet., World Wide, Browser, Search Engine Web (WWW), URL (web address), Domain Name System (DNS), search engines, Working of Web Site.

Unit 2: Hyper Text Markup Language Introduction, Concept of Hyper Text Markup Language, Versions of HTML, HTML Editors, Elements of HTML, HTML Basic Tag, HTML Formatting Tag, HTML Color Coding,

HTML List Ordered List, Unordered List, Description List, HTML Images, HTML Hyperlink, HTML Marquee Tag

Unit 3: HTML Table, HTML Iframe, HTML Form, Attributes, HTML Form Elements, HTML Input Types, HTML Input Attributes, HTML Input Form Attributes, HTML Block and Inline Element, HTML Div Tag, HTML Span Tag, HTML Media Tags, HTML Semantic Tags, HTML Meta Tag

Unit 4: Introduction To CSS, Introduction, CSS Syntax, Types of CSS, CSS Basic Selectors, how to add CSS, CSS Comments, CSS Colors, CSS Background, CSS Gradient, CSS Borders, CSS Margins, CSS Padding, CSS Text, CSS Font, CSS List, CSS Table,

Unit 5: CSS Display, CSS Selectors, CSS Pseudo classes, CSS Pseudo elements, CSS Icons, CSS Position, CSS index, CSS Animation, CSS Media Query.

Practical list

- Write HTML code to create a basic webpage with a heading, paragraph, and an image.
- Develop an HTML page containing an ordered list of your top three hobbies and an unordered list of your favorite books.
- Design a registration form with fields for name, email, password, and a checkbox for agreeing to terms. Style the form using CSS to improve its visual appeal.
- Embed a map using the <iframe> tags.
- Implement a navigation bar with multiple links. Use CSS pseudo classes like: hover to create interactive effects, such as changing the background color of a link when the mouse hovers over it.
- Create a responsive HTML webpage introducing yourself. Include sections such as personal information, your photo, hobbies, and a list of your skills.
- Apply CSS pseudo classes like: hover and: active to create interactive effects for buttons or links on your webpage.
- Integrate CSS icons into your webpage using popular icon libraries such as Font Awesome or Material Icons
- Create a CSS animation that smoothly transforms an element's size or position. Experiment with different animation properties

Reference books

- HTML and CSS: Design and Build Websites, by Jon Duckett
- Level Made Simple – Web Designing and Publishing
- HTML and CSS: THE COMPLETE REFERENCE by Thomas Powell
- Level Made Simple – Web Designing and Publishing –वेब डिजाइनिंग और पब्लिशिंग (M2R5) by Prof. Satish Jain and M. Geetha Iyer

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
1	<ul style="list-style-type: none"> • Understand the basics of the internet • Differentiate between Internet and Intranet • Explain the concept of World Wide Web (WWW) 	<ul style="list-style-type: none"> • Lecture on the history and structure of the internet, using video animated know the Concept of How Internet Works Using Animated. • Discussion on the role of browsers and search engines 	<ul style="list-style-type: none"> • Quiz on internet concepts. • Make presentation on How website works.

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
	<ul style="list-style-type: none"> Familiarize with URL and DNS Understand the working of websites 	<ul style="list-style-type: none"> Handson activity to access websites and URLs 	
2	<ul style="list-style-type: none"> Understand the basics of HTML Familiarize with HTML elements and tags Learn about HTML formatting and colors Explore HTML lists, images, and hyperlinks 	<ul style="list-style-type: none"> Lecture on HTML fundamentals and versions Demonstration of various HTML editors Practice creating HTML documents with different tags and attributes 	<ul style="list-style-type: none"> Handson HTML coding exercises HTML coding project
3	<ul style="list-style-type: none"> Learn about HTML tables, forms, and attributes Understand HTML input types and form elements Explore block and inline elements in HTML Familiarize with HTML media and meta tags 	<ul style="list-style-type: none"> Lecture on HTML tables, forms, and attributes Handson practice with HTML forms and input types Group discussion on semantic HTML tags 	<ul style="list-style-type: none"> HTML form creation and submission HTML table creation HTML coding assignment
4	<ul style="list-style-type: none"> Understand the basics of CSS Learn about CSS selectors and comments Explore CSS properties for colors, backgrounds, borders, margins, and padding Understand CSS for text, fonts, lists, tables 	<ul style="list-style-type: none"> Lecture on CSS syntax and types Demonstration of adding CSS to HTML documents Handson CSS coding practice with basic CSS properties 	<ul style="list-style-type: none"> CSS styling exercise CSS design project Card Design
5	<ul style="list-style-type: none"> Learn about CSS display, selectors, and pseudo classes and pseudo elements Explore CSS icons, positioning, and animation 	<ul style="list-style-type: none"> Lecture on advanced CSS properties Interactive sessions for practicing CSS selectors and pseudo elements Demonstration of CSS animations and media queries 	<ul style="list-style-type: none"> Basic Layout Design using CSS. CSS positioning and animation task. CSS media query implementation. Final project (portfolio create and design) with responsive design.



BACHLOR OF COMPUTER APPLICATION

II Semester

(BCA – CKUG02A03)

(Effective from Academic Year 2023-24)

MAJOR CORE**Object Oriented Programming with C++-3IOPC203**

Credit: Theory -3 Tutorial - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.		
			Theory			Practical			Total End Sem	Theory Mid Sem	Practical Assignment
			End Sem	Mid Sem	Assignment	End Sem	Term work				
3IOPC203	Object-oriented Programming with C++	4-0-2	60	20	20	60	40	200	3 hr	2hr	

Course objective

Student will be able

1. To understand the basic knowledge of oop with C++ language.
2. To understand the structure and classes concepts, data member.
3. To understand the Array, Pointer's operations.
4. To understand the Function overloading and Operator Overloading.
5. To understand the Inheritance and C++I/O system.

Course outcomes

Students will develop a comprehensive understanding of object-oriented programming, including inheritance and polymorphism. They will be proficient in creating and running C++ programs, working with arrays, functions, pointers, and dynamic memory allocation. Additionally, students will master the implementation of classes and objects, operator overloading, virtual functions, and the C++ I/O system for formatted input and output.

Syllabus**Theory**

Unit-I Overview of C++ - Overview of C++, Software crisis, Object oriented programming paradigm, Basic concepts of OOP, Advantages/Benefits of OOP, Usage/applications of OOP

C++ Environment, Program development environment, The language and the C++ language standards, Introduction to various C++ compilers, The C++ standard library, Prototype of main () function, i/o operator, manipulator, comments, data types

Creating and Compiling C++ Programs - TURBO C++ IDE, Creating, compiling and running a C++ program using idea and through command line, Elements of C++ Language, Structure of a C++ program, C++ tokens, Type conversion in expressions.

Decision Making and Branching - Introduction, Sequential statements, Mathematical Functions, branching statements, looping Statements, Nested loops, Programming examples.

Unit-II Arrays and Functions- Arrays, the meaning of an array, Single-dimensional arrays, Two-dimensional arrays (multi-dimensional arrays), User Defined Functions, Elements of user-defined

functions, return values and their types, Function calls, Categories of functions, Passing parameters to functions, Recursion, Command Line Arguments, Storage Class Specifiers.

Classes and Objects - Classes, Structures and classes, Unions and classes, Friend function, Friend classes, Inline function, Scope resolution operator, Static class members, Static data members, Static member functions, passing object to functions, Returning objects, Object assignment

Array, Pointers, References and the Dynamic Allocation Operators - Array of objects, Pointer to object, Type checking in C++, this pointer, Pointer to Derived Types, Pointer to class members, References, C++'s Dynamic Allocation Operators.

Constructors and Destructors - Introduction, Constructors, Default Constructor, Parameterized constructors, Copy Constructors, Multiple Constructors in a class, Constructors with default arguments, Default Arguments, Special Characteristics of Constructor functions, Destructors.

Unit-III Polymorphism – Introduction to polymorphism, Types of polymorphism, Function overloading, Overloading Constructor Function, Finding the address of an overloaded function, Operator Overloading, creating a Member Operator Function, Creating Prefix and Postfix forms of the increment (++) and decrement (--) operators (Overloading Unary Operator), Overloading the Shorthand Operators (i.e., +=, == etc.),

Operator Overloading Restriction (Rules), Operator Overloading using friend function, Overloading new and delete operator, Overloading some special operators, Overloading [](Subscripting) operator, Overloading() (Function Call) operator, Overloading Binary Arithmetic operators, Concatenating String, Overloading Comma (,) operator, Overloading the I/O operators.

Unit-IV Base class Access control, Inheritance and protected members, protected base class inheritance, inheriting multiple base classes, Constructors, destructors and Inheritance, when constructor and destructor function are executed, passing parameters to base class constructors, Granting access, Virtual base classes.

Virtual function, Pure Virtual functions, early Vs. late binding.

Unit-V The C++ I/O System Basics - The C++ I/O System basics, C++ predefined streams, Formatting using the ios members, Clearing Format Flags, An Overloaded form of setf(), Examining the Formatted Flags, Using width(), Using precision(), Using fill(), Using Manipulators to format I/O, Creating your own Manipulators

Outcomes – After Study This Student Will Be Able to Know About and Concepts of Oops with C++ Language, Classes. Student will be able to create Arrays its uses, Uses of function overloading, inheritance and C++/I/O system.

Practical

1. WAP to add, subtract, multiply and divide two numbers using concepts of C++.
2. WAP to show swapping of two numbers using C++.
3. WAP to calculate volume of cube, cylinder, rectangular box using three times function overloading in C++.
4. WAP using virtual function.
5. WAP using copy constructor.
6. WAP to show multiple inheritances.
7. WAP to find mean value of two numbers using friend function.
8. WAP using inline function.
9. WAP to demonstrate the use of Local Object, Static Object and Global Object using C ++.
10. WAP in C++ to demonstrate the creation and the use of dynamic object.
11. Derive the two classes son and daughter and, demonstrate polymorphism in action.

Reference books

1. A Complete Guide to Programming in C++ by Ulla Kirch-Prinz
2. Learn To Program with C++ by John Smiley
3. The C++ Programming Language by Bjarne Stroustrup
4. C++ Programming Language by Amir Khan

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
I	<ol style="list-style-type: none"> 1. Understand the overview of C++ and the software crisis 2. Comprehend the basic concepts and benefits of Object-Oriented Programming (OOP) 3. Familiarize with C++ environment and program development. 	<ol style="list-style-type: none"> 1. Lectures and discussions on C++ overview and OOP. 2. Hands-on practice using C++ IDE (e.g., TURBO C++) Code writing and compilation exercises 3. Problem-solving using decision-making and branching concepts 	<ol style="list-style-type: none"> 1. Quiz on C++ overview and OOP concepts 2. Programming assignments to create and compile C++ programs 3. Practical Test on decision making and branching in C++
II	<ol style="list-style-type: none"> 1. Understand arrays, functions, and their usage. 2. Learn about classes, objects, and their features 3. Familiarize with pointers, references, and dynamic allocation 	<ol style="list-style-type: none"> 1. Lectures and discussions on arrays, functions, classes, and pointers 2. Hands-on exercises involving arrays, functions, and classes 3. Code writing assignments for constructors and destructors. 	<ol style="list-style-type: none"> 1. Programming assignments on arrays, functions, and classes. 2. Practical exercises on pointers and dynamic allocation, constructors and destructors
III	<ol style="list-style-type: none"> 1. Explore the concept of polymorphism and its types 2. Learn function overloading and operator overloading 3. Understand how to use friend functions and special operators 	<ol style="list-style-type: none"> 1. Lectures on polymorphism and operator overloading 2. Code demonstrations for function overloading and operator overloading 3. Hands-on practice with friend functions and special operators 	<ol style="list-style-type: none"> 1. Programming assignments on function overloading and operator overloading 2. Quiz on polymorphism and operator overloading concepts 3. Coding assessment for implementing operator overloading
IV	<ol style="list-style-type: none"> 1. Understand access control in base classes 2. Learn about inheritance and its types 3. Explore constructors, destructors, and their role in inheritance 4. Comprehend virtual functions and their use 	<ol style="list-style-type: none"> 1. Lectures on access control, inheritance, and virtual functions 2. Code examples illustrating inheritance and constructors 3. Practical exercises for virtual functions 4. Class discussions on access control and constructors 	<ol style="list-style-type: none"> 1. Written test on access control and inheritance 2. Coding assessment for implementing inheritance and Virtual Functions.
V	<ol style="list-style-type: none"> 1. Learn the basics of the C++ I/O System 2. Explore predefined streams and formatting 3. Understand manipulators and create custom ones 	<ol style="list-style-type: none"> 1. Lectures on C++ I/O System basics and formatting 2. Practical exercises using predefined streams and manipulators 3. Hands-on practice with formatting and custom manipulators 	<ol style="list-style-type: none"> 1. Programming assignments on C++ I/O System and formatting, manipulators.

MINOR CORE**Windows and MS Office (word, power point, Excel, Outlook Express)-3IWMM204**

Credit: Theory -3 Tutorial - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work			
3IWMM204	Windows and MS Office (word, power point, Excel, Outlook Express)	4-0-2	60	20	20	60	40	200	3 hr	2hr

Course objective

1. To understand the basic knowledge of MS Windows.
2. To understand the Office Packages.
3. To understand the MS Excel.
4. To understand the MS PowerPoint and Outlook Express.

Course outcomes

5. After studying this student will be able to know about terms and concepts of Microsoft suite completely. (Like MS-word, power-point-excel sheets, outlook express)

Syllabus**Theory**

Unit-I Know the Windows 10, Introduction, what is Windows 10? Evolution of Windows Operating System, Features of Windows 10, What's New in Windows 10, Windows and Its Elements.

Accessories And Other Tools, Introduction, The Calculator, Using THE Calculator, The Character Map, Using Outlook Express, The Address Book, The Paint, The Notepad, The WordPad, The NetMeeting, The Internet Explorer, The Windows Media Player, The MS-DOS, The Control Panel, The Windows Picture and Fax Viewer, The HyperTerminal, The Windows Messenger, Using Windows Movie Maker. Managing Files and Folders, Introduction, viewing files and folders, arranging files and folders, creating a new folder, creating a file using short-cut.

Unit-II Customizing Your Computer: Introduction, customizing Your Desktop, Changing the Start menu style, setting a screen saver, reversing your mouse buttons, Changing the appearance of your mouse pointer, adding a new font to your computer, logging off from the computer, Adding or Removing Programs, Hiding and displaying quick launch bar. Introduction of Microsoft Office Suite

Office Task Panes, Introduction, The Task Pane, Displaying and Hiding a Task Pane, Types of Task pane, Additional Task Panes, Insert ClipArt Task Pane, Styles and Formatting Task Pane, Mail Merge Task Pane, Exercise.

Unit-III Word Processing and MS-Word: Introduction, Features of Word Processor, MS-WORD—a powerful word processor, Starting MS-Word, Chief Elements Of MS-Word Window, Displaying and Hiding the Toolbar, File operations in MS-WORD, Using Help Online, Customizing Office Assistant.

Text Formatting: Introduction, Typing the text, Selecting Text with a mouse, Deleting Text, Restoring the deleted text, typing over the existing text, Undoing/Cancelling the last action, Redoing/Repeating the last action, Formatting font, Advanced text formatting, Customizing Spelling Check, Using the thesaurus.

Document Formatting: Introduction, using page border, Bullets and numbering, Setting and removing tab stops, making word count, Using Auto text, Using autocorrects, Headers and Footers, setting up columns in the document, removing columns from the document, inserting page numbering, Formatting the page numbering, manual and automatic page breaks, setting margins, inserting date and time, Using Go to, Cursor movement with key-board.

Tables And Graphics: Introduction, creating tables, calculating numeric data in a table, deleting columns and rows, formatting a table, aligning text in the table, Formatting text in the table, applying borders and shadings, add a border to a table, automatically format a table, Using Drawing, creating a Shape, Using Word Art, Using Auto shapes, insert a clip from the Clip Organizer, inserting a text box, what is Drawing Canvas? Using auto shapes.

Mail Merge, Views, Template and Wizard, Introduction, Mail merge, Views, Overview of templates, creating a document template, create a Web page based on a template, modify a document template, RULER, ZOOM, PROTECTING YOUR DOCUMENT, INSERTING A FILE INTO ANOTHER, overview of wizard, Inserting Hyperlinks to a Web Page or a Word Document, EXERCISE.

Unit-IV Spreadsheet and MS-Excel: Introduction, Starting MS-Excel, Spreadsheet and its Elements, Application Window, Document Window, Cell, Standard Toolbar, Formatting Toolbar, Workbook, Worksheet, Handling Files.

Worksheet Formatting, Introduction, Entering Text Data, Entering Formula, Editing the Cell Content, Formatting the Cell, Formatting Font, Setting Border Around Cell, highlighting gridlines, Using Format Painter, Finding and Replacing the Text, Using Spelling and Grammar.

Function and Operator, Introduction, Entering Functions, Editing Functions, Using Mathematical Functions, Using Statistical Functions, Using Date and Time Functions, Changing the default date format, Text Function, Logical Functions, Financial Function, Operators, AutoSum, Function Wizard.

Unit-V Chart and Web Object: Introduction, Types of Charts, creating a Quick Chart Sheet, Parts of a Chart, Types of Charts, Creating A Chart using wizard, Using Pivot Table, Object Linking and Embedding (OLE), Linking Cells, Linking Formula, Hyper Links, Previewing charts, printing charts, Exercise.

Presentation Package And MS-PowerPoint: Introduction, Chief Elements of Presentation, Starting PowerPoint, Creating A Presentation, Creating A Presentation with Auto Content Wizard, create a presentation using a design template, creating a blank presentation, PowerPoint window and its Elements, Using Help Online, Customizing Office Assistant.

Text Formatting in Slides: Introduction, adding text to slides, editing text on a slide, Using Format Painter, Setting Paragraph Indents, Line Spacing in a Paragraph, Setting and Removing Tab Stops, Checking Spelling of the text, Finding and replacing the text, Moving slides.

Table, Chart and other Drawing Objects, Introduction, creating a table, creating an embedded Word table, Adding Columns and Rows, Deleting Columns and Rows, Changing Table Borders, Using

Auto shapes, Chart, inserting a clip to your slide, Using Word Art, Inserting A Word Art, Working with Drawing Toolbar, Creating A Shape.

Slides, Views, Notes and Handouts: Introduction, PowerPoint Views, Notes Pages, Using Handouts, Inserting Header and Footer in the, Slide, Transition, Custom Show, Assigning Custom Animation, adding a motion path, animating a chart, publish a presentation or HTML file, to the Web, preview a presentation as a Web page, Showing Slides, Printing Slides.

Outlook Express: Introduction, WHAT IS outlook express?, Features of Outlook Express, starting outlook express, Concepts of CC and BCC, Email address, Reading a received message , composing message, Replying And Forwarding Messages, attaching files, Creating signature in outlook express, Formatting message text, What is mime?, applying stationery, Inserting a hyperlink or HTML page into a message, Flagging an e-mail or news message, Importing messages from other e-mail programs, What are newsgroups?, Adding a newsgroup account, Switching between e-mail and news reading, Identities (Multiple User on A Single Computer), Adding a new identity, Managing contacts with outlook, creating addresses, Importing an address book from another program, Using keyboard shortcuts in Outlook Express.

Practical

1. Introduction of Microsoft windows.
2. Creation of file and folder in MS Windows.
3. Introduction of MS Word.
4. Inserting Number, Bullets, Footer and Header.
5. Creating text, document and table in MS Word.
6. Write steps for mail merge.
7. Introduction of Microsoft excel.
8. Write steps to inserting formula in MS Excel.
9. Creating text, row and Column in MS Excel.
10. Introduction of Microsoft Power Point.
11. Write steps how to using graphics in power point.
12. Introduction and theory of Microsoft Outlook.

Reference books

1. Prof Satish Jain and Geetha Kratika, MS-Office 2010, BPB Publication.
2. Kumar Bittu, MS-Office, V & S Publication.
3. Rakesh Sangwan, MS-Office, Ascent Publication.
4. Freedman Jay and Couch Andrew, MS-Office, PHI Learning.

Facilitating the Achievement of Course Learning Objectives

Unit	Course learning outcomes	Teaching and learning activities	Assessment tasks
I	Understand the evolution and features of the Windows 10 operating system. Familiarize with various Windows accessories and tools. Learn to manage files and folders in Windows 10.	Conduct lectures on Windows 10 history and its elements. Demonstrate the use of Windows accessories and tools through live demonstrations. Provide hands-on practice sessions for managing files and folders.	Windows 10 Features Report, Accessories and Tools Presentation, File Management Exercise.
II	Understand how to customize the Windows 10 desktop. Learn to add or remove programs. Gain an introduction to the Microsoft Office Suite.	Lecture on customizing the desktop and adding/removing programs. Practical exercises for customizing the desktop. Introduction to the Microsoft Office Suite.	Desktop Customization Project, Adding/Removing Programs Report, Microsoft Office Suite Quiz.
III	Understand the features and elements of MS-Word. Learn text formatting and document formatting techniques. Gain proficiency in tables, graphics, and mail merge.	In-depth lectures on MS-Word elements and features. Hands-on practice sessions for text formatting, document formatting, and mail merge. Demonstrate table and graphics creation in MS-Word.	MS-Word Features and Functions Essay, Document Formatting Exercise, Mail Merge and Tables Task.
IV	Understand the elements of MS-Excel and its application window. Learn to format worksheets, enter data, and use functions and operators. Gain proficiency in creating charts and utilizing web objects.	Lectures on MS-Excel basics, including elements and formatting. Practical exercises for data entry, formatting, and formula usage. Demonstrate chart creation and web object integration.	MS-Excel Basics Test, Data Formatting and Chart Creation Quiz, Web Object Integration Exercise.
V	Learn about presentation views, notes, and handouts in PowerPoint. Understand the features and functions of Outlook Express for email communication.	Theory classes on presentation views, notes, handouts, and Outlook Express features. Practical sessions for hands-on experience with different PowerPoint views and email tasks. Exercises on creating presentations with notes and handouts, as well as using Outlook Express for email communication.	Presentation Views and Handouts MCQ, Outlook Express Email Task.

Ability Enhancement Course
3HELA201: English Language and Indian culture
 (Credit: Theory -2 Tutorial - 0)

Scheme of Examination

Course Code	Course Name	Credit	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assign	End Sem	Term Sem			
3HELA201	English Language and Indian culture	2(2+0)	60	30	10	-	-	100	3	-

Course Objective

- To Study the basic concept and Language Skills of English Language.
- Comprehensive study of different kinds of vocabulary in English Language
- To Study the different era in every story and moos in poems.

Course Outcomes

- Students will be able to understand the basic concept and Language Skills of English Language.
- Students will be able to understand the different use of vocabulary in their sentences.
- Students will be able to understand the varieties of stories on different issues and on different format.

Syllabus

Unit 1:

- Amalkanti: Nirendranth Chakrabarti
- Sita: Toru Dutt
- Preface to the Mahabharata: C. Rajagopalachari
- Satyagraha: M.K. Gandhi
- Toasted English: R.K. Narayan
- The Portrait of a lady: Khushwant Singh

Unit 2: Comprehension (unseen passages, summary, note making)

Unit 3: Composition and Paragraph Writing (Based on expansion of an idea)

Unit 4: Basic Language Skills: Vocabulary – Synonyms, Antonyms, Word Formation. Prefixes and Suffixes, Words likely to be confused and Misused, Words similar in Meaning or Form, Distinction between Similar Expressions, Speech Skill.

Unit 5: Basic Language Skills: Grammar and usage – The Tense Forms, Propositions, Determiners and Countable/Uncountable Nouns, Verb, Articles Adverbs.

Reference Books

- English language & Indian Culture – Dr. Pankaj Kumar Singh, Dr. Ashwin Joshi - Thakur Publication, Bhopal.
- Indian Art & Culture – Dr. Manish Rannian (IAS) – Prabhat Prakashn
- Indian Culture & Heritage – Romila Thapar – Kindle Unlimited

Facilitating the Achievement of Course Learning Objectives

Unit	Learning Outcome	Teaching-Learning Activities	Assessment Tasks
1	<ul style="list-style-type: none"> ● Understand the historical context of English in India and its impact on Indian culture. 	<ul style="list-style-type: none"> ● Lecture on the history of English in India. ● Group discussions on the cultural influences of English. ● Readings and analysis of relevant texts. 	<ul style="list-style-type: none"> ● Research paper on the historical development of English in India. ● Group presentation on cultural assimilation.
2	<ul style="list-style-type: none"> ● Analyze the influence of literature in English on Indian culture and identity. 	<ul style="list-style-type: none"> ● Close reading of select literary works by Indian authors writing in English. ● Comparative analysis of Indian and Western literary traditions. ● Guest lectures by Indian authors. 	<ul style="list-style-type: none"> ● Essay on the impact of Indian English literature on cultural identity. ● In-class quizzes on literary analysis.
3	<ul style="list-style-type: none"> ● Explore the role of English in contemporary Indian society and media. 	<ul style="list-style-type: none"> ● Case studies on the use of English in Indian media. ● Group projects on language in advertising. ● Guest speakers from the media industry. 	<ul style="list-style-type: none"> ● Media analysis report on the use of English in Indian news outlets. ● Group presentation on language in advertising campaigns.
4	<ul style="list-style-type: none"> ● Investigate the intersection of English and Indian languages and their cultural significance. 	<ul style="list-style-type: none"> ● Language workshops on common Indian languages and their influence on English. ● Interviews with bilingual/multilingual individuals. ● Analysis of code-switching in communication. 	<ul style="list-style-type: none"> ● Research paper on language convergence and divergence in bilingualism. ● Oral presentations on code-switching in real-life contexts.
5	<ul style="list-style-type: none"> ● Reflect on the challenges and opportunities of bilingualism and multiculturalism in India. 	<ul style="list-style-type: none"> ● Group discussions on identity and language choices. ● Debates on language policy and diversity in India. ● Field visits to multilingual communities. 	<ul style="list-style-type: none"> ● Final reflective essay on personal experiences and insights regarding bilingualism and multiculturalism in India. ● Participation in debates and discussions.

Interdisciplinary Course**3IOSI202: Operating System**

(Credit: Theory -3 Tutorial - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted					Duration of Exam.		
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work			
3IOSI202	Operating System	3(3-0-0)	60	30	10			100	3 hr	-

Course objective

- To develop the understanding of functioning of Operating System.
- To understand the Process Concepts, process state and process control
- To understand the Critical Section Problem
- To understand the Contiguous Allocating, Paging
- To understand the Disk Scheduling, Disk Management

Course Outcome

After study this student will be able to know about the functioning of Operating System. To make students able to learn different types of operating systems along with concept of file systems and CPU scheduling algorithms used in operating system. To provide students' knowledge of memory management and deadlock handling algorithms. At the end of the course, students will be able to implement various algorithms required for management, scheduling, allocation and communication used in operating system.

Syllabus**Theory**

Unit 1: Operating Systems: Overview Introduction of Operating System, Types of Operating System, System Components and it's services, System Calls, System Programs, Structure, Design and, Implementation, Operating System Generation.

Unit 2: Process: Concept, Description and Control Concept of process, Process state model, Process description PCB, Process control, Threads, Threads in Linux

Process Scheduling Types of Schedulers, Scheduling Criteria, Uniprocessor, Scheduling, Multiprocessor scheduling, Algorithm Evaluation, Process Scheduling in Linux

Concurrency Introduction to concurrency, Critical section problem, Mutual Exclusion solutions, S/w approach, H/w support, semaphore, monitor, Classical problem of synchronization.

Unit 3: Deadlock, Deadlock Characterization, Deadlock Prevention, Deadlock Detection, Deadlock Avoidance, Combined Approach

Protection Goal of Protection, Protection Domains, Access Matrix, Implementation of Access Matrix, Revocation of Access Rights, Language Based Protection

Security and Encryption Security Problem, User Authentication, Program Threats, System Threats, Securing System and Facilities, Encryption and Decryption – Cryptography.

Unit 4: Memory Management Memory Management Requirements, Address Space, Linking and Loading, Swapping, Partitioning, Paging, Segmentation.

Virtual Memory Introduction to Virtual Memory, Demand Paging, Page Replacement, Thrashing, Demand Segmentation

Input Output Systems Input Output Devices, Hardware Support for I/o, I/o Communication Techniques, I/o Software Device Drivers, Performance Consideration.

Unit 5: Disk Structure Introduction to Disks, Disk Scheduling, Disk Management, Disk Reliability, Swap Space Management, Stable Storage Implementation.

File Management File Concepts, Directory structure, File Sharing, Protection, File system in Linux.

Reference Books

- Operating Systems – Flynn, McHoes, Cengage Learning.
- Operating Systems – Pabitra Pal Choudhury, PHI.
- Operating Systems – William Stallings, Prentice Hall.
- Operating Systems – H.M. Deitel, P. J. Deitel, D. R. Choffnes, 3rd Edition, Pearson.

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
1	<ul style="list-style-type: none"> • Understand the fundamentals of Operating Systems • Differentiate between various types of Operating Systems • Learn about system calls and system programs • Explore the structure, design, and generation of OS 	<ul style="list-style-type: none"> • Lecture on the introduction to Operating Systems • Explanation of system components and their services • Practical exercises demonstrating system calls and system programs • Handson practice with OS structure, design, and generation 	<ul style="list-style-type: none"> • Quiz on Operating System basics • Written assignment on different OS types • Writing code using system calls • Building a simple OS generation concept
2	<ul style="list-style-type: none"> • Understand the concept of processes and their control • Learn about threads and their usage in Linux • Explore process scheduling techniques • Understand the basics of concurrency and synchronization 	<ul style="list-style-type: none"> • Lecture on the concept of processes and their states • Handson practice with threads in a Linux environment • Discussion on hardware and software approaches to concurrency 	<ul style="list-style-type: none"> • Creating a process control block (PCB) • Implementing multithreading in a Linux program • Writing code to evaluate scheduling algorithms

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
			<ul style="list-style-type: none"> Implementing mutual exclusion using semaphores
3	<ul style="list-style-type: none"> Learn about deadlock and its prevention Understand protection mechanisms in OS Explore security, user authentication, and encryption 	<ul style="list-style-type: none"> Explanation of deadlock characterization and prevention strategies Lecture on protection, access matrix, and its implementation Practical exercises for user authentication and encryption 	<ul style="list-style-type: none"> Written report on deadlock prevention methods Creating access matrix and protection domains Implementing user authentication and encryption in an OS
4	<ul style="list-style-type: none"> Learn about memory management in Operating Systems Understand virtual memory and its advantages Explore Input Output Systems in OS 	<ul style="list-style-type: none"> Explanation of memory management requirements and techniques exercises for virtual memory, demand paging, and page replacement Handson practice with I/O devices, communication techniques, and drivers 	<ul style="list-style-type: none"> Implementing memory management techniques Writing code for demand paging and page replacement
5	<ul style="list-style-type: none"> Learn about disk structure and management Understand file management in Operating Systems Explore file systems in Linux 	<ul style="list-style-type: none"> Lecture on disk structure, scheduling, and reliability Explanation of file concepts, directory structure, and file sharing Practical exercises on Linux file systems and protection 	<ul style="list-style-type: none"> Implementing disk scheduling and reliability features Creating a file management system in an OS Configuring and securing file systems in Linux

VALUE ADDED COURSE**3ICSV206 Cyber Security**

(Credit: Theory -3 Tutorial - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assign.	End Sem	Term work			
3ICSV206	Cyber Security	3-0-0	60	30	10	-	-	100	3 hr	-

Course Objective

- Learn to analyze the security of in-built cryptosystems.
- Know the fundamental mathematical concepts related to security.
- Develop cryptographic algorithms for information security.
- Understand cybercrimes and cyber security.

Course Outcomes

Understand the fundamentals of networks security, security architecture, threats and vulnerabilities. Apply the different cryptographic operations of symmetric cryptographic algorithms. Apply the different cryptographic operations of public key cryptography. Apply the various Authentication schemes to simulate different applications. Understand various cybercrimes and cyber security.

Syllabus:**Theory:****Unit 1: Introduction to Cyber Security Introduction,**

Computer Security, Threats, Harm, Vulnerabilities, Controls, Authentication, Access Control and Cryptography. Web attack: Browser Attacks, Web Attacks Targeting Users, Obtaining User or Website Data, Email Attacks. Network Vulnerabilities: Overview of vulnerability scanning, Open, Port / Service Identification, Banner /Version Check, Traffic Probe, Vulnerability Probe, Vulnerability Examples, OpenVAS, Metasploit. Networks Vulnerability Scanning (Ncat, Socat), Network Sniffers and Injection tools.

Unit 2: Network Défense tools Firewalls and Packet Filters:

Firewall Basics, Packet Filter Vs Firewall, how a Firewall Protects a Network, Packet Characteristic to Filter, Stateless Vs Stateful Firewalls, Network Address Translation (NAT) and Port Forwarding. VPN: the basic of Virtual Private Networks. Firewall: Introduction, Linux Firewall, Windows Firewall. Snort: Introduction Detection System.

Unit 3: Web Application Tools Scanning for web vulnerabilities tools:

Nikto, W3af, HTTP utilities - Curl, OpenSSL and S-tunnel. Application Inspection tools – Zed Attack Proxy, Sql-map, DVWA, Webgoat. Password Cracking and Brute-Force Tools: John the Ripper, LOhtcrack, Pwdump, HTC-Hydra.

Unit 4: Introduction to Cyber Crime, law and Investigation:

Cyber Crimes, Types of Cybercrime, Hacking, Attack vectors, Cyberspace and Criminal Behavior, Clarification of Terms, Traditional Problems Associated with Computer Crime, Introduction to Incident Response, Digital Forensics, Computer Language, Network Language, Realms of the Cyber world. Internet crime and Act: A Brief History of the Internet, Recognizing.

Unit 5: Defining Computer Crime, Contemporary Crimes, Computers as Targets, Contaminants and Destruction of Data, Indian IT ACT Page 3 of 23 2000. Firewalls and Packet Filters, password Cracking, Keyloggers and Spyware, Virus and Worms, Trojan and backdoors, Steganography, DOS and DDOS attack, SQL injection, Buffer Overflow, Attack on wireless Networks.

Reference Books

- Behrouz A. Ferouzan, Deb deep Mukhopadhyay, "Cryptography and Network Security", 3rd Edition, Tata McGraw Hill, 2015.
- Charles Pfleeger, Shari Pfleeger, Jonathan Margulies, "Security in Computing", Fifth Edition, Prentice Hall, New Delhi, 2015.

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
1	<ul style="list-style-type: none"> • Understand the fundamentals of Cyber Security • Explore Computer Security and its importance • Learn about authentication, access control, and cryptography • Understand various web attacks and how to protect against them • Explore network vulnerabilities and scanning techniques 	<ul style="list-style-type: none"> • Lecture on the introduction to Cyber Security • Explanation of computer security, threats, vulnerabilities, and controls • Practical exercises on authentication, access control, and cryptography • Explanation of web attacks, browser attacks, and email attacks • Introduction to vulnerability scanning, network sniffers, and injection tools 	<ul style="list-style-type: none"> • Quiz on Cyber Security basics • Writing a short essay on the importance of Cyber Security • Implementing authentication and access control measures • Identifying and mitigating web vulnerabilities • Conducting network vulnerability scans
2	<ul style="list-style-type: none"> • Learn about network defense tools and techniques 	<ul style="list-style-type: none"> • Lecture on firewalls, packet filters, and VPNs 	<ul style="list-style-type: none"> • Configuring firewalls and VPNs • Implementing packet filtering rules

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
	<ul style="list-style-type: none"> • Understand the role of firewalls and packet filters • Explore VPNs and their basic concepts • Learn about intrusion detection systems (IDS) • Explore web application scanning and security tools • Learn about scanning tools like Nikto and W3af 	<ul style="list-style-type: none"> • Practical exercises on configuring firewalls and packet filters • Explanation of Virtual Private Networks (VPNs) • Introduction to intrusion detection systems (IDS) • Lecture on web application security, scanning tools, and password cracking • Practical exercises with Nikto, W3af, and HTTP utilities 	<ul style="list-style-type: none"> • Setting up a VPN for secure communication • Configuring and using Snort IDS • Conducting web vulnerability scans • Identifying and addressing web vulnerabilities
3	<ul style="list-style-type: none"> • Understand application inspection tools like ZAP and Sqlmap • Learn about password cracking and brute-force tools 	<ul style="list-style-type: none"> • Explanation and hands on practice with Zed Attack Proxy (ZAP) • Introduction to password cracking tools and techniques 	<ul style="list-style-type: none"> • Conducting application security assessments • Cracking passwords and evaluating password security
4	<ul style="list-style-type: none"> • Gain insights into Cyber Crime, laws, and investigation • Understand the types of Cybercrime and attack vectors • Learn about cyber laws and regulations 	<ul style="list-style-type: none"> • Lecture on Cyber Crime, types, and incident response • Explanation of hacking, attack vectors, and digital forensics • Practical exercises on Indian IT Act 2000 and cyber laws 	<ul style="list-style-type: none"> • Quiz on Cyber Crime and laws • Investigating a simulated cybercrime incident • Analyzing legal aspects of Cyber Security incidents
5	<ul style="list-style-type: none"> • Explore various contemporary Cyber Security threats and attacks • Understand the risks associated with different types of attacks <p>Learn about SQL injection, buffer overflow, and wireless network attacks</p>	<ul style="list-style-type: none"> • Introduction to contemporary Cyber Security threats and attacks • Explanation of password cracking, keyloggers, viruses, and malware • Practical exercises on SQL injection, buffer overflow, and wireless attacks 	<ul style="list-style-type: none"> • Identifying and mitigating Cyber Security threats and attacks • Evaluating and mitigating security risks associated with attacks <p>Conducting simulated attacks and defending against them</p>

VALUE ADDED COURSE**3HYEV206: Yoga Education**

(Credit: Practical -2 Tutorial - 1)

Scheme of Examination

Course Details				End Term Practical Exam		Lab Performance		Credit Distribution			Allocated Credits
Course Code	Course Type	Course Title	Total Marks	Major		Minor Sessional ***		L	T	P	Subject wise Distribution
				Max Marks	Min Marks	Max Marks	Min Marks				
Practical Group											
3HYEV206	Value added course	Yoga Education	100	60	20	40	14	-	1	2	3

Objectives

- To provide an understanding of the meaning and definition of Yoga.
- To identify the aims and objectives of Yoga.
- To analyze the role of Yoga in Early Upanishads.
- To understand the Yoga Sutra: General Consideration.
- To recognize the need and importance of Yoga in Physical Education. and Sports

Outcomes

- Understand the definitions and concepts of Yoga.
- Describe the historical development of Yoga in India.
- Know the major schools of Yoga.
- Demonstrate the different stages of the Surya Namaskar.
- Name the different types of Asanas.
- Analyze the preventive and curative effects of Yoga.

Syllabus**Unit 1: Introduction**

- Meaning, History and Development of Yoga.
- Aims and Objectives of Yoga, Time and food.
- The Yoga Sutra: General Consideration.
- Need and Importance of Yoga.

Unit 2: Foundation of Yoga

- Various kind of Yoga (Bhakti yoga, karma yoga, hatha yoga, and Ashtang yoga).
- General guidelines for yoga practice.

- Yoga practice for health and wellness.

Unit 3: Asanas

- Effect of Asanas and Pranayama on various system of the body
- Classification of asanas.
- Influences of relatives, meditative posture on various system of the body.
- Types of Bandhas, mudras and kriyas.

Unit 4: Yoga Education

- Basic, applied and action research in Yoga.
- Difference between yogic practices and physical exercises.
- Yoga education centers in India and abroad.

Unit 5: Yoga and Holistic Health

- Holistic Health and Yoga- Explore the concept of holistic health and how yoga contributes to overall well-being, including physical, mental, and emotional health.
- Yoga for Stress Management - Examine the role of yoga in managing stress and promoting relaxation, with a focus on specific techniques and practices.
- Yoga and Nutrition - Discuss the connection between yoga and nutrition, emphasizing the importance of a balanced diet for a healthy lifestyle.
- Yoga Philosophy and Ethics - Delve into the ethical and philosophical aspects of yoga, including concepts like Ahinsa (non-violence) and Dharma (duty), and how they can be applied in daily life.

Practical

- Prayer: Concept and recitation of pranava.
- Surya Nasmaskar
- Aasana- (Uttanpadasan, Halasan, Pawanmuktasan, Makrasan, Bhujangasan Shaslabhasan, Dhanurasan, Ardha-Mastsyendrasan, Janushirasana, Supta-Vajrasan, Chakrasana, Tadasa, Uktatasana, Padamsana, Gomukhasana, Vajrasana, Pashchimottasana, Sarvangasana, Matsyasan.)
- Chalana kriya/ Loosening Practice
 - Neck Movement
 - Shoulder movement
 - Bhujja Valli shakti vikasaka
 - Purna Bhujja shakti vikasaka
 - Knee Movement
- Yogasana Standing Posture – Tadasana, vrikshasana, Ardha Chakrasana, sarvangasana, trikonasana
Sitting posture – Bhadrasana, vajrasana, Ardha- ushtrasana, shashankasana, vakrasana Prone Posture-
Makarasana, bhujangasana, Shalabhasana
Supine posture- Ardhasana, Setubandhasana, pawanmuktasana, shavasana.
- Pranayam (Anulom-vilom, Nadi-Shodhan, Surya, Bhedi Ujjayi, Shitkari, Sheetali, Bhastrika, Bhramri.)
- Shat-Karma (Cleansing process) (Jal-Neti, Sutra Neti, Kunjal, Trataka, Kapalbhathi)
- Mudra (Mahamudra, Mahabandha, Viparitkarani, Shambhri, Kaki)
- Dhyana (Meditation): "OM" recitation, Body Awareness, Breath Awareness, yoga nidra.

Viva

Practical work

Facilitating the Achievement of Course Learning Objectives

Reference Books

- Gupta S.N. Dass Yoga Philosophy Dr. Bhardwaj Ishwar Upnishdhik & Adhyatmik Yigyan.
- Swami Kuvalayananda Hathyog Preedipika Mukherjee, Wishvananth Bharat Ke Mahaan Yogies.
- Swami Tirth, Omanand Patanjali Yog Pradeep Swami Kuvalayananda Pranayam.
- Swami Saraswati Sataya Nand Asan Pranayam and Mudra Bandh Bharamchari, Swami Dhirender Yogic Suksham vgyan.
- Dr. Nagendra H.R. Pranayama the Arts & Science.
- Swami Kuvalayananda Yogic Chikitisa Ananda Swamy Shankaradev Yogic management and Common.

Unit	Learning Outcome	Teaching and Learning Activities	Achievements
1	Understand the meaning and definition of yoga.	Lecture and discussion on the concept and definition of yoga. - Reading assignments on the history and evolution of yoga.	Define yoga and its historical context.
2	Identify the aims and objectives of yoga practice.	Group discussions on the purposes of practicing yoga. - Research projects on the benefits of yoga in various aspects of life.	List the aims and objectives of yoga.
3	Trace the presence of yoga in early Upanishads.	Analysis of select Upanishadic texts with references to yoga. Group presentations on the historical development of yoga.	Summarize the influence of Upanishads on yoga.
4	Summarize the Yoga Sutra and its general considerations.	In depth study of Patanjali's Yoga Sutras and their significance. Group debates on the key principles in the Yoga Sutra.	Explain the fundamental concepts in the Yoga Sutra.
5	Recognize the need and importance of yoga in physical education and sports.	Guest lectures by yoga experts in sports and physical education. Practical sessions of yoga for athletes and physical education students.	Identify the benefits and relevance of yoga in sports and physical education.

VALUE ADDED COURSE
3HCIV206: Contemporary India
 (Credit: Theory -3 Tutorial - 0)

Scheme of Examination

Course Code	Course Name	Credit	Maximum marks Allotted					Duration of Exam.		
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assign	End Sem	Term Sem			
3HCIV206	Contemporary India	3(3+0)	60	30	10	-	-	100	3	-

Course Objective

- Develop a clear and comprehensive understanding of the definition and scope of Contemporary India.
- Identify and critically analyze the key elements that define the contemporary period.
- Trace and summarize the post-independence historical background, highlighting major events and their significance.
- Evaluate the impact of historical developments on the current socio-political and economic landscape.
- Examine India's demographic profile, including population distribution, age structure, and regional diversity.
- Analyze the cultural and linguistic diversity within India and its implications for national identity.
- Understand the trajectory of economic growth in India, including key sectors and challenges.
- Analyze the role of economic policies in shaping India's growth and development.
- Evaluate social indicators such as education, health, and poverty, understanding their significance in measuring societal well-being.
- Examine the interconnections between social indicators and their impact on the overall quality of life.

DR. C.V.RAMAN UNIVERSITY

Course Outcome

- Students will articulate a precise definition of Contemporary India, demonstrating an understanding of its multidimensional nature.
- Students will categorize and interpret key aspects defining the scope of Contemporary India.
- Students will construct a chronological timeline of post-independence events, demonstrating an understanding of their historical context.
- Students will assess the significance of historical events in shaping the contemporary landscape.
- Students will analyze India's demographic landscape, producing insights into population distribution and diversity.
- Students will recognize and appreciate the cultural diversity within India, linking it to the nation's identity.
- Students will explain the trajectory of economic growth in India, illustrating their understanding of key economic sectors.
- Students will critically assess economic challenges, demonstrating an understanding of their complexities.

- Students will interpret social indicators, showcasing their ability to evaluate education, health, and poverty metrics.
- Students will demonstrate an understanding of the interconnectedness of social indicators and their implications for societal well-being.

Syllabus

Unit 1: Introduction to Contemporary India

- Definition and scope of Contemporary India
- Historical background: post-independence period
- Demographic profile and diversity
- Economic overview: Growth, sectors, and challenges
- Social indicators: Education, health, and poverty

Unit 2: Political Landscape

- Constitution of India: Features and amendments
- Political institutions: Parliament, President, Prime Minister, Judiciary
- Electoral system: Elections, political parties, and regional dynamics
- Major political issues and challenges

Unit 3: Economic Development

- Economic planning and policies
- Agriculture: Green Revolution, challenges, and reforms
- Industry and services sector
- Infrastructure development
- Economic inequality and inclusive growth

Unit 4: Social Issues and Cultural Dynamics

- Social diversity: Caste, religion, ethnicity, and language
- Gender issues: Women empowerment, equality, and challenges
- Cultural heritage: Art, literature, music, and cinema
- Urbanization and changing lifestyles

Unit 5: Contemporary Challenges and Future Prospects

- Environmental challenges: Climate change, pollution, and conservation
- Technological advancements and their impact
- Globalization and India's role in the international community
- Future prospects: Opportunities and challenges

Recommended Texts:

- "India After Gandhi" by Ramachandra Guha
- "India Unbound" by Gurcharan Das
- "The Argumentative Indian" by Amartya Sen
- "Pax Indica" by Shashi Tharoor
- Articles and research papers on contemporary issues

Reference Books

- Author: Ramachandra Guha, Book Title: "India After Gandhi: The History of the World's Largest Democracy", Publication Year: 2007
- Author: Bipan Chandra, Mridula Mukherjee, Aditya Mukherjee, and Sucheta Mahajan, Book Title: "India Since Independence", Edition: 1st Edition Publication Year: 2008

- Author: Amartya Sen, Book Title: "The Argumentative Indian: Writings on Indian History, Culture and Identity", Edition: 1st Edition, Publication Year: 2005
- Author: Shashi Tharoor, Book Title: "Pax Indica: India and the World of the 21st Century", Edition: 1st Edition, Publication Year: 2012
- Author: Arvind Panagariya, Book Title: "India: The Emerging Giant", Edition: Updated and Expanded Edition Publication Year: 2011

Facilitating the Achievement of Course Learning Objectives

Unit	Learning Outcome	Teaching and Learning Activities	Achievements
1	<ul style="list-style-type: none"> ● Introduction to Contemporary India ● Definition and scope of Contemporary India ● Historical background: post-independence period ● Demographic profile and diversity ● Economic overview: Growth, sectors, and challenges ● Social indicators: Education, health, and poverty 	<ul style="list-style-type: none"> ● Lectures and discussions ● Case studies on demographic trends ● Analysis of economic indicators ● Research projects on social issues 	<ul style="list-style-type: none"> ● Increased understanding of Contemporary India ● Enhanced analytical skills
2	<ul style="list-style-type: none"> ● Political Landscape ● Constitution of India: Features and amendments ● Political institutions: Parliament, President, Prime Minister, Judiciary ● Electoral system: Elections, political parties, and regional dynamics ● Major political issues and challenges 	<ul style="list-style-type: none"> ● Interactive sessions on constitutional features ● Role-playing exercises on political processes ● Debates on major political issues 	<ul style="list-style-type: none"> ● Improved understanding of India's political landscape ● Enhanced debating and critical thinking skills
3	<ul style="list-style-type: none"> ● Economic Development ● Economic planning and policies ● Agriculture: Green Revolution, challenges, and reforms ● Industry and services sector ● Infrastructure development ● Economic inequality and inclusive growth 	<ul style="list-style-type: none"> ● Guest lectures from economists ● Case studies on economic policies ● Field visits to industries and farms 	<ul style="list-style-type: none"> ● Increased awareness of economic policies ● Practical insights into economic sectors
4	<ul style="list-style-type: none"> ● Social Issues and Cultural Dynamics 	<ul style="list-style-type: none"> ● Group discussions on social diversity 	<ul style="list-style-type: none"> ● Improved sensitivity to social issues

Unit	Learning Outcome	Teaching and Learning Activities	Achievements
	<ul style="list-style-type: none"> ● Social diversity: Caste, religion, ethnicity, and language ● Gender issues: Women empowerment, equality, and challenges ● Cultural heritage: Art, literature, music, and cinema ● Urbanization and changing lifestyles 	<p>Workshops on gender equality Cultural events and presentations</p>	<p>Enhanced understanding of cultural diversity</p>
5	<ul style="list-style-type: none"> ● Contemporary Challenges and Future Prospects ● Environmental challenges: Climate change, pollution, and conservation ● Technological advancements and their impact ● Globalization and India's role in the international community ● Future prospects: Opportunities and challenges 	<ul style="list-style-type: none"> ● Seminars on environmental challenges ● Analysis of technological impacts ● Model United Nations (MUN) simulations 	<ul style="list-style-type: none"> ● Heightened awareness of global issues ● Improved diplomatic and negotiation skills

SKILL ENHANCEMENT COURSES
3IWDS205: Web Development II (Java Script)
 (Credit: Theory -0 Tutorial – 1 Practical - 1)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assign.	End Sem	Term work			
3IWDS205	Web Development II (Java Script)	0-1-1	-	-	-	60	40	100	-	2h

Course Objective

- The objective of this course is to provide students with a solid foundation in JavaScript programming language.
- By the end of this course, students should be able to understand the fundamental concepts of JavaScript, its role in web development, and how to use JavaScript to manipulate the Document Object Model (DOM) and create dynamic and interactive web pages.
- They should also be proficient in using JavaScript to handle events, manipulate styles, work with strings, arrays, and objects.

Course Outcomes

- Understand JavaScript as a programming language and its role in web development, including its key features and benefits.
- Embed JavaScript code in HTML documents and effectively manipulate the Document Object Model (DOM) to create dynamic and interactive web pages.
- Utilize conditional statements, loops, and functions to control the flow of program execution and solve programming problems.
- Handle events in JavaScript, respond to user interactions, and create interactive web experiences.
- Manipulate HTML content, modify element attributes, and dynamically create and append elements using JavaScript.
- Apply string methods to manipulate and transform text data, work with arrays, and understand the concept of objects, including their properties and methods, in JavaScript.

Syllabus

Theory

Unit 1: Introduction to JavaScript What is JavaScript? JavaScript vs. other programming languages, JavaScript in web development, Embedding JavaScript in HTML, JavaScript data types and variables, Operators and expressions

Unit 2: Control Flow and Functions Conditional statements (if, else if, switch), Loops (for, while, do while), Functions in JavaScript, Function parameters and return values, Scope and closures, Arrow Function.

Unit 3: DOM Manipulation, Introduction to the Document Object Model (DOM), Accessing and modifying DOM elements, Manipulating HTML content with inner HTML, Creating and appending elements dynamically, Modifying element attributes with JavaScript.

Unit 4: Events, Styling and Window object Event handling in JavaScript, commonly used DOM events, Manipulating CSS styles with JavaScript, Adding, removing, and toggling CSS classes, Methods of window object

Unit 5: String Methods, Arrays and Object Introduction to strings in JavaScript, String properties and methods, Introduction to array in JavaScript, Creating and manipulating arrays, Array methods and properties, Working with objects, Object properties and methods.

Practical

- Create an HTML file with a <script> tag to embed a simple JavaScript script that alerts a message.
- Declare variables of different data types (string, number, boolean) and log their types to the console.
- Print only the ODD values from 3 - 29, one number per line
- Write a JavaScript program to calculate the factorial of a given number.
- Attach a JavaScript function to a button click event that toggles the visibility of a hidden element on the page.
- Write a JavaScript function to check if a given number is prime.
- Given an array of numbers, use the filter method to select only the even numbers. Then, use the map method to square each of the selected numbers.
- Create a function that takes a string as input and uses JavaScript to reverse it. For example, inputting "hello" should return "olleh".
- Write a function that checks if a given string is a palindrome using ifelse statements and string manipulation.

Reference Books

- Effective JavaScript: 68 Specific Ways to Harness the Power of JavaScript – David Herman
- JavaScript and jQuery Interactive Front End Web Development by Jon Duckett (Author)
- A Smarter Way to Learn JavaScript – Mark Myers
- JavaScript Programming for Beginner's to Advance 2022 Guide Hindi Version (Hardcover) (Hardcover, Aamer Khan)

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
1	<ul style="list-style-type: none"> • Understand the role of JavaScript in web development • Differentiate JavaScript from other programming languages • Gain knowledge of JavaScript data types and variables 	<ul style="list-style-type: none"> • Lecture on JavaScript fundamentals • Handson practice embedding JavaScript in HTML documents • Interactive coding sessions for data types, variables, operators, etc. 	<ul style="list-style-type: none"> • Quiz on JavaScript basics • Coding exercises for JavaScript variables • JavaScript code review and debugging

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
2	<ul style="list-style-type: none"> • Learn control flow structures in JavaScript • Understand loops and their use in JavaScript • Explore the concept of functions in JavaScript • Learn about function parameters, return values, and scope 	<ul style="list-style-type: none"> • Lecture on conditional statements (if, else if, switch) • Practical examples and exercises for using loops in JavaScript • Handson coding sessions for creating and using functions • Explanation and practice of function parameters, return values, and scope 	<ul style="list-style-type: none"> • Implementing conditional logic in JavaScript • Loop based programming tasks • Function implementation and testing
3	<ul style="list-style-type: none"> • Understand the Document Object Model (DOM) • Learn how to access and modify DOM elements • Explore innerHTML for modifying HTML content • Learn to create and append elements dynamically • Understand how to modify element attributes with JS 	<ul style="list-style-type: none"> • Lecture on DOM fundamentals and its role in web development • Interactive exercises for selecting and manipulating DOM elements • Handson practice with inner HTML to change HTML content • Coding examples for adding and removing elements using JavaScript • Practical sessions on changing element attributes using JavaScript 	<ul style="list-style-type: none"> • DOM manipulation exercises • Building a dynamic webpage using the DOM
4	<ul style="list-style-type: none"> • Gain knowledge of event handling in JavaScript • Learn to manipulate CSS styles with JavaScript • Explore adding, removing, and toggling CSS classes • Understand the methods of the window object 	<ul style="list-style-type: none"> • Lecture on JavaScript event handling and common DOM events • Handson coding sessions for CSS style manipulation with JavaScript • Coding exercises for managing CSS classes in JavaScript • Explanation and practice of various window object methods 	<ul style="list-style-type: none"> • Implementing event listeners and handlers • Styling a webpage with JavaScript
5	<ul style="list-style-type: none"> • Learn about string methods in JavaScript 	<ul style="list-style-type: none"> • Lecture on JavaScript string properties and methods 	<ul style="list-style-type: none"> • JavaScript string manipulation exercises

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
	<ul style="list-style-type: none">• Gain proficiency in working with arrays in JavaScript• Understand objects and their properties and methods	<ul style="list-style-type: none">• Practical examples and exercises for array manipulation in JavaScript• Explanation and practice of object properties and methods in JavaScript	<ul style="list-style-type: none">• Implementing common array operations• Creating and working with JavaScript objects



BACHLOR OF COMPUTER APPLICATION

III Semester

(BCA – CKUG02A03)

(Effective from Academic Year 2023-24)

DR. C.V.RAMAN UNIVERSITY

ABILITY ENHANCEMENT COURSE (AEC)**3HCSA301: Communication Skill**

(Credit: Theory - 2 Tutorial -0)

Course Code	Course Name	Credit	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assign	End Sem	Term Sem			
3HCSA301	Communication Skill	2(2-0-0)	60	30	10	-	-	100	3 hrs	-

Course Objective

- Develop Effective Communication Skills Spoken and written.
- Develop Effective Presentation Skills.
- Conduct Effective business Correspondence, business reports, team management and all-round personality Development.

Course Outcomes

- Projects Role plays, quizzes and Various other participatory sessions. The emphasis will be on learning by doing.
- The student will learn the skills and attributes but also internalize them over a period of time.
- Internalization ensures that the skills and attributes become part of the student's nature. Thus, the changes will be genuine and positive.

Syllabus**Unit 1: Introduction**

General Introduction of self by students, Importance of the Training sessions, Importance of Presentation Skills, Public Speaking

Unit 2: Basic English Grammar

Vocabulary, Kinds of Sentences, Verb, Adverb, Tenses, Preposition, Conjunction, Formation of Sentences, Sentence Making, Translation

Unit 3: Communication Skills

Communication meaning, Function, Process, Types of communication, Guidelines for effective communication, Purpose of Good communication, Importance of right Pronunciation

Unit 4: Listening and Writing Skills

Importance of effective listening, Importance of effective writing skills, Conversation Practice, Guidelines for Effective writing

Unit 5: Body Language

Gestures, Voice Modulation, Eye Contact, Facial Expression, Posture, Dressing Sense, Attire, Hand, movements, General Etiquette, Mannerism, Smiling Gestures, Confidence building. Email Etiquette, Email Drafting

Reference Books

- English Communicative skill by Pramod Singla.
- English Language skill a Practical.
- Communicative skill by Sanjay Kumar & Pushpalata.

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcome	Teaching-Learning Activities	Assessment Tasks
1	<ul style="list-style-type: none"> ● Understand the basics of communication. 	<ul style="list-style-type: none"> ● Lecture on communication fundamentals ● Group discussion on the importance of effective communication ● Role-play exercises demonstrating different communication styles 	<ul style="list-style-type: none"> ● Quiz on communication basics ● Individual reflection on group discussion performance
2	<ul style="list-style-type: none"> ● Develop active listening skills. 	<ul style="list-style-type: none"> ● Listening comprehension exercises ● Class debates with active listening requirements ● Peer feedback sessions 	<ul style="list-style-type: none"> ● Listening comprehension test ● Debate participation and assessment by peers
3	<ul style="list-style-type: none"> ● Improve verbal communication skills. 	<ul style="list-style-type: none"> ● Public speaking workshops ● Mock interviews and feedback sessions. ● Impromptu speech exercises 	<ul style="list-style-type: none"> ● Public speaking assessment with feedback ● Mock interview performance evaluation
4	<ul style="list-style-type: none"> ● Enhance non-verbal communication skills. 	<ul style="list-style-type: none"> ● Body language analysis and practice activities ● Group activities emphasizing non-verbal cues ● Video analysis of non-verbal communication 	<ul style="list-style-type: none"> ● Non-verbal communication assessment with peer feedback ● Written reflection on video analysis
5	<ul style="list-style-type: none"> ● Develop effective written communication skills. 	<ul style="list-style-type: none"> ● Writing workshops on emails, reports, and proposals ● Collaborative document editing exercises ● Writing assignments on various topics 	<ul style="list-style-type: none"> ● Written assignments on emails, reports, and proposals with rubric assessment ● Peer review of collaborative document editing

INTERDISCIPLINARY COURSE (IDC)**3IDMI302: Discrete Mathematics**

(Credit: Theory - 3 Tutorial -0)

Scheme of Examination

Course Code	Course Name	Credit	Maximum marks Allotted						Duration of Exam.		
			Theory			Practical			Total	Theory	Practical
			End Sem	Mid Sem	Assign	End Sem	Term Sem				
3IDMI302	Discrete Mathematics	3(3-0-0)	60	30	10	-	-	100	3 hrs	-	

Course Objective

- This course introduces the applications of discrete mathematics in the field of computer science.
- It covers sets, logic, proving techniques, combinatory, functions, relations, Graph theory and algebraic structures.
- These basic concepts of sets, logic functions and graph theory are applied to Boolean algebra and logic networks while the advanced concepts of functions and algebraic structures are applied to finite state machines and coding theory.

Course Outcomes

After study this student will be able to know about some fundamental mathematical concepts and terminology, how to use and analyses recursive definitions, how to count some different types of discrete structures, techniques for constructing mathematical proofs, illustrated by discrete mathematics examples.

Syllabus**Theory**

Unit 1: Set Theory Set and Subsets, Operations on Sets, Countable and Uncountable Sets, the Principle of Inclusion Exclusion, Derangements, Propositions.

Unit 2: Permutation, Combinations, Discrete Probabilities The rules of sum and product, Permutations, Combinations, Binomial and Multinomial Theorems, Combinations with Repetitions, Probability, Random Variables and Probability Distributions, Repeated Trials. Relation And Function Cartesian (Cross) Product of Sets, Relation, Operation on Relations, Properties of Relation as Binary Relation on a Set, Two Important Relations, Partial Ordered Relation, Lattices, Functions Mappings, Types of Functions, Cardinality of Set, Composition of Relation and Function, Composition of Function, Existence of Inverse Function (Mapping), Set Image/Preset Image of Function.

Unit 3: Groups Introduction, Necessary and sufficient Condition for any subset of a group to be subgroup, Partition of a Group, Characteristics of Cosets of a Subgroups, Normal Subgroups, Necessary and sufficient condition for any subgroup of group to be normal subgroup, Characteristics of Normal (Sub groups), Quotient groups, Concept of Homomorphism. Graph Directed Graphs, Graphs, Isomorphism, Subgraphs, Operations on Graphs, Walks and their classification, Connected and Disconnected Graphs, Euler circuits Euler trails, Planar and nonplanar graphs.

Unit 4: Propositional Logic: Introduction, Statement Or Proposition, Logical connectives or Sentence Connectives ,use of brackets, Kinds of sentences, truth value of s statement, Statement Pattern or Statement form or proposition Form, Truth Value Function, principal Connective, Open Statement, Truth Table, Basic Logic Operation, Kinds of conditional, Tautology, Contradiction, Logical Equivalence, Algebra of Proposition, arguments, Propositional function or Open Sentence or Predicates, Quantifiers, Negation of a Quantifier, De Morgan laws.

Unit 5: Discrete Numeric Functions and Generating Functions Discrete Numeric Functions, Manipulation of Numeric Functions, Asymptotic Behavior of Numeric Functions, Binomial Coefficients.

Reference Books

- Discrete Mathematics by “Rajendra A Kerkar” Publisher Pearson.
- Discrete Mathematics by “Hari Kishan” Publisher Pragati Edition.
- Discrete Mathematics by “Dr. H.K. Pathak.

Facilitating the Achievement of Course Learning Objectives

Unit	Course learning outcomes	Teaching and learning activities	Assessment tasks
1	<ul style="list-style-type: none"> • In a Set Theory course, students will gain a foundational understanding of sets and subsets, learn how to perform operations on sets like union and intersection, grasp the concept of countable and uncountable sets. 	<ul style="list-style-type: none"> • Begin with lectures to introduce fundamental concepts and definitions related to sets, subsets, and set operations. Teaching will be done through lectures and discussion mode. 	<ul style="list-style-type: none"> • Class Discussions and Analysis, Group Projects, Problem Solving Sessions
2	<ul style="list-style-type: none"> • In a course covering Permutations, Combinations, Discrete Probabilities, Relations, and Functions, students will achieve a comprehensive understanding of fundamental mathematical concepts. They will learn to apply the rules of sum and product for counting, master permutations and combinations. 	<ul style="list-style-type: none"> • Start with lectures to introduce key concepts, definitions, and theorems. Teaching will be done through lectures and discussion mode. 	<ul style="list-style-type: none"> • Class Discussions and Analysis, Group Projects, Problem Solving Sessions
3	<ul style="list-style-type: none"> • Students will analyse directed and undirected graphs, distinguish isomorphism, comprehend subgraphs, perform graph operations, categorize walks, identify connected and disconnected graphs, and understand Euler circuits, Euler trails, and planar vs. non planar graphs. 	<ul style="list-style-type: none"> • Provide a set of example graphs and ask students to determine if they are isomorphic or find isomorphic subgraphs. 	<ul style="list-style-type: none"> • Class Discussions and Analysis, Group Projects, Problem Solving Sessions
4	<ul style="list-style-type: none"> • Students will understand group theory, including subgroup conditions, partitioning, cosets, normal subgroups, quotient groups, and homomorphism concepts. They will also grasp ring theory, 	<ul style="list-style-type: none"> • Start with concrete examples to introduce the basic concepts of groups, such as the definition of a group, 	<ul style="list-style-type: none"> • Class Discussions and Analysis, Group Projects, Problem Solving Sessions

Unit	Course learning outcomes	Teaching and learning activities	Assessment tasks
	covering ring types, properties, subrings, ideals, and ring homomorphism	group operations, and identity elements.	
5	<ul style="list-style-type: none"> • Students will learn to work with discrete numeric functions, manipulate them, analyses their asymptotic behavior, and apply these concepts to binomial coefficients using generating functions. 	<ul style="list-style-type: none"> • Begin with lectures and discussions to introduce students to discrete numeric functions, such as factorials, powers, and binomial coefficients. 	<ul style="list-style-type: none"> • Class Discussions and Analysis, Group Projects, Problem Solving Sessions

INTERDISCIPLINARY COURSE (IDC)**3IOAI302: Office Automation**

(Credit: Theory - 0 Tutorial -1 Practical-2)

Scheme of Examination

Course Code	Course Name	Credit	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assign	End Sem	Term Sem			
3IOAI302	Office Automation	3(0-1-2)	-	-	-	-	40	100	-	2hr

Course Objective

- To understand the basic knowledge of MS Windows.
- To understand the Office Packages.
- To understand the MS Excel.

Course Outcomes

After studying this student will be able to know about terms and concepts of Microsoft suite completely. (Like MS word, MS Excel)

Syllabus**Theory**

Unit 1: MS Word Basics: Introduction to MS Office; its components, Introduction to MSWord; Features and area of use. Working with MS Word.; Menus and Commands; Toolbars and Buttons; Shortcut Menus, Wizards and Templates; Creating a New Document; Saving document, saving as different format, Different Page Views and layouts; Applying various Text Enhancements; Working with – Styles, Text Attributes; Paragraph and Page.

Unit 2: Formatting Using page border and watermark, Text Editing using various features; Bullets, Numbering, Auto formatting, word count, various page view options, Printing and various print options

Advanced Features of MSWord: Spell Check, Thesaurus, Find and Replace; Headers and Footers; Inserting – Page Numbers, Pictures, Files, Auto texts, Symbols etc.; Working with section breaks and page breaks, Working with Columns, Tabs and Indents; Creation and Working with Tables including conversion to and from text; Margins and Space management in Document.

Unit 3: Adding References – footnotes, endnotes, and Table of contents, insert drawing, Mail Merge, Envelops and Mailing Labels, protect and secure documents in MS Word, Working in different languages in MS Word. Using Unicode in MS Word, Insert WordArt and other objects like shapes, clipart, charts and Smart Arts, symbol in Document. Using Macros in Word Record, edit and run macros.

Unit 4: MS Excel: Introduction and area of use; Working with MS Excel.; concepts of Workbook and Worksheets; Using Wizards; Various Data Types; Using different features with Data, Cell and Texts; Inserting, Removing and Resizing of Columns and Rows; Working with Data and Ranges; Different Views of Worksheets; Zooming, Column Freezing, Labels, Hiding, splitting etc.; Using different features with Data and Text.

Unit 5: Formulas Use of Formulas, Calculations using various type of functions Logical, string, date and time, math's and other types; Cell Formatting including Borders and Shading; conditional formatting, sorting data items, Working with Different Chart Types; Printing of Workbook and Worksheets with various options. Import and export excel sheets to/from various format, add headers and footers, using macros in excel sheet Record, edit and run macros.

Practical list

- Introduction of Microsoft windows.
- Creation of file and folder in MS Windows.
- Introduction of MS Word.
- Inserting Number, Bullets, Footer and Header.
- Creating text, document and table in MS Word.
- Write steps for mail merge.
- Introduction of Microsoft excel.
- Write steps to inserting formula in MS Excel.
- Creating text, row and Column in MS Excel.
- Introduction of Microsoft Power Point.
- Write steps how to using graphics in power point.
- Introduction and theory of Microsoft Outlook.

Reference books

- Prof Satish Jain and Gerhardtian “MSOffice 2010” BPB Publication.
- Kumar Bittu “MSOffice” V and S Publication.
- Rakesh Sangwan “MSOffice” Ascent Publication.
- Freedman Jay and Couch Andrew “MSOffice” PHI Learning

Facilitating the Achievement of Course Learning Objectives

Unit No.	Course learning outcomes	Teaching and learning activities	Assessment tasks
1	<ul style="list-style-type: none"> • Understand the fundamentals of Microsoft Office and its components. Learn the basics of MS Word and its features. Gain proficiency in using menus, commands, toolbars, and buttons in MS Word. Learn to create, save, and format documents in MS Word. Explore various text enhancements, styles, and page layout options. Understand how to apply styles, text attributes, and manage paragraphs and pages in documents. 	<ul style="list-style-type: none"> • MS Word basics. Practical exercises on creating, saving, and formatting documents. Hands on experience with menus, commands, and toolbars. Demonstrations of text enhancements, styles, and page layout options. Group discussions on document creation and management. Case studies on real world document formatting. 	<ul style="list-style-type: none"> • Quizzes on Microsoft Office and MS Word basics. Document creation and formatting assignments. Group project on applying text enhancements and styles.

Unit No.	Course learning outcomes	Teaching and learning activities	Assessment tasks
2	<ul style="list-style-type: none"> Gain proficiency in advanced formatting features of MS Word, including page borders, watermarks, and text editing options. Learn about bullets, numbering, auto formatting, word count, and various page view options. Understand printing options and various print settings in MS Word. Explore advanced features like Spell Check, Thesaurus, and Find and Replace. Learn about headers, footers, and inserting page elements like page numbers, pictures, files, and symbols. Understand section breaks, working with columns, tabs, indents, and table creation and management. 	<ul style="list-style-type: none"> Practical exercises on advanced formatting in MS Word. Hands-on experience with text editing and page view options. Coding practice for document printing and advanced features. Demonstrations of Spell Check, Thesaurus, and Find and Replace. Group discussions on headers, footers, and page elements. Case studies on practical applications of advanced MS Word features. 	<ul style="list-style-type: none"> Implementation and testing of advanced formatting in MS Word. Document formatting assignments. Quizzes on advanced MS Word features.
3	<ul style="list-style-type: none"> Understand how to add references such as footnotes, endnotes, and tables of contents in MS Word. Learn about drawing tools, mail merge, envelopes, mailing labels, and document protection. Explore working in different languages and using Unicode in MS Word. Gain proficiency in inserting objects like WordArt, shapes, clipart, charts, and SmartArt into documents. Learn the basics of using macros in MS Word, including recording, editing, and running macros. 	<ul style="list-style-type: none"> Practical exercises on adding references and using drawing tools. Demonstrations of working in different languages and using Unicode. Group discussions on inserting objects into documents. Case studies on real world applications of macros in MS Word. 	<ul style="list-style-type: none"> Assignments on adding references and using drawing tools. Quizzes on mail merge, document protection, and object insertion.
4	<ul style="list-style-type: none"> Understand the basics of MS Excel and its workbook and worksheet 	<ul style="list-style-type: none"> Practical exercises on workbook and worksheet operations. Hands-on 	<ul style="list-style-type: none"> Implementation and testing of workbook and worksheet operations.

Unit No.	Course learning outcomes	Teaching and learning activities	Assessment tasks
	<p>concepts. Learn about workbook and worksheet operations. Explore various data types, including text, numbers, and dates. Gain proficiency in using different features with data, cells, and texts in MS Excel. Learn how to insert, remove, and resize columns and rows. Explore different views of worksheets, zooming, column freezing, labels, hiding, and splitting. Understand features like cell formatting, borders, shading, conditional formatting, and sorting data items.</p>	<p>experience with data types and cell formatting. Demonstrations of column and row management in Excel. Group discussions on data features and worksheet views. Case studies on practical applications of Excel features.</p>	<p>Problem solving assignments on data features and cell formatting. Quizzes on Excel basics and column/row management.</p>
5	<ul style="list-style-type: none"> Gain proficiency in using Excel formulas and various functions, including logical, string, date and time, math, and others. Understand cell formatting techniques, including borders, shading, and conditional formatting. Learn how to sort data items and work with different chart types in Excel. Understand printing options for workbooks and worksheets. Learn how to import and export Excel sheets to/from various formats. Explore adding headers and footers, and using macros in Excel, including recording, editing, and running macros. 	<ul style="list-style-type: none"> Practical exercises on using Excel formulas and functions. Hands-on experience with cell formatting and data sorting. Coding practice for chart creation and printing options. Demonstrations of data import/export and macro usage in Excel. Group discussions on Excel features and functions. Case studies on real world Excel applications and macros. 	<ul style="list-style-type: none"> Implementation and testing of Excel formulas and functions. Problem-solving assignments on cell formatting and data sorting. Quizzes on Excel features and chart creation. Group discussions, Unit Test.

MAJOR CORE

3IPJC303 - Programming with JAVA
(Credit: Theory - 4 Tutorial -1 Practical-2)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work			
3IPJC303	Programming with JAVA	4-0-2	60	20	20	60	40	200	3 hr	2 hr

Course objective

1. To introduce and understand students to programming concepts and techniques using the Java language and programming environment, class, objects, also learn about lifetime, scope and the initialization mechanism of variables and improve the ability general problem-solving abilities in programming.
2. Be able to use the Java SDK environment to create, debug and run simple Java program.

Course outcomes

Students will complete software projects comprised of an object-oriented design, implementation, and test plan.

1. Designs will demonstrate the use of good object-oriented design principles including encapsulation and information hiding.
2. The implementation will demonstrate the use of a variety of basic control structures including selection and repetition; classes and objects in a tiered architecture (user interface, controller, and application logic layers); primitive and reference data types including composition; basic AWT components; file-based I/O; and one-dimensional arrays.
3. Test plans will include test cases demonstrating both black box and glass box testing strategies.

Syllabus**Theory**

Unit-I OVERVIEW OF JAVA - Introduction, Programming paradigm, OOPS Concepts, Evolution of Java, Features of Java, C++ Vs Java, Java and Internet, Java and WWW, Java support systems, Java Environment.

Key Features of Java - Introduction, Java Program Structure, Simple Java Program, Tokens, Java Statements, Java Virtual Machine, Constants and Variables, Declaration of Variables, Scope of Variables, Data types, Symbolic Constants, Type Casting, Command line arguments.

Unit-II OPERATORS - Operators, Arithmetic Operators, Relational Operators, Logical Operators, Bitwise Operators, Increment and Decrement, Conditional Operators, Special Operators, Assignment Operators, Expression and its evaluation.

CONTROL STATEMENTS - Introduction, Control Statements, Sequence Control Statement, Decision Control Statement, Case Control Statement, Iteration Control Statement, jump in loops, Labeled Loops.

ARRAYS AND STRINGS - Introduction, Array, Need of Array, Types of Arrays, one dimensional Array, Two-Dimensional Array, Multidimensional Array, Strings, Concatenation of Strings, Methods for String Comparison, Methods for searching Strings, Changing the case of characters, String Buffer.

Unit-III

Classes - Introduction, defining a Class, Adding Variables, Adding Methods, Creating Objects, Accessing Class members, call by value and call by reference, Recursion, Access Control, Constructors, Method overloading, Constructor Overloading, Garbage Collection, finalize () method, this keyword, Static Members, Nesting of Methods

Inheritance - Inheritance, Single Inheritance, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance, Using Super, Constructor -Order of Execution in Inheritance, Overriding methods, Final variables and methods, Final Classes, Abstract methods and Classes, Containership, Visibility Control.

Unit-IV

Wrapper classes and vectors - Introduction, Wrapper Classes, Number Class, Byte class, short class, Integer class, long class, Converting Numbers to and from Strings, Float class, Double class, Character class, Boolean class, Vectors, Creating a vector

Interface and Packages - Introduction, Interfaces, defining interface, implementing interface, accessing interface method, accessing interface variable, extending interfaces, Packages, System packages, using system packages, User defined packages, adding class to a package, Accessing and using package.

Exception Handling - Introduction, Exceptions, Using try and catch, Multiple catch clauses, Finally, Throw, Throws

Multithreading - Introduction, The Main Thread, Creating Threads, Life cycle of Thread, Using Threads Methods, Thread Priorities, Stopping and Blocking a thread, Thread Exceptions, Using is Alive () and join (), Synchronization.

Unit-V

APPLETS - Introduction, Local and remote applets, Applet vs applications, writing applets, Life cycle of an applet, creating source code of applet, creating an executable applet, creating applet tag, adding applet tag to html, Running the applet, Detailed form of applet tag, passing parameters to applet, Aligning the display, Html tags, Getting input from user

Input-Output Streams and File Management - Introduction, Stream, Stream Classes, Byte Stream Classes, Character Stream Classes, System Class, Reading Console Input, Writing Console Output, Using the File Class, Random Access File

graphics programming - Introduction, The Graphics Class, Drawing Lines and Rectangles, using draw Oval () and fill Oval () method, drawing arcs, Drawing Polygon, Line Graphs, Drawing Bar Charts.

Practical

1. Write a Java Program to Display message on computer screen.
2. Write a Java Program to develop a class for Rational numbers
3. Design a Date class in Java
4. Write a Java Program to design an interface for Stack ADT and implement Stack ADT using both Array and Linked List.
5. To develop a vehicle class hierarchy in Java to demonstrate the concept of polymorphism
6. Design a Date class in Java

7. To write a Java Program to randomly generate objects and write them into a file using concept of Object Serialization
8. Develop a scientific calculator using even-driven programming paradigm of Java.
9. To write a multi-threaded Java program to print all numbers below 100,000 that is both prime and Fibonacci number
10. To develop a Java Program that supports multithreaded echo server and a GUI client.
11. To implement a calculator using GUI Environment with the help of java x. swing package.

Reference books

1. E BALAGURUSAMY, Programming with Java, McGraw Hill.
2. Dr. R. NAGESWARA RAO, Core Java an Integrated Approach, Dreamtech Press.
3. HERBERT SCHILDT, Java a Beginner's Guide, McGraw Hill.
4. Herbert Schildt, The Complete Reference, McGraw Hill.

Facilitating the Achievement of Course Learning Objectives

Unit	Learning Outcomes	Teaching & Learning Activities	Assessment Tasks
1	<ol style="list-style-type: none"> 1. Understand the basics of Java, OOPS, and its features. 2. Compare Java with C++ and understand its role on the web. 3. Explore Java environment components and support systems. 	<p>Lectures on Java introduction and evolution.</p> <p>Discussions on Java's role in internet and WWW.</p> <p>Demonstrations of Java environment setup.</p> <p>-Practical examples of Java programming.</p>	<p>Quiz on Java basics.</p> <p>Assignment: C++ vs. Java comparison.</p> <p>Lab exercises on Java environment.</p>
2	<ol style="list-style-type: none"> 1. Comprehend Java program structure and operators. 2. Master control statements for flow control. 3. Understand arrays, strings, and their manipulation. 	<p>Lectures on Java program structure and operators.</p> <p>Code walkthroughs of control statements.</p> <p>Hands-on practice with arrays and strings.</p>	<p>Practical exercises on Java programs.</p> <p>-Assignment: Control statement practice.</p> <p>- Lab tasks on arrays and strings.</p>
3	<ol style="list-style-type: none"> 1. Learn about classes, objects, and inheritance. 2. Explore constructors, method overloading, and this keyword. 3. Understand the concepts of interfaces and packages. 4. Grasp exception handling and multithreading. 	<p>- In-class coding exercises for class and object creation.</p> <p>- Group discussions on constructors and overloading.</p> <p>- Hands-on exercises to create and use interfaces and packages.</p> <p>- Live debugging sessions for exception handling.</p>	<p>- Class project: Implementing inheritance.</p> <p>- Individual project: Recursion in Java.</p> <p>-Assessment: Interface implementation.</p> <p>- Practical exam: Multithreading in Java.</p>

Unit	Learning Outcomes	Teaching & Learning Activities	Assessment Tasks
4	1. Familiarize with wrapper classes and vectors. 2. Explore abstract classes, final classes, and containership.	- Code walkthroughs for wrapper classes and vectors. - Practical examples of abstract and final classes.	- Assignment: Vector implementation. Group Discussion: Designing a container.
5	1. Learn about applets and their integration with HTML. 2. Understand input-output streams and file management. 3. Explore graphics programming in Java.	- Applet development hands-on session. - Coding exercises for input-output streams and file handling. - Graphics programming demonstrations.	- Applet project with HTML integration. - Practical exam: File management in Java. Machine Test: Creating a graphical app.

MINOR CORE

3IDMM304 - Database Management System

(Credit: Theory - 4 Tutorial -0 Practical-2)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical			Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work	Total		
3IDMM304	Database Management System	4-0-2	60	20	20	60	40	200	3 hr	2 hr

Course objective

1. To understand the basic knowledge of DBMS Concepts.
2. To understand the Database Design.
3. To understand the RELATIONAL DATA MODEL.
4. To understand the RELATIONAL DATABASE DESIGN.
5. To understand the Indexing and Hashing-Basic Concepts and Recovery System.
6. System.

Course outcomes

After study this student will be able to know about and concepts and fundamentals of DBMS, Concept of keys, RELATIONAL DATA MODEL and design. Student will also able to create table and implement commands.

Syllabus**Theory****Unit-I**

DBMS Concepts and architecture Introduction, Database approach v/s Traditional file accessing approach, Advantages, of database systems, Data models, Schemas and instances, Data independence, Overall Database Structure, Functions of DBA, ER data model: Entities and attributes, Entity types, Defining the E-R diagram, Concept of Generalization, Aggregation and Specialization. transforming ER diagram into the tables. Various other data models object-oriented data Model, Network data model, and Relational data model.

Unit-II

Relational Data models: Domains, Tuples, Attributes, Relations, Characteristics of relations, Keys, Key attributes of relation, Relational database, Schemas, Integrity constraints. Referential integrity, Intension and Extension, Relational Query languages: SQL-DDL, DML, integrity constraints, Complex queries, various joins, indexing, Relational algebra and relational calculus, Relational algebra operations like select, Project, Join, Division, outer union.

Unit-III

(Constraints, Views and SQL) What constraints, types are of constrains, Integrity constraints, Views: Introduction to views, data independence, security, updates on views, comparison between tables and views SQL: data definition, aggregate function, Null Values, nested sub queries, Joined relations. Triggers.

Unit-IV Relational Database design: Functional Dependency –Definition, trivial and non-trivial FD, closure of FD set, closure of attributes, irreducible set of FD, Normalization –1NF, 2NF, 3NF, Decomposition using FD-dependency preservation, lossless join, BCNF, Multi-valued dependency, 4NF, Join dependency and 5NF.

Unit-V (Transaction management and Concurrency control)

Transaction management: ACID properties, serializability and concurrency control, Lock based concurrency control (2PL, Deadlocks), Time stamping methods, optimistic methods, database recovery management.

Practical's

1. Write a query to implement Different types of DDL statements in SQL.
2. Write a query to implement Different types of DML statements in SQL.
3. Write a query to implement Different types of DQL statements in SQL.
4. Write a query to implement Different types of DCL statements in SQL.
5. Write a query to explore 'select' clause using where, order by, between, like, group-by, having etc.
6. Write a query to implement the concept of Joins in SQL.
7. Write a query to implement the concept of Indexes and views.
8. Write a query to implement the restrictions on the table.
9. Write a query to implement the concept of Sub Questionaries.
10. Write a query to implement the structure of the table.

Reference books

1. "Database System Concepts" by Silberschatz, Korth, Sudarshan, 4th Edition, McGraw Hill Publication.
2. "Database Systems, Concepts, Design and Applications" by S.K. Singh, Pearson Education.
3. "Database Management Systems" by Raghu Ramakrishnan, Johannes Gehrke, McGraw Hill Publication.
4. "Fundamentals of Database Systems" by Elmsari, Navathe, 5th Edition, Pearson Education (2008).

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
1	1. Understand DBMS concepts and architecture. 2. Compare the database approach with the traditional file accessing approach. 3. Recognize the advantages of database systems. 4. Comprehend data models and data independence.	-Lectures on DBMS concepts and architecture - In-class discussions and case studies on database vs. file systems -Reading assignments on the advantages of database systems - Practical exercises on creating schemas and instances - Hands-on lab sessions for ER modeling and table transformation.	- Quiz on DBMS concepts and architecture - Group presentation on database vs. file systems - Practical assignments on schema creation and ER modeling - Group discussion evaluation on data models
2	1. Understand the relational data model and its components. 2.	- Lectures on the relational data model, domains, tuples, and attributes -	-Class Test on the relational data model and SQL - Practical assignments

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
	Learn about integrity constraints and relational query languages.	Hands-on exercises with relation characteristics and keys - SQL, Practice sessions for various joins and indexing - Introduction to relational algebra and calculus - Group discussions on relational database schemas and integrity constraints	involving SQL queries and constraints - Group Discussion on designing relational schemas - Quizzes on relational algebra and calculus - Class participation and discussions
3	1. Explore constraints, views, and SQL. 2. Understand data independence and security. 3. Learn about triggers.	- Lectures on different types of constraints and views - Hands-on practice with creating and managing views, aggregate functions, and subqueries - Triggers and their applications - Comparative analysis of tables and views - Security considerations in SQL	- Assignments on creating and using constraints - Practical assessment on views and security in SQL - Triggers implementation project - Quiz on SQL and views - Comparative analysis report on tables vs. views
4	1. Understand relational database design principles. 2. Learn about functional Dependency and normalization.	- Lectures on functional dependency and normalization concepts - Practical exercises on identifying functional dependencies and closure - Decomposition using FD with preservation and lossless join - BCNF and multi-valued dependencies - Discussions on join dependency and 5NF	- Assignments on identifying functional dependencies and normalization - Presentation on BCNF and multi-valued dependencies
5	1. Learn about transaction management and concurrency control. 2. Understand ACID properties and database recovery management.	- Lectures on ACID properties and serializability - In-depth discussions on lock-based concurrency control and deadlock handling - Time-stamping and optimistic concurrency control methods - Database recovery management techniques	- Class Test on transaction management and concurrency control - Case studies on ACID properties and concurrency control - Practical assignments on implementing concurrency control.

SKILL ENHANCEMENT COURSE

3IWDS305: Web Development III (React.js)

(Credit: Theory - 0 Tutorial -1 Practical-2)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical			Theory	Practical
			End Sem	Mid Sem	Assign.	End Sem	Term Work	Total		
3IWDS305	Web Development III (React.js)	3(0-1-2)	-	-	-	60	40	100	-	2hr

Course Objective

- The objective of this course is to introduce students to React.js and provide them with a comprehensive understanding of its core concepts and features.
- By the end of this course, students should be able to develop dynamic and interactive web applications using React.js,
- understand the advantages of React over other JavaScript frameworks, and effectively integrate React with other libraries and APIs.

Course Outcomes

- Understand the fundamentals of React.js, including its purpose in web development and its advantages over other JavaScript frameworks.
- Develop components in React.js using JSX syntax, understand component lifecycle, and create reusable and modular UI elements.
- Style React components using different approaches, such as inline styles, CSS modules, and CSS in JS libraries.
- Handle data flow and component communication in React.js using props and effectively manage events.
- Implement routing in React applications using React Router, including setting up routes, working with parameters, and nested routing.
- Build forms in React.js, integrate form libraries like Formik, perform form validation using Yup schema validation, and handle form submissions.

Syllabus**Theory**

Unit 1: Introduction to React.js What is React.js? Concept of React.js, Why learn React.js, Key Features of React.js, History of React.js, React versions, React.js vs. other JavaScript frameworks, Virtual DOM and its advantages, setting up a React development environment, Create React app using Vite. File and Folder Structure.

Unit 2: Main Concepts – I Hello World, Introduction of JSX, Function and Class Components, rendering a Component, Difference between function and class components, Component composition and reusability, Extracting components, Styling, Props.

Unit 3: Main Concepts – II Use State, changing state, Handling Events in react, Conditional Rendering, List – (Rendering Multiple Components, Basic List Component), Keys, Forms.

Unit 4: React Router and Introduction to React Router, setting up routing in React, Route parameters and URL parameters, Browser Router, Link, Outlet, Nested routing.

Unit 5: React Hooks Introduction to React hooks, use State, use Effect, use Context and Custom hooks and their usage, react hooks best practices, Migrating from class components to functional components with hooks.

Practical

- Use Vite to set up a new React app and create a React component that renders "Hello, React!".
- Develop a functional component that displays a list of items.
- Create separate components for header, content, and footer, and compose them into a complete page.
- Pass a prop from a parent component to a child component and display it.
- Build a counter component using use State for state management.
- Implement a form with input fields, and conditionally render content based on form submission.
- Render a list of components based on an array of objects, ensuring each item has a unique key.
- Use React Router to set up routes for Home and About pages.
- Extend the previous example to include nested routes and components.
- Use the use Effect hook to fetch data from an API and update the component state.

Reference Books

- The Road to Learn React: Your journey to master plain yet pragmatic React.js by Robin Wieruch
- React.js For Beginners by Mayur Patil
- Beginning React with Hooks by Greg Lim
- Fullstack React: The Complete Guide to ReactJS and Friendsby Anthony Accomazzo

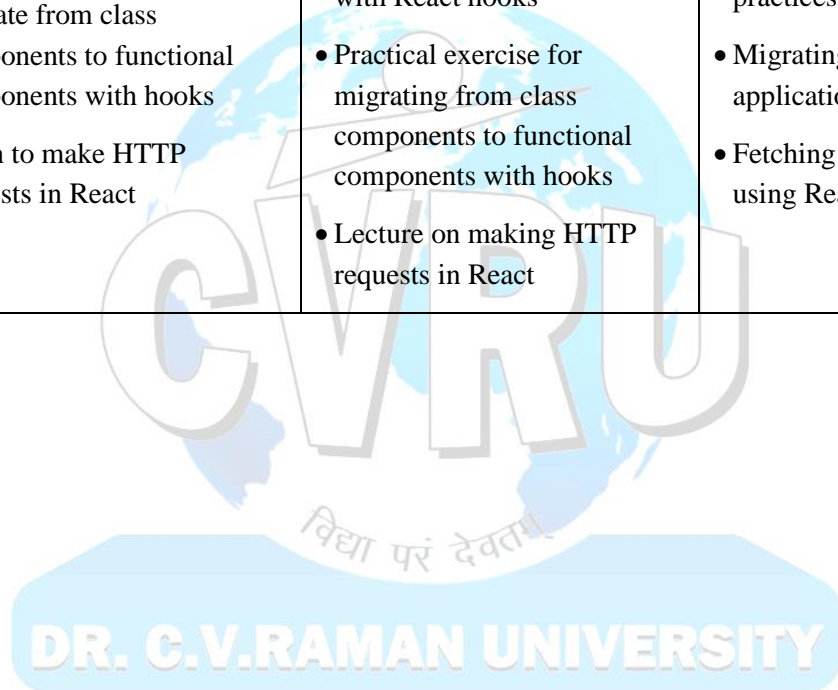
DR. C.V.RAMAN UNIVERSITY

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
1	<ul style="list-style-type: none"> • Understand the fundamentals of React.js • Differentiate React.js from other JavaScript frameworks • Learn about Virtual DOM and its advantages • Gain proficiency in JSX syntax and its usage 	<ul style="list-style-type: none"> • Lecture on React.js basics and its role in web development • Setting up a React development environment • Practical exercises demonstrating the Virtual DOM concept • Handson coding sessions using JSX syntax 	<ul style="list-style-type: none"> • Environment setup and configuration. • Building a simple React component and lifecycle usage • Styling a React application • Quiz on React.js concepts •

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
	<ul style="list-style-type: none"> • Understand React components and their lifecycle • Learn various styling techniques in React 	<ul style="list-style-type: none"> • Explanation and practical examples of React components and lifecycle • Styling components using CSS and other styling methods 	
	<ul style="list-style-type: none"> • Learn to create functional components in React • Explore component composition and reusability • Understand how to work with props and prop types • Learn event handling techniques in React 	<ul style="list-style-type: none"> • Lecture on functional components and their advantages • Handson practice for composing and reusing React components • Practical exercises for passing and validating props • Coding examples and exercises for handling events in React 	<ul style="list-style-type: none"> • Creating and using functional components • Building a modular React application with components • Prop based component creation and validation • Implementing event handling in a React application
3	<ul style="list-style-type: none"> • Gain knowledge of React Router and its role in SPAs • Learn to set up routing in React applications • Explore route parameters and URL parameters • Understand nested routing and route guarding • Learn about React Router hooks (use Params, use History, etc.) 	<ul style="list-style-type: none"> • Lecture on the importance of React Router in single page applications Handson practice for configuring routes and navigation • Practical exercises for handling route parameters and URL parameters • Explanation and examples of nested routing and route protection • Demonstrations and exercises for using React Router hooks 	<ul style="list-style-type: none"> • Setting up routing in a React application • Creating a multipage React application with routing • Implementing dynamic routes with parameters • Implementing nested routes and route guarding • Incorporating React Router hooks in an application
4	<ul style="list-style-type: none"> • Learn to create forms with React • Explore form libraries and integrate Formik • Understand form validation with Yup schema validation • Handle form submissions and display validation errors 	<ul style="list-style-type: none"> • Lecture on form creation in React • Handson practice integrating Formik with React forms • Practical exercises for implementing form validation with Yup • Coding examples and exercises for form 	<ul style="list-style-type: none"> • Creating a simple form with React • Building forms with Formik and validation • Validating forms and displaying errors with Yup • Handling form submissions and error display

Unit	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
	<ul style="list-style-type: none"> • Introduction to React hooks and their use • Learn about use State, use Effect, use Context, and more • Explore custom hooks and their usage 	<ul style="list-style-type: none"> • submission and error rendering • Lecture on React hooks and their advantages • Handson coding sessions for using various React hooks • Explanation and practical examples of creating and using custom hooks 	<ul style="list-style-type: none"> • Implementing use State and use Effect hooks • Incorporating React hooks in a React application • Creating and using custom hooks
5	<ul style="list-style-type: none"> • Understand best practices for using React hooks • Migrate from class components to functional components with hooks • Learn to make HTTP requests in React 	<ul style="list-style-type: none"> • Discussions and coding exercises on best practices with React hooks • Practical exercise for migrating from class components to functional components with hooks • Lecture on making HTTP requests in React 	<ul style="list-style-type: none"> • Implementing React hooks following best practices • Migrating an existing application to use hooks • Fetching data from an API using React





BACHLOR OF COMPUTER APPLICATION

IV Semester

(BCA – CKUG02A03)

(Effective from Academic Year 2023-24)

ABILITY ENHANCEMENT COURSE (AEC)

3HCHA 401: Culture Heritage

(Credit: Theory -2 Tutorial - 0)

Scheme of Examination

Course Code	Course Name	Total Marks	External Assessment		Internal Assessment		Credit Distribution			Allotted Credits
			Major		Minor		L	T	P	
			Max Marks	Min Marks	Max Marks	Min Marks				
3HCHA401	Culture Heritage	100	60	20	40	14	2	-	-	02

Course Objective

- The primary objective of this course is to explore the significance, evolution, and preservation of culture and heritage across different societies and time periods.
- Students will gain a deeper understanding of how culture and heritage shape identities, influence societies, and contribute to a sense of belonging.

Course Outcomes

- Upon completion of this course, students will have a comprehensive understanding of culture and heritage, their significance in society, and the various challenges and methods associated with their preservation.
- They will develop critical thinking skills to evaluate cultural issues and contribute to the sustainable management of cultural heritage.

Syllabus:**Duration 30 hrs (Credit – 2)****Unit 1: Introduction to Culture and Heritage**

- Definition of culture and heritage
- Importance of studying culture and heritage
- Overview of key concepts and terms (e.g., cultural diversity, cultural identity, intangible cultural heritage)

Unit 2: Cultural Expressions and Artifacts

- Exploration of various forms of cultural expressions (e.g., music, dance, visual arts, literature, traditional crafts)
- Analysis of the role of cultural expressions in preserving and transmitting heritage
- Case studies of significant cultural artifacts and their historical and cultural significance

Unit 3: UNESCO World Heritage Sites

- Introduction to UNESCO and its World Heritage program
- Study of selected UNESCO World Heritage Sites from different regions
- Analysis of the criteria for selection and the challenges faced in preserving and protecting these sites

Unit 4: Cultural Identity and Cultural Heritage

- Examination of the relationship between cultural identity and cultural heritage
- Discussion on the ways in which cultural heritage shapes individual and collective identities
- Impact of globalization and cultural assimilation on cultural identity and heritage preservation

Unit 5: UNESCO World Heritage Sites

- Introduction to UNESCO and its World Heritage program
- Study of selected UNESCO World Heritage Sites from different regions
- Analysis of the criteria for selection and the challenges faced in preserving and protecting these sites

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Objectives	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
Unit 1: Introduction to Culture and Heritage	<ul style="list-style-type: none"> ● Define culture and heritage. ● Explain the importance of studying culture and heritage. ● Familiarize students with key concepts and terms related to culture and heritage (e.g., cultural diversity, cultural identity, intangible cultural heritage). 	<ul style="list-style-type: none"> ● Students will be able to define culture and heritage. ● Students will understand the significance of studying culture and heritage. ● Students will demonstrate knowledge of key concepts related to culture and heritage. 	<ul style="list-style-type: none"> ● Lectures and discussions on the definition and importance of culture and heritage. ● Reading assignments and case studies to explore key concepts. ● Group discussions and presentations on cultural diversity and identity. 	<ul style="list-style-type: none"> ● Class participation and engagement in discussions. ● Written assignments or quizzes on key concepts. ● Group presentations on cultural diversity and identity.
Unit 2: Cultural Expressions and Artifacts	<ul style="list-style-type: none"> ● Explore various forms of cultural expressions such as music, dance, visual arts, literature, and traditional crafts. Analyze the role of cultural expressions in preserving and transmitting heritage 	<ul style="list-style-type: none"> ● Students will be able to identify different forms of cultural expressions. ● Students will understand the importance of cultural expressions in heritage preservation. Students will analyze the historical and cultural significance of selected artifacts. 	<ul style="list-style-type: none"> ● Practical sessions or workshops on various cultural expressions (e.g., music and dance workshops, art exhibitions). ● Case study analysis and group discussions on the role of cultural expressions in heritage preservation. ● Research assignments on selected cultural artifacts and their significance. 	<ul style="list-style-type: none"> ● Presentation or demonstration of a chosen cultural expression. ● Written analysis of the role of cultural expressions in heritage preservation. ● Research paper or presentation on a selected cultural artifact and its significance
Unit 3: UNESCO	<ul style="list-style-type: none"> ● Introduce students to UNESCO and its 	<ul style="list-style-type: none"> ● Students will understand the role of UNESCO 	<ul style="list-style-type: none"> ● Lectures and presentations on UNESCO and the 	<ul style="list-style-type: none"> ● Research project or presentation on a specific UNESCO

Unit	Course Learning Objectives	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
World Heritage Sites	<p>World Heritage program.</p> <ul style="list-style-type: none"> • Study selected UNESCO World Heritage Sites from different regions. • Analyze the criteria for selection and the challenges faced in preserving and protecting these sites. 	<p>in preserving cultural and natural heritage.</p> <ul style="list-style-type: none"> • Students will gain knowledge of specific UNESCO World Heritage Sites and their cultural importance • Students will analyze the criteria used for selecting World Heritage Sites and the preservation challenges they face. 	<p>World Heritage program.</p> <ul style="list-style-type: none"> • Virtual or physical tours of selected UNESCO World Heritage Sites. • Discussions on the criteria for World Heritage Site selection and the preservation challenges they pose. 	<p>World Heritage Site.</p> <ul style="list-style-type: none"> • Essay or report on the criteria for selecting World Heritage Sites and the challenges they face in preservation
Unit 4 Cultural Identity and Cultural Heritage	<ul style="list-style-type: none"> • Examination of the relationship between cultural identity and cultural heritage Discussion on the ways in which cultural heritage shapes individual and collective identities Impact of globalization and cultural assimilation on cultural identity and heritage preservation 	<ul style="list-style-type: none"> • Students will be able to: Analyze the relationship between cultural identity and cultural heritage Evaluate the impact of cultural heritage on individual and collective identities Assess the impact of globalization and cultural assimilation on cultural identity and heritage preservation 	<ul style="list-style-type: none"> • Lectures and presentations on cultural identity and cultural heritage Group discussions and debates on the impact of cultural heritage on individual and collective identities Case studies on the impact of globalization and cultural assimilation on cultural identity and heritage preservation 	<ul style="list-style-type: none"> • Written assignments on the analysis of the relationship between cultural identity and cultural heritage Oral presentations on the impact of cultural heritage on individual and collective identities Case study analysis on the impact of globalization and cultural assimilation on cultural identity and heritage preservation
Unit 5 Cultural Identity and Cultural Heritage	<ul style="list-style-type: none"> • Introduction to UNESCO and its World Heritage program Study of selected UNESCO World Heritage Sites from different regions Analysis 	<ul style="list-style-type: none"> • Students will be able to: Understand the importance of UNESCO World Heritage sites Analyze the criteria for selection of 	<ul style="list-style-type: none"> • Lectures and presentations on UNESCO and World Heritage sites Group discussions and debates on the criteria for selection and 	<ul style="list-style-type: none"> • Written assignments on the analysis of the criteria for selection and challenges faced in preserving and protecting World Heritage sites Oral

Unit	Course Learning Objectives	Course Learning Outcomes	Teaching and Learning Activities	Assessment Tasks
	of the criteria for selection and the challenges faced in preserving and protecting these sites	World Heritage sites Evaluate the challenges faced in preserving and protecting World Heritage sites	challenges faced in preserving and protecting World Heritage sites Field trips to selected World Heritage sites	presentations on the importance of UNESCO World Heritage sites Group projects on the preservation and protection of World Heritage site.

MAJOR CORE - I

3IPPC403P - Python Programming –I

(Credit: Theory -4 Practical - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.		
			Theory			Practical			Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work				
3IPPC403	Python Programming -I	4-0-2	60	20	20	60	40	200	3 hr	2 hr	

Course objective

1. Develop a strong understanding of Python basics, including its syntax, data types, and language features, enabling them to write Python code effectively.
2. Master the use of operators and control flow in Python, allowing them to implement conditional statements, loops, and iterators in their programs.
3. Acquire proficiency in functions and arrays, understanding their applications and leveraging NumPy for array processing.

Course outcomes

Students will develop a strong understanding of Python programming, including its basics, syntax, and data types. They will become proficient in using operators and control flow to implement conditional statements, loops, and iterators. Students will gain expertise in functions, arrays, lists, and tuples, leveraging NumPy for array processing and list comprehensions for data manipulation

Syllabus**Theory****Unit-I**

Python Basics and Data Type: Python language introduction, Python 3 basics, Python the new generation language, Comparisons between C and Python, Comparisons between Java and Python, Important difference between python 2.x and python 3.x with example,

Viewing the Byte Code, Python Virtual Machine (PVM), Keywords in Python, Namespaces and Scope in Python, Statement, Indentation and Comment in Python, Structuring Python Programs, Decision making and Python Language advantages and applications. Taking input in Python, taking input from console in Python, Taking multiple inputs from user in Python, Python Input Methods for Competitive Programming, Python | Output using print () function, Python | end parameter in print (), Python | sep parameter in print(), Python | Output Formatting. Introduction to Data Types, Single line comments, Multi line comments, Docstrings, How Python Sees Variables. Datatypes in Python, Built-in data types, The None Type, Numeric Types, Representing Binary, Octal and Hexadecimal Numbers, Converting the Datatypes Explicitly bool Datatype, Sequences in Python, str Datatype, bytes Datatype, byte array, Datatype, list Datatype, tuple Datatype range, Datatype. Sets set Datatype, frozen set Datatype, Mapping Types, Literals in Python, Numeric Literals Boolean Literals, String Literals, Determining the Datatype of a Variable, What about Characters, User-defined Datatypes Constants in Python, Identifiers and Reserved words, Naming Conventions in Python.

Unit-II Operator and Control Flow: Basic operator in python, Assignment operators, Unary minus operators, Relational operators, Logical operators, Boolean operators, Bitwise operators, Membership operators, Identity operators, Operator precedence and Associativity, Operator function in python,

Difference between == and is operator in Python, Python Membership and Identity Operators, Loops and Control Statements (continue, break and pass) in Python, Chaining comparison in python, else with for, switch function, Python Itertools, Python `__iter__()` and `__next__()` | Converting an object into an iterator, Difference between iterable and iterator, Generators in python, Generator's expression in python.

Unit-III Functions and Array: Functions in Python, class method vs static method in Python, Partial Functions in Python, Precision Handling, Python closures, Function Decorators, Decorators in Python,

Decorator with parameters in Python, Memorization using decorators in Python, Help function in Python, Coroutine in Python, Python bit functions on int (bitlength, to bytes and from bytes).Array, Advantage of arrays, Creating an array, Importing the Array Module, Indexing and Slicing on Array, Processing the Arrays, Types of Array, Working with Array using NumPy mathematical Operations on Arrays, Comparing Arrays, Aliasing the Arrays, Viewing and Copying Arrays, Dimensions of Arrays, Attributes of an Array.

Unit-IV Lists and Tuples: List, Creating Lists using range() Function, Updating the Elements of a List, Concatenation of Two Lists, Repetition of Lists, Membership in Lists, Aliasing and Cloning Lists, Methods to Process Lists, Finding Biggest and Smallest Elements in a List, Sorting the List Elements, Number of Occurrences of an Element in the List, Finding Common Elements in Two Lists, Storing Different Types of Data in a List, Nested Lists, Nested Lists as Matrices, List Comprehensions, Tuples, Creating Tuples, Accessing the Tuple Elements, Basic Operations on Tuples, Functions to Process Tuples, Nested Tuples, Inserting Elements in a Tuple Modifying Elements of a Tuple, Deleting Elements from a Tuple.

Unit -V Introduction to OOPS and Data Handling Using Pandas

Problems in Procedure Oriented Approach, Specialty of Python Language, Features of Object-Oriented Programming System (OOPS), Classes and Objects, Encapsulation, Abstraction, Inheritance, Polymorphism. Introduction to python Libraries, Overview of Data Structures, working with series, working with Data frame, Importing/Exporting Data between CSV files Data frames, Overview of

descriptive statistics, Explanation of Data frame Operations, handling missing values, Importing/Exporting Data between MySQL Database and Pandas.

Practicals

1. Write a program to demonstrate basic data type in python.
2. a program to compute distance between two points taking input from the user Write a program add.py that takes 2 numbers as command line arguments and prints its sum.
3. Write a Program for checking whether the given number is an even number or not. Using a for loop.
2. Write a Program to demonstrate list and tuple in python. Write a program using a for loop that loops over a sequence.
3. Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.
4. Write a program to use split and join methods in the string and trace a birthday of a person with a dictionary data structure

5. Create a Python script using Pandas to import data from a CSV file, perform descriptive statistics, and handle missing values.
6. Explore and explain the operator precedence and associativity in Python.
7. Write a Python program that uses control statements (if-else) for decision-making and loops for iteration.
8. 10) Implement a Python script that uses decorators to add functionality to a function. Demonstrate the use of bit functions on an integer.

Reference books

1. R. Nageswara Rao Core Python Programming, Dreamtech Press
2. Eric Matthes, Python Crash Course, No Starch Press
3. John M Zelle, Python Programming, Ingram short title
2. Wes Mckinney Python for Data Analysis, O'R

Facilitating the Achievement of Course Learning Objectives

Unit	Course learning outcomes	Teaching and learning activities	Assessment tasks
1	Understand the basics of Python programming. Differentiate between Python 2.x and Python 3.x. Comprehend Python's data types and their usage.	<ul style="list-style-type: none"> - Lectures on Python language introduction, syntax, and data types. - Comparisons between Python 2.x and Python 3.x with code examples. - Hands-on exercises for input and output in Python. - Group discussions on Python's advantages and applications. 	<ul style="list-style-type: none"> - Quiz on Python basics and differences between versions. - Assignment on writing Python programs to solve specific problems.
2	<ul style="list-style-type: none"> - Understand and apply various Python operators. - Demonstrate control flow constructs in Python. - Utilize iterators, generators, and decorators. 	<ul style="list-style-type: none"> - Lectures on Python operators, control statements, and iterators. - Hands-on exercises on operator usage and control flow. - Code walkthroughs for iterators, generators, and decorators. 	<ul style="list-style-type: none"> - Practical exam on writing Python code using operators and control flow constructs. - Assignment on implementing iterators, generators, or decorators.
3	<ul style="list-style-type: none"> - Master functions and closures in Python. - Explain decorators and coroutines. - Explore Python's bit manipulation functions. 	<ul style="list-style-type: none"> - Lectures on Python functions, closures, decorators, and coroutines. - Hands-on practice writing functions and decorators. - In-class exercises on Python's bit manipulation functions. 	<ul style="list-style-type: none"> - Coding project involving the use of closures, decorators, or coroutines. - Quiz on Python functions and bit manipulation.
4	<ul style="list-style-type: none"> - Learn about Python lists and tuples. - Understand array manipulation using NumPy. 	<ul style="list-style-type: none"> - Lectures on lists, tuples, and NumPy arrays. - Practical sessions on creating and manipulating lists and arrays. - Code walkthroughs for array operations using NumPy. 	<ul style="list-style-type: none"> - Assignment on implementing specific list and array operations. - Practical exam on working with NumPy arrays.
5	- Introduce Object-Oriented Programming	- Lectures on OOP principles, classes, and objects in Python.	- Project involving the design of Python

Unit	Course learning outcomes	Teaching and learning activities	Assessment tasks
	(OOP) concepts in Python. - Demonstrate data handling using Pandas.	<ul style="list-style-type: none">- Hands-on exercises on creating Python classes and objects.- Practical sessions on data handling with Pandas.	<ul style="list-style-type: none">classes and data analysis using Pandas.- Final exam covering OOP concepts and Panda's data handling.

MAJOR CORE - II

3ISEC403 - Software Engineering and UML

(Credit: Theory -4 Practical - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical			Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work	Total		
3ISEC403	Software Engineering and UML	5-1-0	60	20	20	-	-	100	3 hr	-

Course objective

Student will be able-

1. To understanding the concepts and methods required for the construction of large software intensive systems.
2. To develop a broad understanding of the discipline of software engineering.
3. To understanding the detailed knowledge of techniques for the analysis and design of complex software intensive systems.
4. To understanding the techniques in an appropriate engineering and management context.
5. To understanding the brief account of associated professional and legal issues.

Course outcome

1. Carry out an evaluation and selection of projects against strategic, technical and economic criteria and use a variety of cost benefit evaluation techniques for choosing among competing project proposals. Approach project planning in an organized step by step manner and select an appropriate process model produce an activity plan for a project.
2. Identify project risks, monitor and track project deadlines and produce a work plan and resource schedule. Plan the evaluation of a proposal or a product and manage people in software environments. Understand the importance of teamwork and quality management in software project management. Apply these project management tools and techniques in a diversity of fields such as new product and process development, construction, information technology, health care, and applied research.

Syllabus**Theory**

Unit- I Software Engineering paradigms – Waterfall Life cycle model – Spiral Model – Prototype Model– Software Requirement - Requirements Elicitation Techniques – Initial Requirements Document — SRS Document – Requirements Change Management - Project Management.

Unit- II Software Design Abstraction – Modularity – Software Architecture – Cohesion – Coupling – Various Design Concepts and notations – Development of Detailed Design and Creation of Software Design Document - Dataflow Oriented design – Designing for reuse – Programming standards.

- Unit- III** Software Metrics Scope – Classification of metrics – Measuring Process and Product attributes Direct and Indirect measures – Reliability – Software Quality Assurance – Standards. Need of Software Estimation – Function Point – Risk Management.
- Unit- IV** Software Testing and Maintenance Software Testing Fundamentals – Software testing strategies – Black Box Testing – White Box Testing – System Testing – Functional Testing – Structural Testing – Regression Testing - Testing Tools – Test Case Management – Challenges of Software Maintenance – Types of Maintenance. Software Maintenance Organization, Maintenance Report.
- Unit- V** Unified Modelling Language Introduction to UML: Use Case Approach, Identification of Classes and Relationships, Identifying State and Behavior, Use Case Diagram Class Diagram – State Diagram - Sequence Diagram – Activity Diagram – Deployment Diagrams Case Study – LMS.

Reference books

1. Roger S. Pressman, Software Engineering: A Practitioner's Approach, Tata McGraw –Hill Education, 8th Edition, 2015.
2. Sommerville, Software Engineering, Sixth Edition, Addison Wesley -Longman, 2004.
3. Pankaj Jalote, An Integrated approach to Software Engineering, Second Edition, Springer Verlag, 1997.

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities	Assessment Tasks
1	<ol style="list-style-type: none"> 1. Understand software engineering paradigms 2. Familiarity with software development models 3. Ability to gather and document requirements 4. Manage requirements changes 5. Intro to project management principles 	<ul style="list-style-type: none"> - Lectures on software engineering paradigms - Case studies on Waterfall, Spiral, Prototype - Requirement's elicitation role-play exercises - Group discussion on requirements change - Project management principles lectures 	<ul style="list-style-type: none"> - Quiz on software engineering paradigms - Group presentation on a chosen model - Initial Requirements Document submission - Requirements Change Management report - Project management case study and report
2	<ol style="list-style-type: none"> 1. Grasp software design principles 2. Create software design documents 3. Understand modularity, cohesion, and coupling 4. Develop for reuse 5. Apply programming standards 	<ul style="list-style-type: none"> -Design principles workshops and exercises -Software design documentation workshops -Modularization and coupling exercises -Reuse techniques and examples - Programming standards lecture 	<ul style="list-style-type: none"> - Design principles assessment -Software Design Document submission - Modularity and coupling assessment - Reusable component creation project - Code review and adherence assessment
3	<ol style="list-style-type: none"> 1. Define software metrics and their importance 2. Measure process and product attributes 3. Ensure software quality through standards 	<ul style="list-style-type: none"> - Metrics explanation and real-world examples - Measurement process and tools demonstrations - Software quality standards discussions - Function Point estimation exercises 	<ul style="list-style-type: none"> - Metrics measurement and analysis exercises - Product and process metrics measurement - Software quality audit and report - Function Point estimation project - Risk management plan and presentation

	<ol style="list-style-type: none"> 4. Estimate software projects using Function Point 5. Manage software project risks 	<ul style="list-style-type: none"> - Risk management techniques and case studies 	
4	<ol style="list-style-type: none"> 1. Comprehend software testing fundamentals 2. Apply various testing strategies 3. Utilize testing tools 4. Manage software maintenance challenges 5. Understand software maintenance organization 	<ul style="list-style-type: none"> -Testing fundamentals lectures and examples -Testing strategy workshops and simulations -Testing tool demonstrations and practice - Maintenance challenges discussions -Maintenance organization structure and roles 	<ul style="list-style-type: none"> - Testing fundamentals quiz - Testing strategy selection and justification - Testing tool proficiency assessment - Maintenance challenges case study and report - Maintenance organization analysis report
5	<ol style="list-style-type: none"> 1. Introduce UML and its components 2. Apply UML in a real-world case study 	<ul style="list-style-type: none"> - UML introduction and components overview - Case study on a Learning Management System (LMS) 	<ul style="list-style-type: none"> - UML diagram creation and explanation - UML diagrams for LMS



MINOR CORE

3IDS404 - Data Structure

(Credit: Theory -4 Practical - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.		
			Theory			Practical			Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work				
3IDS404	Data Structure	4-0-2	60	30	10	60	40	200	3 hr	2 hr	

Course objectives

Data structures play a central role in modern computer science. Data structures are essential building blocks in obtaining efficient algorithms.

1. The objective of the course is to teach students how to design, write, and analyze the performance of programs that handle structured data and perform more complex tasks, typical of larger software projects.
2. Students should acquire skills in using generic principles for data representation and manipulation with a view for efficiency, maintainability, and code reuse.
3. Another goal of the course is to teach advance data structures concepts, which allow one to store collections of data with fast updates and queries.

Course outcome

After study this student will be able to know about the concepts of Data Structure Using C++ Language, List and Its Operations Concept of Tree, and Algorithm and Graphs Design. Students will also know about the sorting and searching.

Syllabus**Theory**

Unit-I Analysis of Algorithm-Introduction, Criteria of Algorithm, Time Complexity, Space Complexity, Asymptotic Notation: Big Oh (O) Notation: Big Omega (Ω) Notation: Big Theta (Θ) Notation.

Types of Data structures- Introduction, Types of Data structures, Linear Data Structures, Non-Linear Data Structure, Array, SPARSE MATRICES, Garbage Collection, Benefits, Disadvantages.

Unit-II Stacks-Introduction, Push operation, Pop operation, Stack implementation using arrays, (static implementation of stacks), STACK as a Linked List, Stack as an abstract data structure, Applications of stack, Conversion of Expressions, Precedence and associativity of the operators, Evaluation of Postfix expression, Multiple stacks,

Recursion-Introduction, Working of recursion, Fibonacci series, Tower of Hanoi, Efficiency of recursion.

Queue-Introduction, Different types of queues, Queue (Linear queue), Queue as an abstract data structure, Circular queue, Double ended queue (Deque), Priority queue, QUEUE as a Linked List, Applications of Queue.

Linked Lists-Concept of list and array, Introduction to Data Structures, Arrays, Linked list, Singly or Linear linked list, Circular singly linked list, doubly linked lists, Header Node, Applications of linked lists, Addition of two long positive numbers, Evaluation of a polynomial.

Unit-III Trees-Introduction, Representation of tree, Binary Tree, Representation of binary tree, Array representation of binary tree, Linked List representation of binary tree, Basic Operation on Binary Tree- Traversals, Binary Tree Traversal Algorithms (Recursive), Creation of Binary Search Tree, Types of binary trees, Operations on Binary Search Tree (BST), Threaded binary trees, Application of Binary Tree: B-Tree, Height Balanced Tree,

Graph- Introduction to Graphs, Undirected Graph, Directed Graph or digraph, Graph Representation, Adjacency Matrix Representation, Adjacency List Representation, Graph Traversals, Breadth First

Traversal, Depth First Traversal, Searching in Graph, Minimal Spanning Tree, Kruskal's Algorithm, Prim's Algorithm, Shortest Path in Graph.

Unit-IV Sorting and Searching - Introduction, Bubble sort, Selection Sort, Merge Sort, Quick sort, Insertion Sort, Shell sort, Address calculation sort, Radix sort, Comparison of sorting methods, Hash Table, Collision Resolution Techniques, Linear Search (Sequential Search), Binary Search, searching an ordered table, Indexed sequential search, Interpolation search.

Unit-V File Structure and Indexing- Introduction, Objectives, Terminology, File Organization, Sequential Files, Disadvantages, Direct File Organization, Indexed Sequential File Organization.

Practical list

1. Write a Program for Push and Pop operation in Stack?
2. Write a Program for Sparse matrix in Array?
3. Write a Program for insertion, deletion and display operation on queue using array?
4. Write a Program for insertion, deletion and display operation on Circular queue using array?
5. Write a Program sorting by quick sort?
6. Write a Program sorting by Selection sort?
7. Write a Program sorting by Bubble sort?
8. Write a Program sorting by Merge sort?
9. Write a Program insertion, deletion and traversing in single linked list?
10. Write a Program insertion, deletion, searching and traversing in Binary Search tree?
11. Write a Program for Binary Tree?
12. Write a Program for Dequeue?
13. Write a Program for Single Queue?

Reference books

1. Varun Jain, Amrisha Agrwal Kalyani Publication.
2. Vikram Gupta, S.S. Bhatiya, Kalyani Publication.
3. Nitin N. Sakhare, Dr. Amol V. Dhumane, Nirali Prakashan, 3rd edition, Nirali Publication.
2. Babasaheb Mohite, Amit Sattikar and V.V. Pawar, Vision Publication

Facilitating the Achievement of Course Learning Objectives

Unit No.	Course learning outcomes	Teaching and learning activities	Assessment tasks
1	Understand the principles of algorithm analysis and the criteria for evaluating algorithms. - Learn about time complexity and space complexity. - Familiarize with asymptotic notations (Big O, Big Omega, Big Theta). - Explore different types of data structures, including linear and non-linear structures. - Understand the advantages and disadvantages of data structures. - Gain knowledge about arrays, sparse matrices, and garbage collection.	- Problem-solving exercises on time and space complexity. - Hands-on experience with asymptotic notations. - Demonstrations of various data structures. - Group discussions on data structure benefits and drawbacks. - Case studies on array usage and garbage collection.	- Quizzes on algorithm analysis and asymptotic notations. - Problem-solving assignments on data structure implementation. - data structure benefits and disadvantages. -
2	Understand the concepts of stacks, including push and pop operations. - Learn about stack implementation using arrays and linked lists. - Explore the abstract data structure of a stack. - Gain proficiency in using stacks for expression conversion and evaluation. - Learn about recursion and its efficiency. - Explore the basics of queues, including linear, circular, and priority queues. - Understand linked lists and their applications.	- Lectures on stack concepts, operations, and implementation. - Practical exercises on stack applications and recursion. - Coding practice for expression conversion and evaluation. - Hands-on experience with queue types and linked lists. - Group discussions on recursion efficiency and queue applications. - Case studies on linked list usage and polynomial evaluation.	- Implementation and testing of stack operations and recursion. - Problem-solving assignments on expression conversion and evaluation. - Quizzes on queue types and linked list applications. -
3	Understand the basics of trees and their representations. - Explore binary trees and their operations. - Learn about binary search trees, threaded binary trees, and B-trees. - Gain knowledge about height-balanced trees. - Explore the fundamentals of graphs, including types and representations. - Understand graph traversals and searching algorithms. - Learn about minimal spanning trees and shortest path algorithms.	- Lectures on tree concepts, binary trees, and operations. - Practical exercises on binary search trees and height-balanced trees. - Hands-on experience with graph representations and traversals. - Demonstrations of searching algorithms. - Group discussions on minimal spanning trees and shortest path algorithms. - Case studies on practical applications of trees and graphs.	- Implementation and testing of tree operations and searching algorithms. - Problem-solving assignments on graph traversals and algorithms. - Quizzes on tree concepts and graph representations. -
4	Understand sorting and searching techniques. - Learn about various sorting methods, including bubble sort, selection sort, merge sort, and quick sort. - Explore data structures like hash tables and collision resolution techniques. - Gain knowledge about linear search, binary search, and other searching algorithms. - Learn about file structure objectives and	- Practical exercises on sorting algorithms and data structures. - Coding practice for searching algorithms. - Hands-on experience with hash tables and file structures. - Group discussions on sorting and searching methods. - Case studies on file organization and indexed sequential files.	- Implementation and testing of sorting and searching algorithms. - Problem-solving assignments on data structures and file organization. - Quizzes on sorting and searching methods. -

Unit No.	Course learning outcomes	Teaching and learning activities	Assessment tasks
	disadvantages. - Explore direct file organization and indexed sequential file organization.		
5	Understand file structure objectives and terminology. - Learn about different types of file organization, including sequential files and direct file organization. - Explore indexed sequential file organization. - Gain knowledge about file operations and indexing. - Understand the principles of efficient file access. - Learn about file import/export and macros.	- . Practical exercises on different file organizations and indexing. - Hands-on experience with file operations. - Demonstrations of efficient file access principles. - Group discussions on file import/export and macros. - Case studies on real-world file organization scenarios.	- Implementation and testing of file operations and indexing. - Problem-solving assignments on file organization and access. - Quizzes on file structure objectives and terminology. - Group project on analysing file access principles. -





BACHLOR OF COMPUTER APPLICATION

V Semester

(BCA – CKUG02A03)

(Effective from Academic Year 2023-24)

MAJOR CORE - I

3IPPC503 - Python Programming-II

(Credit: Theory -4 Practical - 2)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.		
			Theory			Practical			Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work				
3IPPC503	Python Programming -II	6(4-0-2)	60	20	20	60	40	200	3 hr	2 hr	

Course objective

1. Master Python's Object-Oriented Programming System (OOPS) concepts, including classes, objects, inheritance, and polymorphism.
2. Learn about abstract classes, interfaces, and handle exceptions effectively in Python.
3. Gain proficiency in developing graphical user interfaces (GUI) and performing data visualization tasks.
2. Understand networking concepts, work with sockets, and implement TCP/IP and UDP protocols.

Course outcome

1. Demonstrate a strong understanding of Object-Oriented Programming (OOPS) concepts, including class creation, inheritance, and polymorphism in Python.
2. Apply abstract classes and interfaces effectively and handle exceptions for error handling in Python programs.
3. Develop Graphical User Interfaces (GUI) and perform data visualization using Python libraries.
4. Understand networking principles and implement TCP/IP and UDP protocols for communication using sockets in Python.

Syllabus**Theory****Unit 1: Concept of Oops**

Problems in Procedure Oriented Approach, Specialty of Python Language, Features of Object-Oriented Programming System (OOPS), Classes and objects, Encapsulation, Abstraction, Inheritance, Polymorphism.

Classes and Objects: Creating a Class, The Self-Variable, Constructor, Types of Variables, Namespaces, Types of Methods (Instance Methods Class Methods and Static Methods), Passing Members of One Class to Another Class, Inner Classes.

Inheritance and Polymorphism: Constructors in Inheritance, Overriding Super Class Constructors and Methods. The super() Method, Types of Inheritance (Single Inheritance and Multiple Inheritance), Method Resolution Order (MRO), Polymorphism, Duck Typing Philosophy of Python, Operator Overloading, Method Overloading, Method Overriding.

Unit 2: Abstract Classes, Interfaces and Exceptions

Abstract Method and Abstract Class, Interfaces in Python, Abstract Classes vs. Interfaces.

Exceptions: Errors in a Python Program (Compile-Time Errors, Runtime Errors, Logical Errors). Exceptions, Exception Handling, Types of Exceptions, The Except Block, the assert Statement, User Defined Exceptions Logging the Exceptions.

Unit 3: Graphical User Interface and Networking in Python and Data Visualization

GUI: GUI in Python, The Root Window, Fonts and Colors. Working with Containers, Canvas, Frame, Widgets, Button Widget, Arranging Widgets in the Frame, Label Widget, Message Widget, Text Widget Scrollbar Widget, Check button Widget, Radio button Widget, Entry Widget, Spin box Widget, List box Widget, Menu Widget, Creating Tables

Data Visualization: Bar Graph, Histogram, creating a Pie Chart, Creating Line Graph, Machine Learning the Machine Learning Process.

Unit 4: Networking in Python

Protocol: TCP/IP Protocol, User Datagram Protocol (UDP), Sockets, Knowing IP Address, URL and types of URL, Reading the Source Code of a Web Page, downloading a Web Page from internet, downloading an image from Internet, A TCP/IP Server, A TCP/IP Client, A UDP Server, A UDP Client, File Server File Client Two-Way Communication between Server and Client Sending a Simple Mail.

Unit 5: Python's Database Connectivity

DBMS, Advantages of a DBMS over Files, Types of Databases Used with Python, Installation of MySQL Database Software, Verifying MySQL in the Windows Operating System, Installing MySQL Module Verifying the MySQL Interface Installation, Working with MySQL Database. Using MySQL from Python, Retrieving All Rows from a Table, Inserting Rows into a Table, Deleting Rows from a Table, Updating Rows in a Table, Creating Database Tables through Python, Installation of Oracle 11g, Verifying Oracle Installation in Windows Operating System, Installing Oracle Database Driver Verifying the Driver Installation, Working with Oracle Database, Using Oracle Database from Python, Stored Procedures.

Practical list

1. Create a Python class representing a "Car" with attributes like model, color, and methods for starting and stopping the car.
2. Implement a base class "Animal" with a method "sound ()". Create derived classes "Dog" and "Cat" that override the "sound ()" method.
3. Define an abstract class "Shape" with an abstract method "area ()". Create two classes "Circle" and "Rectangle" implementing the "Shape" abstract class.
4. Write a program that divides two numbers and handles exceptions such as Zero Division Error and Value Error.
5. Create a simple GUI application with a button that changes the color of a label when clicked.
6. Use Matplotlib to create a bar graph representing the sales data of a fictional company for the last quarter.
7. Implement a TCP server that sends a message to a client, and the client displays the received message.
8. Develop a file server that allows clients to upload and download files. Test the server-client communication.
9. Connect to a MySQL database, retrieve data from a table, and display it in Python. Perform an update operation on a specific record.
10. Create a Python program that connects to an Oracle database, executes a stored procedure, and fetches the result.

Reference books

1. R. Nageswara Rao Core Python Programming, Dreamtech Press.
2. Eric Matthes, Python Crash Course, No Starch Press
3. John M Zelle, Python Programming, Ingram short title
4. Wes McKinney Python for Data Analysis, O'Reilly

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
1	<ul style="list-style-type: none"> - Understand the concept of Object-Oriented Programming (OOP) - Identify the features of Python as an OOP language - Comprehend the concepts of encapsulation, abstraction, inheritance, and polymorphism 	<ul style="list-style-type: none"> - Lecture on Problems in Procedure Oriented Approach and OOP concepts - Interactive coding exercises to create classes and objects - Group discussions and examples on inheritance, polymorphism, and encapsulation - Hands-on practice with Python's classes, objects, and inheritance 	<ul style="list-style-type: none"> - Quiz on OOP concepts - Coding assignments on OOP in Python - Group project on designing an OOP system
2	<ul style="list-style-type: none"> - Differentiate between abstract classes and interfaces - Understand exceptions, error types, and exception handling - Learn to use logging for exception handling 	<ul style="list-style-type: none"> - Lectures on abstract classes and interfaces in Python - Practical exercises on exception handling, assert statements, and custom exceptions - Code review sessions and debugging exercises 	<ul style="list-style-type: none"> - Written test on abstract classes and interfaces - Coding assignments on exception handling - Logging the exceptions in a project
3	<ul style="list-style-type: none"> - Develop graphical user interfaces in Python - Create data visualizations using Python libraries - Understand the basics of machine learning 	<ul style="list-style-type: none"> - Building a simple GUI application step by step, including widgets, frames, and menus - Hands-on workshops on creating various types of data visualizations like bar graphs, pie charts, and line graphs - Introduction to the machine learning process in Python 	<ul style="list-style-type: none"> - GUI application project - Data visualization project - Machine learning project
4	<ul style="list-style-type: none"> - Grasp the concepts of networking and protocols - Learn to work with URLs and download content from the web - Implement TCP/IP and UDP servers and clients - Explore email communication using Python 	<ul style="list-style-type: none"> - Lectures on TCP/IP, UDP, sockets, and basic network communication concepts - Practical exercises on fetching web content and downloading files using Python - Setting up and running TCP/IP and UDP server-client applications - Hands-on practice for sending emails programmatically 	<ul style="list-style-type: none"> - Quiz on networking and protocols - Web scraping project - Networking project - Sending an email project

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
5	<ul style="list-style-type: none"> - Understand database management systems (DBMS) - Learn to interact with MySQL and Oracle databases - Use stored procedures in Oracle databases 	<ul style="list-style-type: none"> - Lectures on DBMS, advantages over files, and types of databases used with Python - Step-by-step guides and coding exercises for working with MySQL and Oracle databases in Python - Practical sessions on creating and using stored procedures in Oracle databases 	<ul style="list-style-type: none"> - Quiz on DBMS and databases - Database manipulation tasks - Oracle stored procedures project

MAJOR CORE - II

3IDSC503 - Distributed System
(Credit: Theory -4 Practical - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.		
			Theory			Practical			Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work				
3IDSC503	Distributed System	4-0-0	60	20	20	-	-	100	3 hr	-	

Course objectives

1. Understand the fundamentals of Distributed Systems, including their types, architectures, and communication mechanisms.
2. Explore naming, synchronization, and consistency concepts, along with fault tolerance and security principles in Distributed Systems.
3. Familiarize themselves with various categories of Distributed Systems and their respective applications

Course outcome

Understand architecture and communication systems in Distributed Systems. Understand synchronization and various election algorithms in Distributed Systems. Analyze various consistency and replication protocols and methods. Recognize security threats and apply cryptography methods for security in Distributed Systems. Understand various types of Distributed Systems.

Syllabus**Theory****Unit-I**

Fundamentals of Distributed System: Definition of a Distributed System, Goals of a Distributed System, Types of Distributed Systems, Basics of Operating System and Networking. Basics of Architectures, Processes, and Communication: Architectures - Types of System Architectures, Self-Management in Distributed Systems; Processes - Basics of Threads, Virtualization, Roles of Client and Server, Code Migration; Communication - Types of Communications, Remote Procedure Calls, Message-Oriented Communication, Stream-Oriented Communication, Multicasting.

UNIT-II

Naming - Names, Identifiers, And Addresses, Flat Naming, Structured Naming, Attribute Based Naming.

Synchronization - Clock Synchronization, Logical Clocks, Mutual Exclusion, Global Positioning of Nodes, Election Algorithms.

Unit-III

Consistency, Replication and Fault Tolerance-Introduction to Replication, Datacentric Consistency Models, Client-Centric Consistency Models, Replica Management, Consistency

Protocols, Basics of Fault Tolerance, Process Resilience, Reliable Client Server Communication, Reliable Group Communication, Distributed Commit, Recover.

Unit-IV Security: Introduction to Security- Security Threats, Policies, and Mechanisms, Design Issues, Basics of Cryptography, Secure Channels- Authentication, Message Integrity and Confidentiality, Secure Group Communication; Access Control- General Issues in Access Control, Firewalls, Secure Mobile Code, Denial of Service; Security Management-Key Management, Secure Group Management, Authorization Management.

Unit-V Categories of Distributed System: Architecture, Processes, Communication, Naming, Synchronization, Consistency and Replication, Fault Tolerance, Security: Distributed Object-based System; Distributed File System; Distributed Web-based System; Distributed Coordination based System.

Reference books

1. Tanenbaum, A.S. and Van Steen, M., 2007. Distributed systems: principles and paradigms. Prentice-Hall.
2. Sinha, P.K., 1998. Distributed operating systems: concepts and design. PHI Learning Pvt. Ltd.
3. Liu, M.L., 2003. Distributed computing: principles and applications. Pearson Education Inc.
4. Lynch, N.A., 1996. Distributed algorithms. Elsevier.
5. Coulouris, G.F., Dollimore, J. and Kindberg, T., 2005. Distributed systems: concepts and design. Pearson education
6. design. Pearson education

Facilitating the Achievement of Course Learning Objectives

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
1	Understand the fundamentals of Distributed Systems. Recognize various system models in Distributed Systems. Understand Interprocess Communication and Distributed Objects.	- Lectures and discussions on distributed systems, challenges, and web resource sharing. - Overview of architectural and fundamental system models. - Interactive sessions on communication between distributed objects.	Distributed Systems Quiz: Assess understanding. System Models Exploration: Explore system models. Communication Case Study: Analyze Java RMI.
2	Learn about Operating System Support in Distributed Systems. Understand Distributed File Systems and their architecture.	- Lectures on the role of the OS layer, protection, and OS architecture. - Discussions on distributed file systems and their architecture.	OS Support Quiz: Assess knowledge of OS support. File System Analysis: Analyze file service architecture.
3	Explore Peer-to-Peer Systems and their history. Understand Time and Global States in distributed environments. Learn about Coordination and Agreement in distributed systems.	- Lectures on P2P systems, Napster, and P2P middleware. - Sessions on clocks, events, process states, and synchronization. - Discussions on distributed mutual exclusion, elections, and consensus.	P2P Systems Quiz: Assess understanding of P2P systems. Time and States Analysis: Analyze time and states. Coordination Case Study: Analyze coordination concepts.

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
4	Understand Transactions and Concurrency Control in distributed systems. Learn about Distributed Deadlocks and Transaction Recovery.	- Lectures on transactions, concurrency control, and distributed transactions. - Interactive sessions on distributed deadlocks and recovery.	Transactions Quiz: Assess understanding of transactions. Deadlock and Recovery Analysis: Analyze recovery methods.
5	Grasp the concept of Replication and Fault Tolerant Services. Understand Distributed Shared Memory and Consistency Models.	- Lectures on replication, fault tolerance, and transactions with replicated data. - Discussions on distributed shared memory and consistency models.	Replication Quiz: Assess knowledge of replication. DSM and Consistency Exploration: Explore DSM concepts.



MAJOR CORE - III

3IITC503 - Internet of Things

(Credit: Theory -4 Practical - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.	
			Theory			Practical			Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work	Total		
3IITC503	Internet Of Things	4-0-0	60	20	20	-	-	100	3 hr	-

Course objectives

1. To study fundamental concepts of IoT
2. To understand roles of sensors in IoT
3. To Learn different protocols used for IoT design
4. To be familiar with data handling and analytics tools in IoT
5. Appreciate the role of big data, cloud computing and data analytics in a typical IoT system.
6. Understand the role of IoT in various domains of Industry.

Course outcomes

On completion of the course, student will be able to

1. Understand the various concepts, terminologies and architecture of IoT systems.
3. Use sensors and actuators for design of IoT.
4. Understand and apply various protocols for design of IoT systems
5. Use various techniques of data storage and analytics in IoT
6. Understand various applications of IoT
7. Understand APIs to connect IoT related technologies

Syllabus**Theory**

Unit-I Fundamentals of IoT: Introduction, Definitions and Characteristics of IoT, IoT Architectures, Physical and Logical Design of IoT, Enabling Technologies in IoT, History of IoT, About Things in IoT, The Identifiers in IoT, About the Internet in IoT, IoT frameworks, IoT and M2M, Basic Electronic and Digital Electronic, Introduction of embedded System

Unit-I Introduction of embedded System, Microcontroller, features of microcontroller, types of microcontrollers, Microprocessor, Features of Microprocessor, Types of Microprocessors, Microprocessors Microcontroller.

Sensors Networks: Definition, Types of Sensors, Types of Actuators, Examples and Working, IoT Development Boards: Arduino IDE and Board Types, Pin Configuration and architecture Device and Platform features, Concept of digital and analog pin, Arduino interfacing Board.

Raspberry-Pi Development Kit, RFID Principles and component, architecture, Comparison of various Rpi Models, Pin description of Raspberry Pi, on board component of Raspberry Pi, Project Using Raspberry Pi.

Unit-III Wireless Sensor Networks: History and Context, the node, Connecting nodes, Networking Nodes, WSN and IoT.

Wireless Technologies for IoT: WPAN Technologies for IoT: IEEE 802.15.4, Zigbee, HART, NFC, Z-Wave, BLE, BACnet, Modbus.

IP Based Protocols for IoT: IPv6, 6LoWPAN, RPL, REST, AMPQ, CoAP, MQTT.

Edge connectivity and protocols

Unit-V Data Handling and Analytics: Introduction, Bigdata, Types of data, Characteristics of Big data, Data handling Technologies, Flow of data, Data acquisition, Data Storage, Introduction to Hadoop. Introduction to data Analytics, Types of Data analytics, Local Analytics, Cloud analytics and applications

Unit-V Applications of IoT: Home Automation, Smart Cities, Energy, Retail Management, Logistics, Agriculture, Health and Lifestyle, Industrial IoT, Legal challenges, IoT design Ethics, IoT in Environmental Protection.

IoT security

Need for encryption, standard encryption protocol, light weight cryptography, Quadruple Trust Model for IoT-A – Threat Analysis and model for IoT-A, Cloud security.

Reference books

1. Hakima Chaouchi, — The Internet of Things Connecting Objects to the Web, ISBN: 978-1-84821-140-7, Wiley Publications
2. Olivier Hersent, David Boswarthick, and Omar Elloumi, — The Internet of Things: Key Applications and Protocols, Wiley Publications
3. Vijay Madiseti and ArshdeepBahga, — Internet of Things (A Hands-on-Approach), 1st Edition, VPT, 2014.
2. J. Biron and J. Follett, Foundational Elements of an IoT Solution, O'Reilly Media, 2016.
3. Keysight Technologies, “The Internet of Things: Enabling Technologies and Solutions for Design and Test”, Application Note, 2016.

Facilitating the Achievement of Course Learning Objectives

Unit No.	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
1	<p>Understand the fundamentals and characteristics of IoT.</p> <p>Comprehend IoT architectures and design principles.</p> <p>Identify IoT enabling technologies and historical context.</p> <p>Explore the role of embedded systems in IoT.</p>	<p>- Lectures on IoT introduction and definitions.</p> <p>- Discussions on IoT components and architectures.</p> <p>- Hands-on activities with IoT devices and technologies.</p> <p>- Guest lectures on embedded systems and IoT integration.</p>	<p>IoT Basics Quiz: Test understanding.</p> <p>IoT Architecture Presentation: Explain an IoT architecture.</p> <p>IoT Technology Practical: Demonstrate IoT technology.</p>

Unit No.	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
2	<p>Understand the basics of microcontrollers and microprocessors.</p> <p>Identify different types of sensors and actuators.</p> <p>Learn about IoT development boards and their usage.</p> <p>Understand RFID principles and its applications in IoT.</p>	<ul style="list-style-type: none"> - Lectures on microcontroller and microprocessor fundamentals. - Hands-on sessions with sensors and actuators. - Practical exercises with Arduino IDE and Raspberry Pi. - RFID demonstrations and comparisons. 	<p>Microcontroller Quiz: Evaluate knowledge.</p> <p>IoT Development Board Exercise: Use IoT boards effectively.</p>
3	<p>Explore the history and context of wireless sensor networks.</p> <p>Learn about various wireless technologies for IoT.</p> <p>Understand IP-based protocols and their role in IoT.</p> <p>Explore edge connectivity and protocols in IoT.</p>	<ul style="list-style-type: none"> - Lectures on the evolution of wireless sensor networks. - Discussions and comparisons of WPAN technologies. - Practical exercises on configuring IP-based protocols. - Case studies and discussions on edge computing. 	<p>Wireless Sensor Networks Review: Summarize key points.</p> <p>Wireless Tech Presentation: Present IoT wireless tech.</p> <p>Protocol Configuration Task: Configure IoT protocols.</p> <p>Edge Connectivity Report: Analyze edge connectivity.</p>
4	<p>Gain insights into data handling and analytics in IoT.</p> <p>Learn about data acquisition, storage, and Hadoop.</p> <p>Explore various types of data analytics and their applications.</p>	<ul style="list-style-type: none"> - Lectures on big data characteristics and handling technologies. - Hands-on experience with data acquisition and storage tools. - Practical exercises on local and cloud analytics tools. 	<p>Big Data Analysis Quiz: Assess understanding.</p> <p>Data Handling Practical: Manage IoT data.</p>
5	<p>Understand IoT applications across various domains.</p> <p>Explore legal and ethical considerations in IoT design.</p> <p>Recognize the importance of IoT security and encryption.</p> <p>Learn about cloud security and threat analysis in IoT.</p>	<ul style="list-style-type: none"> - Lectures and case studies on IoT applications in different sectors. - Discussions on legal challenges and ethical design principles. - Hands-on activities on IoT security measures. - Guest lectures on cloud security and threat modeling. 	<p>IoT Application Analysis: Analyze IoT use cases.</p> <p>Ethical IoT Design Report: Address ethical concerns.</p> <p>IoT Security Assessment: Evaluate IoT security.</p>

MINOR CORE

3IFDC504 - Fundamentals of Data Science

(Credit: Theory -4 Practical - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted					Duration of Exam.		
			Theory			Practical		Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work			
3IFDC504	Fundamentals of Data Science	4-0-0	60	20	10	-	-	100	3hr	-

Course objective

1. To introduce students to the field of Data Science, providing an overview of its history, development, and terminologies.
2. To impart a strong foundation in mathematics and statistics, enabling students to apply descriptive, predictive, and prescriptive statistics in Data Science.
3. To familiarize students with the role of linear algebra in Data Science and its application in exploratory data analysis and visualization.
2. To educate students about various Data Science techniques, including machine learning, and their broad scope in solving real-world business challenges.

Course outcomes

Understand the principles and applications of Data Science, differentiating it from business analytics and recognizing its significance in modern businesses. Comprehend the role of mathematics, statistics, and exploratory data analysis in Data Science projects, utilizing various statistical measures and visualization techniques. Gain insight into machine learning techniques, including supervised, unsupervised, reinforcement, and deep learning, and their importance in contemporary business environments.

Syllabus**Theory****Unit I: Data Science - An Overview**

Introduction to Data Science, Definition and description of Data Science, history and development of Data Science, terminologies related with Data Science, basic framework and architecture, difference between Data Science and business analytics, importance of Data Science in today's business world, primary components of Data Science, users of Data Science and its hierarchy, overview of different Data Science techniques, challenges and opportunities in business analytics, different industrial application of Data Science techniques.

Unit II: Mathematics and Statistics in Data Science Role of mathematics in Data Science

Importance of probability and statistics in Data Science, important types of statistical measures in Data Science : Descriptive, Predictive and prescriptive statistics, introduction to statistical inference and its usage in Data Science, application of statistical techniques in Data Science,

overview of linear algebra : matrix and vector theory, role of linear algebra in Data Science, exploratory data analysis and visualization techniques, difference between exploratory and descriptive statistics, EDA and visualization as key component of Data Science.

Unit III: Machine Learning in Data Science

Role of machine learning in Data Science, different types of machine learning techniques and its broad scope in Data Science: Supervised, unsupervised, reinforcement and deep learning, difference between different machine learning techniques, brief introduction to machine learning algorithms, importance of machine learning in today's business, difference between machine learning classification and prediction.

Unit IV: Computers in Data Science Role of computer science in Data Science

various components of computer science being used for Data Science, role of relation data base systems in Data Science: SQL, NoSQL, role of data warehousing in Data Science, terms related with data warehousing techniques, importance of operating concepts and memory management, various freely available software tools used in Data Science: R, Python, important proprietary software tools, different business intelligence tools and its crucial role in Data Science project presentation.

Unit V: Data Science Project Management

Data Science project framework, execution flow of a Data Science project, various components of Data Science projects, stakeholders of Data Science project, industry use cases of Data Science implementation, challenges and scope of Data Science project management, process evaluation model, comparison of Data Science project methods, improvement in success of Data Science project models

Reference books

1. Data Science for Dummies by Lillian Pierson.
2. Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking by Foster Provost, Tom Fawcett.
3. Data Smart: Using Data Science to Transform Information into Insight 1st Edition by John W. Foreman. (2015) Wiley Publication.

Facilitating the achievement of course learning objectives

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
I	Understand the concept and history of Data Science. Differentiate Data Science from business analytics. Recognize the importance of Data Science in business. Identify various components and terminologies in Data Science.	- Lectures on Data Science introduction and its development. - Discussions on the distinctions between Data Science and analytics. - Case studies on the impact of Data Science in various industries. - Interactive sessions to explore Data Science terminologies and components.	Data Science History Quiz: Assess historical knowledge. Comparison Report: Write a report on differences. Industry Impact Analysis: Analyze industrial applications. Terminology Quiz: Test terminology comprehension.
II	Understand the role of mathematics and statistics in Data Science.	- Lectures on the significance of math and stats in Data Science. - Practical exercises with statistical analysis tools.	Math and Stats Quiz: Evaluate mathematical understanding.

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities (TLAs)	Assessment Tasks
	Apply descriptive, predictive, and prescriptive statistics. Explore linear algebra's role and EDA in Data Science. Distinguish between exploratory and descriptive statistics.	<ul style="list-style-type: none"> - Hands-on sessions with linear algebra concepts and data visualization tools. - Discussions and practice sessions differentiating exploratory and descriptive stats. 	<p>Statistical Analysis Task: Analyze data using statistics. EDA and Linear Algebra Project: Apply concepts practically. Statistics Comparison Report: Compare the two approaches.</p>
III	Recognize the significance of machine learning in Data Science. Differentiate supervised, unsupervised, and reinforcement learning. Understand the basics of machine learning algorithms. Recognize machine learning's impact on business processes.	<ul style="list-style-type: none"> - Lectures on the role and types of machine learning in Data Science. - Practical demonstrations of various machine learning techniques. - Interactive sessions introducing fundamental ML algorithms. - Case studies showcasing real-world applications of ML in business. 	<p>Machine Learning Quiz: Assess machine learning knowledge. ML Technique Presentation: Present a machine learning technique. Algorithm Exploration Task: Explore ML algorithms. Business Impact Analysis: Analyze business applications.</p>
IV	Learn about the role of computer science in Data Science. Explore database systems and data warehousing in Data Science. Identify various software tools used in Data Science. Understand the role of business intelligence tools in Data Science.	<ul style="list-style-type: none"> - Lectures on computer science components used in Data Science. - Hands-on exercises with SQL, NoSQL, and data warehousing tools. - Demonstrations of R, Python, and other Data Science software tools. - Discussions on BI tools and their significance in project presentations. 	<p>Computer Science Quiz: Assess computer science knowledge. Database and Warehousing Task: Work with data systems. Software Tool Exploration: Explore Data Science tools. BI Tool Presentation: Present a business intelligence tool.</p>

DR. C.V.RAMAN UNIVERSITY



BACHLOR OF COMPUTER APPLICATION

VI Semester

(BCA – CKUG02A03)

(Effective from Academic Year 2023-24)

MAJOR CORE - I

3IWPC603 - Web Programming with ASP.Net using C#

(Credit: Theory -4 Practical - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.		
			Theory			Practical			Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work				
3IWPC603	Web Programming with ASP.NET Using C#	4-0-2	60	20	20	60	40	200	3 hr	2 hr	

Course objective

1. To understand networking and the World Wide Web.
2. Building multi-tier enterprise applications.
3. .NET framework.
2. .NET Interoperation services.
3. Client-side programming: HTTP, CGI, Cookies, JavaScript, HTML, XML.
4. Server-side programming: Web Forms, ASP.NET Web Services, ADO.NET Data Access
5. Client/Server Programming, 3-tier architecture.
6. ASP.NET Web services and web service security.
7. Simple Object Access Protocol (SOAP) and Web Services.

Course outcomes

After successfully completing these course students shall be able:

1. Successful students will able to design web applications using ASP.NET
2. Successful students will be able to use ASP.NET controls in web applications.
3. Successful students will be able to debug and deploy ASP.NET web applications
2. Successful students will be able to create database driven ASP.NET web applications and web services

Syllabus**Theory**

Unit – I Overview of ASP.NET framework, Understanding ASP.NET Controls, Applications

Web servers, installation of IIS.

Web forms, web form controls -server controls, client controls, web forms and HTML, Adding controls to a web form, Buttons, Text Box, Labels, Checkbox, Radio Buttons, List Box, etc.

Running a web Application, creating a multiform web project.

Unit – II Form Validation: Client-side validation, server-Side validation, Validation Controls: Required Field Comparison Range. Calendar control, Ad rotator Control, Internet Explorer Control.

State management- View state, Session state, Application state,

Unit – III Architecture of ADO.NET, Connected and Disconnected Database, Create Connection using ADO.NET Object Model, Connection Class, Command Class, Data Adapter Class, Dataset Class. Display data on data bound Controls and Data Grid.

Database Accessing on web applications: Data Binding concept with web, creating data grid, Binding standard web server controls. Display data on web form using Data bound controls.

Unit – IV Writing datasets to XML, Reading datasets with XML.

Web services: Introduction, Remote method call using XML, SOAP, web service description language, building and consuming a web service, Web Application deployment.

Unit – V Overview of C#, C# and .NET, similarities and differences from JAVA, Structure of C# program.

Language features: Type system, boxing and unboxing, flow controls, classes, interfaces, Serialization, Delegates, and Reflection.

Practical's

1. Working with call backs and delegates in C#.
2. Program to display the addition using the windows application.
3. Creating a Windows Service with C#
4. Using Reflection in C#
5. Sending Mail and SMTP Mail and C#
6. Write a program working with Page using ASP.Net.
7. Write a program working with forms using ASP.NET.
8. Write a program using Required Field Validator in ASP.NET.
9. Write a program using Login Form in ASP.NET.
10. Write a program using Checkbox List in ASP.NET.

Reference books

1. ASP.NET in 3.5 Framework by Ashish Tiwari Kamal Prakashan.
2. ASP.NET 4.0 in Practice by Daniele Bochicchio Dream Tech Press New Delhi.
3. ASP.NET 4.0 in simple steps by Kogent Learning Solutions Dream Tech Press New Delhi.

Facilitating the achievement of course learning objective

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities	Assessment Tasks
1	1. Understand the ASP.NET framework and its controls.	- Lectures on ASP.NET framework and web controls. - Hands-on exercises to create web forms and add controls. - Installation of IIS and web server setup.	-Practical assessment of creating a web project.
	2. Ability to work with server and client-side controls.	- Demonstration of server controls vs. client controls. - Adding various controls to a web form. - Running and testing web applications.	- Quiz on ASP.NET controls and web server setup.
	3. Proficiency in using basic web form	- In-depth exploration of buttons, text boxes, labels, etc. - Creating a multiform web project.	- Practical assessment of control usage in a project.

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities	Assessment Tasks
	controls like buttons, text boxes, labels, etc.	- Code walkthroughs and practice exercises.	
2	1. Implement form validation techniques on both client and server sides.	- Lectures on form validation methods. - Code examples of client-side and server-side validation. - Hands-on practice with validation controls.	- Assignment on form validation implementation.
	2. Utilize various validation controls (e.g., Required Field, Comparison, Range).	- Detailed explanation and usage of validation controls. - Integration of Calendar and Ad Rotator Controls. - Internet Explorer Control demonstration.	- Quiz on validation controls and web control usage.
	3. Manage state using View State, Session State, and Application State.	- Theoretical understanding of state management. - Implementing state management in practical scenarios. - Best practices for state management.	- Practical assessment of state management in web apps.
3	1. Comprehend the architecture of ADO.NET and database connectivity.	- Explanation of ADO.NET concepts and architecture. - Creating database connections using ADO.NET objects. - Hands-on practice with Connection, Command, Data Adapter, and Dataset classes.	- Assignment on database connectivity in web apps.
	2. Implement data binding and display data on web forms.	- Introduction to data binding with web controls. - Creating data grids and binding data. - Displaying data on web forms using data-bound controls.	- Practical assessment of data binding in web apps.
4	1. Work with datasets and XML for data manipulation.	- Writing datasets to XML and reading datasets with XML. - Understanding the importance of data serialization. - Web service basics and concepts.	- Assignment on dataset manipulation and XML handling.
	2. Introduction to web services and their usage.	- Explanation of web services, XML, SOAP, and WSDL. - Building and consuming a web service. - Web application deployment strategies.	- Practical assessment of creating and consuming web services.
5	1. Gain an overview of C# and its relationship with .NET.	- Comparison of C# with Java, exploring language features. - Understanding the structure of C# programs. - In-depth study of type system, classes, interfaces, and delegates.	- Quiz on C# language features and .NET integration.
	2. Explore advanced topics like serialization and reflection in C#.	- Hands-on practice with serialization and deserialization. - Introduction to delegates and reflection in C#. - Real-world examples of these advanced features.	- Assignment on using serialization, delegates, and reflection in C#.

MAJOR CORE - II
3IROM603 - RDBMS with oracle
 (Credit: Theory -4 Practical - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.		
			Theory			Practical			Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work				
3IROM603	RDBMS With Oracle	4-0-2	60	20	20	60	40	200	3 hr	2hr	

Course objective

1. To Identify the advantages of the database approach over the file-based data storage system
2. To understand the architecture of a DBMS and functions of the database system components
3. To understand the features of distributed and object-oriented databases
4. To understand the various operations of PL\SQL

Course outcomes

After study this student will be able to know about the core database administration tasks and tools. Restore database from backups, Import and export data. Monitor SQL Server. To understand the concept of Transaction and Query processing.

Syllabus

Theory

Unit-I

Introduction to DBMS and RDBMS - Introduction to database, Introduction DBMS, Different database models, Structure of DBMS, RDBMS an introduction, Cod's law for RDBMS, Components of RDBMS (kernel/data dictionary)

Introduction to Oracle RDBMS and Client/Server Computing - Introduction to Oracle, The Features of Oracle 9i, the oracle product details, An introduction to client/server computing, Oracle and client/server computing.

Overview of Oracle Architecture - Oracle Architecture, Oracle Files, System and User Processes, Oracle Memory, System Database Object, Protecting Data

Unit-II

Introduction to SQL*PLUS -Introduction to SQL, Features of SQL, Components of SQL, Introduction to SQL*plus, Features of SQL*plus, Execution of SQL*plus, Important commands used in SQL*plus, Oracle Data-Types.

Working with Tables -Tables - An Introduction, Use Of Table In SQL, Viewing The Stored Data In Tables, Filtering Table Data, Updating Data, Deleting Data From Tables, Modifying The Structure Of Tables, Destroying A Table, A Few Other SQL Statements

Data Constraints - Data Constraints, The Use of Data Constraints, The Types of Data Constraints, Defining Integrity Constraints By 'Alter Table', Removing Integrity Constraints, 'Null' Value Concept, 'Not Null' Constraint, Default Value Concept, 'User Constraints' Table

Unit-III

Data Manipulation in SQL - Oracle Operators, Range Searching, Pattern Matching, LIKE 'IN' and 'NOT IN' Predicates, an Introduction to 'DUAL' Table, an Introduction to 'SYSDATE'

Oracle Functions - Oracle Function, Function Types, Group Function, Scalar Function, Working With 'Date' in SQL, Grouping Of Data of Different Tables in SQL Joins, Sub-Queries and Views - types of joins, use of sub-query, 'union' and clause, 'Intersect' Clause, Minus Clause, Concept of View, Types of View, Use of View User Accounts Management and Indexing - Creation of User Account, User Account Management, Granting Privileges, Revoking Privileges, Modifying Password, Closing User Account, Concept of Index, Creation of Index, Types of Index, Use of Index, Deleting Index.

Unit-IV

Introduction to PL/SQL Programming - Introduction to PL/SQL, Advantages of PL/SQL, Differences between SQL and PL/SQL, PL/SQL Block Structure, PL/SQL Character set, Variable, Constant and Data type, Assignment Operator and the use of 'SELECT...INTO', PL/SQL Program Control. Structure, The use of 'IF...THEN...ELSE...ENDIF', Iteration Control (The use of LOOP, WHILE, FOR), The use of 'GOTO Statement.

Cursor - Cursor an Introduction, Types of Cursors, Features of Cursor, Implicit Cursor, Explicit Cursor, Application of for Loop with Cursor

Exception Handling in PL/SQL - Exception Handling in PL/SQL, Built in Exception Handling, User Defined Exception Handling, The Raise Application-error Procedure.

Unit-V

Oracle Transaction - Oracle Transaction, Commit Statement, Rollback Statement, save point statement, Concept of lock, Types of locks, Levels of Locks, 'SELECT...FOR UPDATE' Statement, Removing the Lock.

Procedures and Functions- Concept of Procedures and Functions, Advantages of Procedure and Function, Creation of Procedure and Function, Deleting Procedure and Function

Database Triggers - Concept of Triggers, Types of Triggers, Creation of Triggers, Application of Triggers, Deleting Triggers.

Practical's

1. Write a query to implement Different types of DDL statements in SQL.
2. Write a query to implement Different types of DML statements in SQL.
3. Write a query to implement Different types of DQL statements in SQL.
4. Write a query to implement Different types of DCL statements in SQL.
5. Write a query to explore 'select' clause using where, order by, between, like, group-by, having etc.
6. Write a query to implement the concept of Joins in SQL.
7. Write a query to implement the concept of Indexes and views.
8. Write a query to implement the restrictions on the table.
9. Write a query to implement the concept of Sub Questionaries.
10. Write a query to implement the structure of the table.

Reference books

1. Relational Database Management System, Mahesh Pawar and Goutam Kudale, Vision
2. Publication.
3. Fundamentals of Database Management system, Dr. Mukesh Negi, BPB Publication.
4. SQL, PL/SQL - The Programming Language of Oracle Ivan Bayross
5. Oracle Database 12c: The Complete Reference" by Bob Bryla and Kevin Loney

Facilitating the achievement of course learning objective

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities	Assessment Tasks
I	1. Understand the fundamentals of DBMS and RDBMS. 2. Explain the components and structure of RDBMS. 3. Describe the architecture of Oracle RDBMS. 4. Identify the key features of Client/Server Computing and its relation to Oracle. 5. Recognize the importance of data protection in a database.	- Lectures on Introduction to DBMS and RDBMS. - Group discussions on Codd's Law and RDBMS. - Hands-on practice with Oracle 9i. - Case studies on client/server computing. - Interactive session on data protection.	- Quiz on DBMS and RDBMS concepts. - Group presentation on RDBMS. - Oracle architecture diagram. - Report on Oracle and client/server computing. - Assignment on data protection.
II	1. Familiarize students with SQL and SQL*Plus. 2. Teach the use of tables and data constraints in SQL. 3. Explain data manipulation techniques in SQL. 4. Introduce Oracle functions and their applications. 5. Explore sub-queries, joins, and views in SQL.	- SQL*Plus tutorials and exercises. - Live demonstrations of table operations. - SQL queries practice sessions. - Examples of Oracle functions in action. - Group activities on SQL joins and sub-queries.	- SQL*Plus proficiency test. - SQL table manipulation exercises. - SQL query assessment. - Oracle function-based tasks. - SQL join and view assignments.
III	1. Teach user account management and indexing. 2. Provide an introduction to PL/SQL programming. 3. Explain cursor usage in PL/SQL. 4. Introduce exception handling in PL/SQL.	- Hands-on user account creation and privileges. - PL/SQL code examples and practice. - Cursor-related coding exercises. - Exception handling scenarios and practice.	- User account management tasks. - PL/SQL coding assignments. - Cursor-based programming tasks. - Exception handling exercises.
IV	1. Explore Oracle transactions and locking concepts. 2. Teach the creation and usage of procedures and functions in PL/SQL. 3. Explain the concept of database triggers.	- Transaction management simulations. - Procedure and function creation tasks. - Trigger creation and testing.	- Transaction and lock-related quiz. - Procedure and function assessments. - Trigger implementation projects.

Unit	Course Learning Outcomes (CLOs)	Teaching and Learning Activities	Assessment Tasks
V	<p>1. Provide a deep understanding of Oracle transactions.</p> <p>2. Teach the creation and usage of procedures and functions in PL/SQL.</p> <p>3. Explore the concept of database triggers.</p>	<p>- Transaction-related case studies.</p> <p>- In-depth procedure and function exercises.</p> <p>- Advanced trigger implementation and testing.</p>	<p>- Transaction management report.</p> <p>- Comprehensive procedure and function assessment.</p> <p>- Trigger-based projects and evaluations.</p>



MAJOR CORE - III

IDWC603 - Data Warehousing and Mining

(Credit: Theory -4 Practical - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.		
			Theory			Practical			Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work				
3IDWC603	Data Warehousing and Mining	4-0-0	60	20	20	-	-	100	3 hr	-	

Course objective

1. To understand the scope and necessity of Data Mining and Warehousing for the society.
2. To understand the designing of Data Warehousing so that it can be able to solve the root problems.
3. To understand various tools of Data Mining and their techniques to solve the real time problems. 4. To develop ability to design various algorithms based on data mining tools.

Course outcomes

After study this student will be able to know about the

1. Process raw data to make it suitable for various data mining algorithms.
2. Discover and measure interesting patterns from different kinds of databases.
3. Apply the techniques of clustering, classification, association finding, feature selection and visualization to real world data.

Syllabus**Theory****Unit-I**

Strategic Information Management - Need for strategic information, Decision support system, Knowledge discovery and decision making, need for data warehouse, Definitions of Data warehousing and data mining, Common characteristics of Data warehouse, Data Marts, Metadata, Operational versus analytical databases, Trends and planning of Data warehousing.

Unit-II

Data Modeling Strategy - Defining business requirements, Data modeling strategy, Fact tables, Dimensions, Star schema and other schemas, multi-dimensional data models, Data Cube presentation of fact tables, Using the Data warehouse, Designing tools for Data warehouse, OLAP models and operations

Unit-III

Data Warehouse Architecture Components and, Implementation Options - Architectural components, Infrastructure: Operational and Physical, Extraction, Transformation and Loading, Components of an Oracle Data warehouse, Data Transformation Functions, DBA responsibilities, Capacity Planning.

Unit-IV

Data Warehouse Implementation -Implementation of Data warehouse, Physical design: steps, considerations, physical storage, indexing, Performance Optimization, Data warehouse deployment activities, Data security, Backup and recovery concepts, Data warehouse

Maintenance. Data cube computation, Indexing OLAP data, Efficient processing of OLAP query, OLAP server architectures.

Unit-V

Data Mining - Basics of data mining, Related concepts, Data mining techniques, Data Mining Algorithms, Classification, Clustering and Association rules, Knowledge Discovery in databases (KDD) Process, Introduction to Web Mining

Facilitating the Achievement of Course Learning Objectives

Unit	Course learning outcomes	Teaching and learning activities	Assessment tasks
1	<p>Understand the need for strategic information management.</p> <p>Comprehend the role of decision support systems in data management. -</p> <p>Define knowledge discovery and its importance in decision-making.</p> <p>Recognize the need for data warehousing in modern organizations. -</p> <p>Explain the concepts of data warehousing and data mining.</p> <p>Identify common characteristics of data warehouses and data marts. -</p> <p>Understand the role of metadata in data warehousing. -</p> <p>Differentiate between operational and analytical databases. -</p> <p>Explore trends and planning in data warehousing.</p>	Case studies- Group discussions-	Quizzes- Group presentation on data warehousing trends- Written assignments on data warehousing definitions and characteristics- Final examination on unit content
2	<p>Define data modelling strategy in the context of business requirements. -</p> <p>Understand the components of data modelling, including fact tables and dimensions. -</p> <p>Explain different schema types, such as star schema. -</p> <p>Comprehend multi-dimensional data models and data cube presentation. -</p> <p>Learn to use data warehouse design tools. -</p> <p>Explore OLAP models and operations.</p>	Interactive workshops- Hands-on exercises in data modelling- Case studies on data cube presentation-	Individual or group projects involving data modelling strategy- Data cube design and presentation task- In-class quizzes on schema types- Assessment of understanding through problem-solving exercises
3	<p>Understand the architectural components of data warehousing. -</p> <p>Explore infrastructure requirements, including operational and physical aspects. -</p> <p>Learn about the ETL (Extraction, Transformation, Loading) process. -</p> <p>Familiarize with components of an Oracle Data warehouse. -</p> <p>Recognize data transformation functions and DBA responsibilities. -</p> <p>Understand capacity planning in data warehousing.</p>	Hands-on sessions on ETL processes- Case studies on Oracle Data warehouse components-	Group project on designing a data warehouse architecture- Practical assessment of ETL processes- Individual assignments on DBA responsibilities- Capacity planning exercise and assessment
4	<p>Learn about the implementation of data warehousing. -</p> <p>Explore physical design steps and considerations. -</p> <p>Understand physical storage and indexing in data warehousing. -</p> <p>Optimize performance in data warehouses. -</p> <p>Understand data security</p>	- Workshops on data warehouse implementation- Discussions on data security and backup strategies-	- Group project on implementing a data warehouse- Performance optimization task- Data security and backup plan assignment- Maintenance checklist and assessment

Unit	Course learning outcomes	Teaching and learning activities	Assessment tasks
	and backup and recovery concepts. - Learn about data warehouse maintenance and data cube computation. - Explore indexing in OLAP data and efficient processing of OLAP queries. - Familiarize with OLAP server architectures.		
5	Get introduced to the basics of data mining. - Explore related concepts in data mining. - Understand various data mining techniques and algorithms. - Learn about classification, clustering, and association rules in data mining. - Comprehend the Knowledge Discovery in Databases (KDD) process. - Introduction to Web Mining.	Hands-on sessions with data mining tools- Case studies on real-world data mining applications-	Individual or group projects involving data mining applications- Quizzes on data mining techniques- Final examination on unit content- Presentation on a specific data mining algorithm or application

MINOR CORE

3ICCM604 - Computer and Communication Network

(Credit: Theory -4 Practical - 0)

Scheme of Examination

Course Code	Course Name	Credits	Maximum marks Allotted						Duration of Exam.		
			Theory			Practical			Total	Theory	Practical
			End Sem	Mid Sem	Assignment	End Sem	Term work				
3ICCM604	Computer Communication Network	4-0-0	60	20	20	-	-	100	3 hr	-	

Course objective

1. To understand the fundamental concepts of computer networking.
2. To understand the basic taxonomy and terminology of the computer networking area.
3. To understand the advanced networking concepts, preparing the student for entry Advanced courses in computer networking.
2. To understand the various transition method.

Course outcomes

After study this student will be able to know about

1. Independently understand basic computer network technology.
2. Understand and explain Data Communications System and its components.
3. Identify the different types of network topologies and protocols.
4. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.

Syllabus**Theory****UNIT-I**

Introduction to Networking, Introduction to Network, Network, Computer Networks, Need of Network, Uses of Computer Network, Applications of networks, Network Criteria, Network Hardware and Software, network types: client, server and peers, Various Types of Servers.

Transmission Technology, Transmission technology, Data can be analog or digital, Analog and Digital Transmission, Asynchronous and synchronous transmission, Types of Communication Modes, Baseband and Broadband Transmission, Comparison of Baseband and Broadband Signaling.

Transmission Media, Transmission Media, Classification of Transmission Media, Comparison of Guided and Unguided Media, Twisted Pair (TP) Cable, Coaxial Cable, Fiber Optic Cable (FOC), Unguided Media, Radio Frequency Characteristics, Microwave Transmission, Applications of Infrared Transmission.

Unit-II

Network Topology, Network Topology, Types of Networks, Local Area Network (LAN), Metropolitan Area Networks (MAN), Wide Area Networks (WAN), Satellite Networks, Wireless LAN.

Network Adapters, Network adapters, Network interface cards (NIC), Multiple Access Protocol, ALOHA, Carrier Sense Multiple Access (CSMA), CSMA/CD [Carrier Sense Multiple Access/Collision Detection], Collision Free Protocols, Limited Contention Protocol, Controlled Access, Channelization, Code Division Multiple Access (CDMA) The Theoretical Network Model - OSI, OSI Model, open system interconnection model (OSI) Layered Architecture of the OSI Reference Model, Functions of the ISO/OSI Layers, Summary of OSI Layer functions.

Real World Networks, real world network: Ethernet, Fast Ethernet, FDDI (Fiber Distributed Data Interface), Network Operation, ATM (Asynchronous Transfer Mode), ATM Service Categories, ARCNET, AppleTalk.

Unit-III IEEE 802 Standards, IEEE 802 standards, IEEE 802.3 (CSMA/CD), IEEE 802.4 (Token Bus), IEEE 802.5 [Token Ring], IEEE 802.5 cable standards, Comparison between IEEE 802.3, 802.4 and 802.5, Compare Token Passing with CSMA/CD.

Connectivity Devices, Networking scaling, Connectivity Devices, Modems, Transceiver, Repeaters, Hubs, Bridges, Routers.

Unit-IV TCP/IP Reference Model, Overview of tcp/ip reference model, Introduction to TCP/IP: TCP/IP Protocols, User Datagram Protocol, The Internet Control Message Protocol (ICMP), The Address resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol, Dynamic Host Configuration Protocol (DHCP), Remote Login (rlogin), The Network File System (NFS).

IP Addressing and Subnet, Introduction to IP, Domain Name System (DNS), URL (Uniform Resource Locator), Electronic Mail, E-mail address, Subnet and Subnet masks.

Network Security, Network Security, The Need for Security, common threats, security barriers in network pathways, Attacks, Classification of Attacks, Specific Attacks.

Unit-V Approaches to Network Security, Levels of Security, Approaches to network security, Security Services.

Viruses and Security Threats, Virus and Threats, Malicious Programs, Types of Viruses, Virus Countermeasures, Antivirus Approach, Advanced Antivirus Techniques, Distributed Denial of Service Attacks, DDoS Attack Description.

Firewalls, Firewalls, Firewall Design Principles, Types of Firewalls, Firewall Configurations, Demilitarized Zone (DMZ) Networks, VLAN.

Encryption and Decryption, Encryption and Decryption - Cryptography, Terminology, Classification of Cryptography, Substitution Ciphers, Security of algorithms, Steganography, Steganography vs Cryptography, Public key encryption, Comparison of Symmetric and Asymmetric Key Cryptography, Public Key Cryptanalysis.

Digital Signature, Digital Signature, Requirements of Digital Signature, Direct Digital Signature, Arbitrated Digital Signature, Authentication Protocols, Symmetric Encryption Approach, Public-Key Encryption Approach, Digital Signature Standard, RSA and Digital Signature, DSS Approach, The Digital Signature Algorithm,

Reference books

1. Computer Network "Tanenbaum 5th Edition" Pearson Education India.
2. Computer Networking "James Kurose 7th Edition" Pearson Education India.
3. Data and Computer Communication" by William Stallings

Facilitating the Achievement of Course Learning Objectives

Unit	Course learning outcomes	Teaching and learning activities	Assessment tasks
1	Understand the fundamentals of computer networking. - Identify the types of networks and their uses. - Recognize the criteria for evaluating networks. - Comprehend network hardware and software. - Differentiate between client, server, and peer networks. - Explore various transmission technologies and modes. - Learn about different transmission media.	- Case studies on network applications. - Discussions on network criteria. - Demonstrations of network hardware and software. -	- Quizzes on networking fundamentals. - Written assignments on network types and criteria. - Group presentation on transmission technologies. -
2	Understand network topologies and their types. - Learn about LAN, MAN, WAN, and other network types. - Explore network adapters and multiple access protocols. - Familiarize with the OSI model and its layers. - Examine real-world networks and their operation. - Understand IEEE 802 standards for networking. - Identify various connectivity devices.	- Coding practice for network adapters and multiple access protocols. - Explanation of the OSI model and its layers.-	- Implementation and testing of network topologies. - Problem-solving assignments on network adapters and protocols. -
3	Understand the TCP/IP reference model. - Explore TCP/IP protocols and their functions. - Learn about IP addressing, DNS, URL, and email protocols. - Recognize the importance of network security. - Identify common network threats and attacks. - Learn about approaches to network security.	- Practical exercises on IP addressing and DNS. - Case studies on email and network security. - Demonstrations of security measures.	- Implementation and testing of TCP/IP protocols. - Problem-solving assignments on IP addressing and security measures. - Quizzes on network security concepts. -
4	Understand the basics of encryption and decryption. - Learn about IP addressing and subnetting. - Explore network security and common threats. - Identify different types of viruses and countermeasures. - Examine firewall principles and configurations. - Learn about encryption, decryption, and digital signatures.	Hands-on exercises on IP addressing and subnetting. - Case studies on network security and viruses. - Practical demonstrations of firewalls. - Coding practice for encryption and digital signatures. - Guest lectures on network security.	- Implementation and testing of encryption and decryption. - Problem-solving assignments on IP addressing and firewall configurations. - Quizzes on network security and virus types
5	Explore different approaches to network security. - Learn about viruses, threats, and countermeasures. - Understand the principles of firewalls and their types. - Familiarize with encryption, decryption, and digital signatures.	Case studies on security threats and countermeasures. - Hands-on sessions with firewalls. - Practical exercises on encryption and digital signatures.-.	-Problem-solving assignments on security approaches. - Quizzes on security threats and countermeasures.