

ST JOSEPH'S UNIVERSITY BENGALURU-27



Conferred the status of a Public-Private-Partnership University by the Ministry of Human Resource Development (now MoE), GoI, under *Rashtriya Uchchatar Shiksha Abhiyan* (RUSA) 2.0 and established by the St Joseph's University Act no.24 of 2021

DEPARTMENT OF COMPUTER SCIENCE

SYLLABUS FOR BCA UNDERGRADUATE PROGRAMME

Based on Outcome Based Education (OBE)

For Batch 2024-2027



<u>PART – A</u>		
1	Name of the University	St Joseph's University
2	Objectives of the University	1. Academic Excellence 2. Social Concern 3. Character Formation
3	Vision of the University	To form women and men for and with others, who through holistic education, strive for a just, secular, democratic, and ecologically sensitive society which empowers the poor, the oppressed and the marginalized
4	Mission of the University	In keeping with the Jesuit heritage, the university aims at an integral formation of the staff and the students, to be men and women who will be agents of societal change, by enabling them to attain academic and human excellence in a teaching-learning environment that fosters intellectual curiosity, ceaseless enquiry, personal integrity, social commitment, creativity, critical thinking, and innovation.
5	Name of the Degree	Bachelor of Computer Applications
6	Title of the Academic Programme	Bachelor of Computer Applications
7	Programme Code	Bachelor of Computer Applications
8	Name of the School offering the programme	School of Information Technology
9	Name of the Department offering the programme	Department of Computer Science
10	Vision of the Department offering the programme	The Vision of the Department is to ensure that our students have a deep and analytical understanding of the field and to enable them to use their immense potential to enhance the quality of human life by contributing to the field of Technology, preparing them for their Career in the Industry, and to the needs of the Society with Ethical responsibilities.



11	Mission of the Department offering the Programme	The Department offers quality education in the Science of Computing by providing courses with Computer Applications, Courses in combination with Physical Science and Courses with Specialization and by focusing on the Personal, Emotional and Character development.
12	Description of the Programme	Bachelor of Computer Applications (BCA) The Bachelor of Computer Applications (BCA) is a three-year undergraduate programme designed to provide students with a comprehensive understanding of computer science and its applications. The programme aims to develop skilled professionals capable of meeting the challenges of the rapidly evolving information technology industry. The curriculum offers a balanced blend of theoretical knowledge and practical experience in key areas such as programming, database management, computer networks, software engineering, web technologies, data analytics, and emerging fields like artificial intelligence and cloud computing. Students are encouraged to develop analytical and problem-solving skills through project work, case studies, internships, and hands-on lab sessions. The programme also emphasizes communication skills, teamwork, professional ethics, and continuous learning to prepare graduates for both employment and higher studies. Programme Objectives The BCA programme aims to: 1. Equip students with a solid foundation in computing principles and applications. 2. Develop proficiency in modern programming languages and software tools. 3. Foster analytical, logical, and problem-solving abilities to design and implement computing solutions. 4. Prepare students for careers in software development, systems management, web design, data analysis, and related fields. 5. Promote professional ethics, lifelong learning, and adaptability to emerging technologies.
13	Duration of the Programme	3 years (Six semesters)



14	Total No. of Credits	I Sem - IV Sem → 88 V & VI → 40 + 14 Additional Credits ---- Total : 142 —		
15	Programme Outcomes (POs)			Mention Revised Blooms Taxonomy Level
		PO 1	Acquiring knowledge on basics of Computer Science and applying design principles to develop solutions.	K1, K2, K3
		PO 2	Improved reasoning with mathematical ability to identify, formulate, and analyze problems using data structures and algorithms.	K3, K4, K5
		PO 3	Ability to design and develop algorithmic solutions to real-world problems using multiple programming languages.	K3, K5, K6
		PO 4	Identify, select, and use modern IT tools for modeling, prediction, data analysis, and mobile-based applications.	K3, K4, K5
		PO 5	Acquire good communication skills in both oral and written form.	K3, K6
		PO 6	Practice existing projects and independently launch new projects by identifying requirements in multidisciplinary fields.	K3, K4, K5, K6
		PO 7	Exhibit professional ethics and societal concern regarding computing solutions.	K3, K4, K5
		PO 8	Become an independent learner — learn, unlearn, and relearn.	K5, K6



CREDITS FOR SCIENCES: (Refer Handbook pages 26 to 32)

E.g.

School of Information Technology Credit pattern for BCA under SEP													
Sem	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5	total (Core papers)	Lang 1	Lang 2	Foundation Courses	Open Electives	Practical Knowledge/Skill	Elementary research methodology	Total Credits per semester
Sem 1	4(3+1)	4(3+1)	3	3		14	3	3	F1 - 2				22
Sem 2	4(3+1)	4(3+1)	3	3		14	3	3	F2 - 2				22
Sem 3	4(3+1)	4(3+1)	3	3		14	3	3		OE1 - 2			22
Sem 4	4(3+1)	4(3+1)	3	3		14	3	3		OE2 - 2			22
Sem 5	4(3+1)	4(3+1)	3	3	3	17					2	2	21
Sem 6	4(3+1)	4(3+1)	3	3	3(project)	17					2		19
	24	24	18	18	6	90	12	12	4	4	4	2	128
TOTAL CREDITS	90												
		Additional Credits		No. of Credits					Credit from 6 semesters	128			
		HRD		1					Additional Credits	14			
		Ignitors		3					Total	142			
		Outreach		5									
		Additional Credits		5									
		Total		14									



PART - B

SUMMARY OF CREDITS IN

<u>Semester1</u>	Code Num ber	Title	No. of Hours of Instruct ions	Number of Hours of teaching per week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semeste r Marks	Total mark
Theory	CA1124	Fundamentals of Computers	45	03	03	40	60	100
Theory	CA1224	Programming in C	45	03	03	40	60	100
Theory	CA1324	Mathematical Foundations	45	03	03	40	60	100
Theory	CA1424	Unix Programming	45	03	03	40	60	100
Practical	CA1P124	Information Technology Lab	28	02	01	25	25	50
Practical	CA1P224	C Programming Lab	28	02	01	25	25	50



Semester2	Code Numb er	Title	No. of Hours of Instruct ions	Numbe r of teachi ng hrs /week	Numbe r of credits	Continuou s Internal Assessme nt (CIA) Marks	End Seme ster Mark s	Total marks
Theory	CA2124	Data Structures using C	45	03	03	40	60	100
Theory	CA2224	Object Oriented Concepts using JAVA	45	03	03	40	60	100
Theory	CA2324	Discrete Mathematical Structures	45	03	03	40	60	100
Theory	CA2424	Probability and statistics using R	45	03	03	40	60	100
Practical	CA2P124	Data Structures Lab	28	02	01	25	25	50
Practical	CA2P224	JAVA Lab	28	02	01	25	25	50
Total Number of credits:22(Core papers (14), Languages(6), Foundation Course(2))								



Semester3	Code Number	Title	No. of Hours of Instructions	Number of teaching hrs /week	Number of credit	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CA3125	Database Management Systems	45	03	03	40	60	100
Theory	CA3225	Web Development Technologies	45	03	03	40	60	100
Theory	CA3325	Artificial Intelligence	45	03	03	40	60	100
Theory	CA3425	Computer Organization and Architecture	45	03	03	40	60	100
Practical	CA3P125	DBMS Lab	28	02	01	25	25	50
Practical	CA3P225	Web Development Technologies Lab	28	02	01	25	25	50
Total Number of credits:22(Core papers (14), Languages(6), Open Elective(2))								



Semester 4	Code Number	Title	No. of Hours of Instruct ions	Number of teaching Hrs /Week	Number of credits	Continuou s Internal Assessmen t (CIA) Marks	End Semester Mark s	Total marks
Theory	CA4125	Python Programming	45	03	03	40	60	100
Theory	CA4225	Computer Communication and Networks	45	03	03	40	60	100
Theory	CA4325	Operating System Concepts	45	03	03	40	60	100
Theory	CA4425	Software Engineering	45	03	03	40	60	100
Practical	CA4P125	Python programming Lab	28	02	01	25	25	50
Practical	CA4P225	Computer Networks lab	28	02	01	25	25	50
Total Number of credits:22(Core papers (14), Languages(6), Open Elective(2))								



Semester 5	Code Number	Title	No. of Hours of Instru ctions	Number of teachin g Hrs /week	Nu mb er of cre dit s	Conti nuous Intern al Asses sment (CIA) Marks	En d Se	Tot al Mar ks
Theory	CA5126	E Commerce	45	03	03	40	6 0	1 0 0
Theory	CA5226	Machin e Learn ing Essenti als	45	03	03	40	6 0	1 0 0
Theory	CA5326	Cloud Compu ting	45	03	03	40	6 0	1 0 0
Theory	CA5426	UI/UX Design	45	03	03	40	6 0	1 0 0
Theory	CA5526	Block chain Technology and its Application s	45	03	03	40	6 0	1 0 0
Theory	CA5S126	AWS - Cloud Practition er Essentials	30	02	02	20	3 0	5 0
Practical	CA5P126	UI/UX Lab	28	02	01	25	2 5	5 0
Practical	CA5P226	Machine Learning Lab	28	02	01	25	2 5	5 0
Theory	CARM126	Research Methodolog y and Publication Ethics	30	02	02	20	3 0	5 0



Semester 6	Code Number	Title	No. of Hours of Instructions	Number of teaching Hrs /week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CA6126	Internet of Things	45	03	03	40	60	100
Theory	CA6226	Cyber Security and Cyber Law	45	03	03	40	60	100
Theory	CA6326	Mobile Application Development	45	03	03	40	60	100
Theory	CA6426	Data Analytics	45	03	03	40	60	100
Theory	CA6S126	Power BI	30	03	02	20	30	50
Practical	CA6P126	Cyber Security Lab	28	04	01	25	25	50
Practical	CA6P226	Mobile Application Development Lab	28	04	01	25	25	50
Practical	CAINT626	Project Lab/Internship	28	03	03	40	60	100



CORE COURSES (CC)	
Course Title	Code Number
Fundamentals of Computers	CA1124
Programming in C	CA1224
Mathematical Foundations	CA1324
Unix Programming	CA1424
Information Technology Lab	CA1P124
C Programming Lab	CA1P224
Data Structures using C	CA2124
Object Oriented Concepts using JAVA	CA2224
Discrete Mathematical Structures	CA2324
Probability and statistics using R	CA2424
Data Structures Lab	CA2P124
JAVA Lab	CA2P224
Database Management Systems	CA3125
Web Development Technologies	CA3225
Artificial Intelligence	CA3325
Computer Organization and Architecture	CA3425
DBMS Lab	CA3P125
Web Development Technologies Lab	CA3P225
Fundamentals of Computers	CA4125
Programming in C	CA4225
Mathematical Foundations	CA4325



Unix Programming	CA4425
Information Technology Lab	CA4P125
C Programming Lab	CA4P225
E Commerce	CA5126
Machine Learning essentials	CA5226
Cloud computing	CA5326
Ui/UX Design	CA5426
Block chain technology and Applications	CA5526
UI/UX Design Lab	CA5P126
Machine Learning lab	CA5P226
Research Methodology and publication Ethics	CARM5626
AWS-Cloud Practitioner essentials	CA5S26
Internet of Things	CA6126
Cyber Security and Cyber law	CA6226
Mobile application development	CA6326
Data Analytics	CA6426
Power BI	CA6S26
Cyber Security Lab	CA6P126
Mobile application development Lab	CA6P226
Project/Internship lab	CAINT626
DISCIPLINE SPECIFIC ELECTIVE COURSES (DSE)	
Course Title	Code Number
ELECTIVE COURSES	
Course Title	Code Number
MULTI DISCIPLINARY COURSES	



Course Title	Code Number
Entrepreneurship and Innovation	DAMDE3125
Digital Electronics	DAMDE4125
Internet of Things	DAMDE4225
SKILL DEVELOPMENT COURSES	
Course Title	Code Number
AWS-Cloud Practioner essentials	CA5S26
PowerBI	CA6S26
CERTIFICATE COURSES	
Course Title	Code Number
AI Driven Data course-Foundation for Business Analysts	
AWS-Generative AI services	
Robotics Development	
Interactive Web Design development	
Cyber Analyst	
Python Programming and Exploring ChatGPT	
Monetise your hobby	
Online courses offered or recommended by the department to be listed	
Course Title	Code Number
Project work/Internship related course	Code Number
Any other Course/Field work/Rural Camp/Block Placement	Code Number



SUMMARY OF COURSE ARTICULATION MATRICES

Mapping of Course Outcomes (COs) to Programme Outcomes (POs 1 to 8)

[According to Revised Bloom's Taxonomy Level (BTL)]

Course Code & Title	COs	PO									
		1	2	3	4	5	6	7	8	9	10
CA1124 Fundamentals of Computers	CO 1	*									
	CO 2	*	*	*	*						
	CO 3	*	*	*	*	*					
	CO 4	*	*	*	*	*	*				
	CO 5	*	*	*	*	*	*	*	*		
Course Code & Title	COs	PO									
		1	2	3	4	5	6	7	8	9	10
CA1224 Programming in C	CO1	*			*				*		
	CO2	*	*		*				*		
	CO3	*	*	*	*				*		
	CO4	*	*	*	*		*		*		
	CO5	*	*	*	*		*		*		
Course Code & Title	COs	PO									
		1	2	3	4	5	6	7	8	9	10
CA1324 Mathematical Foundations	CO 1	*	*								
	CO 2	*	*	*							
	CO 3	*	*								
	CO 4	*	*		*						
	CO 5	*				*					
Course Code & Title	COs	PO									
		1	2	3	4	5	6	7	8	9	10
CA1424 Unix Programming	CO 1	*									
	CO 2		*	*							
	CO 3		*	*	*						
	CO 4	*		*	*				*		
	CO 5	*									
Course Code & Title	COs	PO									
		1	2	3	4	5	6	7	8	9	10
CA1P124 Information Technology Lab	CO 1	*									
	CO 2	*	*	*	*						



	CO 3	*	*	*	*	*					
	CO 4	*	*	*	*	*	*				
	CO 5	*	*	*	*	*	*	*	*		
Course Code & Title	COs	PO									
		1	2	3	4	5	6	7	8	9	10
CA1P224 C Programming Lab	CO 1	*	*								
	CO 2	*	*	*							
	CO 3	*	*	*	*						
	CO 4	*	*	*	*						
	CO 5	*	*	*	*				*		
Course Code & Title	COs	PO									
		1	2	3	4	5	6	7	8	9	10
CA2124 Data Structures using C	CO 1	*			*				*		
	CO 2	*	*		*	*		*			
	CO 3	*	*	*	*	*	*		*		
	CO 4		*	*	*	*					
	CO 5	*	*	*			*				
Course Code & Title	COs	PO									
		1	2	3	4	5	6	7	8	9	10
CA2224 Object Oriented Concepts using JAVA	CO 1	*									
	CO 2		*	*							
	CO 3			*							
	CO 4				*				*		
	CO 5										
Course Code & Title	COs	PO									
		1	2	3	4	5	6	7	8	9	10
CA2324 Discrete Mathematical Structures	CO 1	*	*						*		
	CO 2	*	*						*		
	CO 3	*	*						*		
	CO 4	*	*	*					*		
	CO 5	*	*	*	*				*		
Course Code & Title	COs	PO									
		1	2	3	4	5	6	7	8	9	10
CA2424 Probability and statistics using R	CO 1	*									
	CO 2	*	*								



	CO 3		T	T							
	CO 4		T	T	T						
	CO 5			T	T	T			T		
Course Code & Title	COs	PO									
CA2P124 Data Structures Lab		1	2	3	4	5	6	7	8	9	10
	CO 1	*	*	*							
	CO 2	*	*	*	*						
	CO 3	*	*	*	*						
	CO 4	*	*	*	*						
	CO 5	*	*	*	*		*		*		
Course Code & Title	COs	PO									
CA2P224 JAVA Lab		1	2	3	4	5	6	7	8	9	10
	CO 1	*									
	CO 2		*	*							
	CO 3			*							
	CO 4				*				*		
	CO 5										
Course Code & Title	COs	PO									
CA3125 Database Management Systems		1	2	3	4	5	6	7	8	9	10
	CO 1	*									
	CO 2	*	*								
	CO 3	*	*	*							
	CO 4	*	*	*	*						
	CO 5	*	*	*		*					
Course Code & Title	COs	PO									
CA3225 Web Development Technologies		1	2	3	4	5	6	7	8	9	10
	CO 1	*				*					
	CO 2	*	*	*	*	*					
	CO 3	*	*	*	*	*					
	CO 4	*	*	*		*					
	CO 5	*	*	*	*	*	*	*	*		
Course Code & Title	COs	PO									
CA3325 Artificial Intelligence		1	2	3	4	5	6	7	8	9	10
	CO 1	T	T								
	CO 2		T	T							
	CO 3	T	T	T							
	CO 4		T	T	T						



	CO 5		✓	✓	✓	✓	✓				
Course Code & Title	COs	PO									
CA3425 Computer Organization and Architecture	CO 1	*	*	*		*					
	CO 2	*	*	*							
	CO 3	*	*	*	*						
	CO 4	*	*	*	*	*					
	CO 5	*	*	*	*	*	*				
Course Code & Title	COs	PO									
CA3P125 DBMS Lab	CO 1	✓	✓								
	CO 2	✓	✓	✓		✓					
	CO 3	✓	✓	✓		✓					
	CO 4	✓	✓	✓		✓					
	CO 5	✓	✓	✓	✓	✓					
Course Code & Title	COs	PO									
		1	2	3	4	5	6	7	8	9	10
CA3P225 Web Development Technologies Lab	CO 1	*	*			*					
	CO 2	*	*	*	*				*		
	CO 3	*	*	*		*					
	CO 4	*	*		*	*	*				
	CO 5	*	*	*	*	*	*	*	*		
Course Code & Title	COs	PO									
CA4125 Python Programming	CO 1								*		
	CO 2	*	*								
	CO 3	*	*	*							
	CO 4			*	*			*	*		
	CO 5	*	*	*		*					
Course Code & Title	COs	PO									
CA4225 Computer Communication and Networks	CO 1	✓									
	CO 2	✓	✓								
	CO 3		✓	✓							



	CO 4		T	T	T						
	CO 5			T	T	T			T		
Course Code & Title	COs	PO									
CA4325 Operating System Concepts	CO 1	*	*								
	CO 2	*		*							
	CO 3	*	*	*	*						
	CO 4	*		*	*	*					
	CO 5	*	*		*	*	*				
Course Code & Title	COs	PO									
CA4425 Software Engineering	CO 1	*	*								
	CO 2	*		*							
	CO 3	*	*	*	*						
	CO 4	*		*	*	*					
	CO 5	*	*		*	*	*				
Course Code & Title	COs	PO									
CA4P125 Python programming Lab	CO 1	*									
	CO 2		*	*							
	CO 3			*							
	CO 4				*				*		
	CO 5										
Course Code & Title	COs	PO									
CA4P225 Computer Networks lab	CO 1	*	*								
	CO 2	*	*	*							
	CO 3	*	*	*	*						
	CO 4		*	*	*	*					
	CO 5			*		*	*				
Course Code & Title	COs	PO									



COURSE CONTENT AND COURSE OUTCOMES

DEPARTMENT OF COMPUTER SCIENCE

Semester	I
Course Code	CA1124
Course Title	FUNDAMENTAL OF COMPUTERS
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

Course Objective:
1. Introduce the fundamental concepts of computers, including their characteristics, evolution, classification, and architecture. 2. Familiarize students with the internal components of a computer system such as CPU, memory, input-output devices, and microcontrollers. 3. Explain the concepts and functions of operating systems, their classifications, and introduce Unix basics and shell commands. 4. Develop an understanding of database systems, emphasizing the need for databases, SQL basics, and operations on data using DDL and DML commands. 5. Provide knowledge about the Internet, its services, protocols, and the role of Internet Service Providers and the Domain Name System.

<u>UNIT I</u> FUNDAMENTAL OF COMPUTERS	Total Hrs
Introduction to Computers - Computer Definition, Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organization of a Digital Computer; Number Systems – different types, conversion from one number system to another; Computer Codes – BCD, Gray Code, ASCII and Unicode; Boolean Algebra – Boolean Operators with Truth Tables; Types of Software – System Software and Utility Software; Computer Languages - Machine Level, Assembly Level & High Level Languages, Translator Programs – Assembler, Interpreter and Compiler.	06
<u>UNIT II</u> INTRODUCTION TO COMPUTERS	



Characteristics of computers, Classification of Digital Computer Systems: Microcomputers, Minicomputers, Mainframes, Super computers. Anatomy of Computer: Introduction, Functions & Components of a Computer, Central Processing Unit, Microprocessor, Storage units, Input and output Devices. How CPU and memory works. Program execution with illustrative examples. Introduction to microcontrollers.	10
<u>UNIT III</u> OPERATING SYSTEM FUNDAMENTALS	
Operating Systems: Introduction, Functions of an Operating System, Classification of Operating Systems, System programs, Application programs, Utilities, The Unix Operating System, Basic Unix commands, Microkernel Based Operating System, Booting.	08
<u>UNIT IV</u> INTRODUCTION TO DATABASE MANAGEMENT SYSTEMS	
Database, DBMS, Why Database -File system vs DBMS, Database applications, Database users, Introduction to SQL, Data types, Classification of SQL-DDL with constraints, DML, DCL, TCL	10
<u>UNIT V</u> INTERNET AND WEB BASICS	
Introduction, Features of Internet, Internet application, Services of Internet, Logical and physical addresses, Internet Service Providers, Domain Name System. Introduction to web, web browsers, http/https, URL, Introduction to HTML5, Introduction to CSS. Introduction to Emerging technologies: Overview and features of Augmented Reality, Virtual Reality, Artificial Intelligence, Machine learning, Data Warehousing, Data Mining, Big data analytics, Internet of Things.	08

REFERENCES:

1	Pradeep K. Sinha and Priti Sinha: Computer Fundamentals (Sixth Edition), BPB Publication.
2	David Riley and Kenny Hunt, Computational thinking for modern solver, Chapman & Hall/CRC.
3	J. Glenn Brook shear, "Computer Science: An Overview", Addison-Wesley, Twelfth Edition.
4	R.G. Dromey, "How to solve it by Computer", PHI.



BLUEPRINT:

Course Code: CA1124

Course Title: Fundamentals of Computers

Unit Number	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	06	9
Unit II	10	10
Unit III	08	14
Unit IV	10	20
Unit V	08	23
Total	42	76
Maximum marks for the course (Excluding bonus questions) = 60		



Semester	I
Course Code	CA1224
Course Title	PROGRAMMING IN C
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

<u>UNIT I</u> Introduction to C Programming	Total Hrs
Introduction to C Programming: Overview of C; History and Features of C; Planning a Computer Program-Algorithm, Flowchart and Pseudo code with Examples. Structure of a C Program with Examples; Creating and Executing a C Program; Compilation process in C. C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants. Input and output with C: Formatted I/O functions - <i>printf</i> and <i>scanf</i>, control stings and escape sequences, output specifications with <i>printf</i> functions; Unformatted I/O functions to read and display single character and a string - <i>getchar</i>, <i>putchar</i>, <i>gets</i> and <i>puts</i> functions.	10
<u>UNIT II</u>	
C Operators & Expressions: Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associativity; Evaluation of arithmetic expressions; Type conversion. Control Structures: Decision making Statements - <i>Simple if</i>, <i>if_else</i>, <i>nested if_else</i>, <i>else_if ladder</i>, <i>Switch Case</i>, <i>goto</i>, <i>break</i> & <i>continue</i> statements; Looping Statements - Entry controlled and exit controlled statements, <i>while</i>, <i>do-while</i>, <i>for</i> loops, Nested loops.	12
<u>UNIT III</u>	
Derived data types in C: Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays -Declaration, Initialization and Memory representation. Strings: Declaring & Initializing string variables; String handling functions - <i>strlen</i>,	08



<i>strcmp, strcpy and strcat</i> ; Character handling functions - <i>toascii, toupper, tolower, isalpha, isnumeric</i> etc.	
<u>UNIT IV</u>	
User Defined Functions: Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type. User defined data types: Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.	08
<u>UNIT V</u>	
Pointers in C: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Advantages and disadvantages of using pointers; File Operations : File modes, File functions, and File operations, Text and Binary files, Command Line arguments.	06

REFERENCES:

Text Books: C: The Complete Reference, By Herbert Schildt. Kernighan & Ritchie: The C Programming Language (PHI)	
Reference Books:	
1.	P. K. Sinha & Priti Sinha: Computer Fundamentals (BPB)
2.	R.G. Dromey, "How to solve it by Computer", PHI.
3.	E. Balaguruswamy: Programming in ANSI C (TMH)
4.	Kamthane: Programming with ANSI and TURBO C (Pearson Education)
5.	V. Rajaraman: Programming in C (PHI – EEE)
6.	S. Byron Gottfried: Programming with C (TMH)
7.	Yashwant Kanitkar: Let us C
8.	P.B. Kottur: Programming in C (Sapna Book House)



BLUEPRINT:

Course Code: **CA1224**

Course Title: **Programming in C**

Unit Number	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	10	9
Unit II	12	10
Unit III	8	14
Unit IV	6	20
Unit V	6	23
Total	42	76
Maximum marks for the course (Excluding bonus questions) = 60		



Semester	I
Course Code	CA1424
Course Title	UNIX PROGRAMMING
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

Course Objective:
This course will enable students to: • Understand the UNIX Architecture, File systems and use of basic Commands. • Use of editors commands. • Understand Shell Programming and to write shell scripts. • Understand and analyze UNIX System calls, Process Creation, Control & Relationship.

<u>UNIT I</u> INTRODUCTION	Total Hrs
Introduction- Unix Architecture. Features of Unix. The login prompt. General features of Unix commands. Understanding of some basic commands such as echo, printf, ls, who, date, passwd, cal. Meaning of Internal and external commands. The type command, the man command, more command, Knowing the user terminal. The root login. Becoming the super user: su command. Commands to add, modify and delete users.	09
<u>UNIT II</u> UNIX FILES	
Naming files. Basic file types/categories. Parent child relationship. The home directory and the HOME variable. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations represent present and parent directories and their usage in relative path names. File related commands – cat, mv, rm cp, wc and od commands. File attributes and permissions. The ls command with options. Changing file permissions.	09
<u>UNIT III</u> THE VI EDITOR	
Basics- Different ways of invoking and quitting vi. Different modes of vi. Input mode commands. Command mode commands. The ex mode commands. Illustrative examples- Navigation commands. Simple examples using these commands. The shell's interpretive cycle. Wild cards and file name generation. Three standard files and redirection. Connecting commands: Basic and Extended regular expressions. The grep.	08



UNIT IV SHELL PROGRAMMING	
Shell programming. Ordinary and environment variables. Read and read only commands. Command line arguments. exit and exit status of a command. Logical operators for conditional execution. The if, while, for and case control statements. Simple shell program examples.	10
UNIT V FILE INODES AND PROCESS	
File inodes and the inode structure. File links – hard and soft links. Filters- Head and tail commands. Cut and paste commands. The sort command and its usage with different options. Meaning of a process. Mechanism of process creation. Parent and child process. The ps command with its options. Background processes. The bg and fg command, system calls.	09
Self Study: Comparison of different operating systems (UNIX ,ANDROID,IOS)	

REFERENCES:

Text Books: Sumitabha Das., Unix Concepts and Applications., 4th Edition., Tata McGraw Hill Behrouz A. Forouzan, Richard F. Gilberg : UNIX and Shell Programming- Cengage Learning – India Edition. 2009.	
Reference Books:	
1.	M.G. Venkatesh Murthy: UNIX & Shell Programming, Pearson Education.
2.	Richard Blum , Christine Bresnahan : Linux Command Line and Shell Scripting Bible, 2 nd Edition , Wiley,2014.

BLUEPRINT:

Course Code: CA1424

Course Title: UNIX PROGRAMMING

Unit Number	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	09	11
Unit II	09	20
Unit III	08	20
Unit IV	10	11
Unit V	09	20
Total	42	82
Maximum marks for the course (Excluding bonus questions) = 60		



Semester	I
Course Code	CA1324
Course Title	Mathematical Foundations
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

<u>UNIT I</u>	Total Hrs
Basic concepts of logic: Mathematical logic introduction-statements Connectives- negation, conjunction, disjunction- statement formulas and truth tables- conditional and bi Conditional statements- tautology contradiction- equivalence of formulas- duality law-Predicates and Quantifiers, Arguments.	06
<u>UNIT II</u>	
Operations on sets: power set- Venn diagram Cartesian product-relations - functions- types of functions - composition of functions..	10
<u>UNIT III</u>	
Matrix algebra: Introduction-Types of matrices-matrix operations- transpose of a matrix -determinant of matrix - inverse of a matrix- Cramer's rule	08
<u>UNIT IV</u>	
Matrix: finding rank of a matrix - normal form-echelon form cayley Hamilton theorem-Eigenvalues	10
<u>UNIT V</u>	
Differential calculus: Functions and limits - Simple Differentiation of Algebraic Functions – Evaluation of First and Second Order Derivatives – Maxima and Minima	10



REFERENCES:

Text Books: P. R. Vittal-Business Mathematics and Statistics, Margham Publications, Chennai,	
Reference Books:	
1.	B. S. Vatsa-Discrete Mathematics –New Age International Limited Publishers, NewDelhi
2.	Discrete Mathematics and Its Applications, Kenneth H. Rosen: Seventh Edition,2012.

Unit Number	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	09	11
Unit II	09	20
Unit III	08	20
Unit IV	10	11
Unit V	09	20
Total	42	82
Maximum marks for the course (Excluding bonus questions) = 60		



Practical I

Course Code and Title :CA1P124 : Information technology and Unix Lab

(3 hours/11 sessions)

	Part A: Information Technology
1.	Identification of the peripherals of a computer, components in a CPU and their functions.
2.	Activities using Word Processor Software.
3.	Activities using Spreadsheets Software.
4.	Activities using Presentation Software.
5.	Activities involving Multimedia Editing (Images, Video, Audio ...)
6.	Flow charts: Installation and use of flow software for different arithmetic tasks like sum, average, product, difference, quotient and remainder of given numbers, calculate area of Shapes (Square, Rectangle, Circle and Triangle), arrays and recursion.
	Part B: Unix Programs
1.	Implement Unix internal and external command list: man, who, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown.
2.	Implement file and directory related command list: mkdir, rmdir cd, cp , more , cat, mv, rm ,diff,wc, etc command for following structure. <pre> graph TD Root([Root Directory (/)]) --> D1([Directory]) Root --> D2([Directory]) Root --> D3([Directory]) D1 --> F1[File] D1 --> F2[File] D3 --> D4([Directory]) D3 --> D5([Directory]) D4 --> F3[File] D4 --> F4[File] </pre>
3.	Write a shell script program to check whether given input is even or odd.
4.	Write a shell script program to find the largest of three numbers.



5.	Write a script to compute gross salary of an employee according to the rule given below. If basic salary is < 10000 then HRA=15% of basic & DA=45% of basic If basic salary is >=10000 then HRA=20% of basic & DA=50% of basic.
6.	Write a shell script that accept one or more filenames as arguments and convert all of them to uppercase.
7.	Implement regular expression commands for following basic regex: . , ^ , \$, * , \ , () , ?
8.	Implement process based and filter based commands in Unix.

Reference:

1. Computational Thinking for the Modern Problem Solver, By Riley D, Hunt K.A CRC press, 2014
2. Ferragina P, Luccio F. Computational Thinking: First Algorithms, Then Code. Springer

Web References:

<http://www.flowgorithm.org/documentation/>



Practical I

Course Code and Title :CA1P224 : C Programming Lab

(3 hours/11 sessions)

	Part A
1.	Program to read radius of a circle and to find area and Circumference
2.	Program to read three numbers and find the biggest of three
3.	Program to demonstrate library functions in math.h
4.	Program to check for prime
5.	Program to generate n primes
6.	Program to read a number, find the sum of the digits, reverse the number and check it for palindrome
7	Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers
8.	Program to read percentage of marks and to display appropriate message (Demonstration of else-if ladder)
9	Program to find the roots of quadratic equation (demonstration of switchCase statement)
10	Program to read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
11	Program to remove Duplicate Element in a single dimensional Array
12	Program to perform addition and subtraction of Matrices
	Part B:
14	Program to find the length of a string without using built in function
15	Program to demonstrate string functions.
16	Program to demonstrate pointers in C
17	Program to check a number for prime by defining isprime() function
18	Program to read, display and to find the trace of a square matrix



19	Program to read, display and add two m x n matrices using functions
20	Program to read, display and multiply two m x n matrices using functions
21	Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
22	Program to Reverse a String using Pointer
23	Program to Swap Two Numbers using Pointers
24	Program to demonstrate student structure to read & display records of n students.
25	Program to demonstrate the difference between structure & union.



SEMESTER: II

Semester	II
Course Code	CA2124
Course Title	DATA STRUCTURES USING C
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

<u>UNIT I</u> INTRODUCTION	Total Hrs
Introduction to data structures: Definition; Types of data structures - Primitive & Non- primitive, Linear and Non-linear; Operations on data structures. Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - <i>malloc</i> , <i>calloc</i> , <i>realloc</i> and <i>free</i> . Algorithm Specification, Performance Analysis, Performance Measurement-Asymptotic notations.	06
<u>UNIT II</u> ARRAYS	
Arrays: Basic Concepts – Definition, Declaration, Initialization, Operations on arrays; Types of arrays; Arrays as abstract data types (ADT); Representation of Linear Arrays in memory; Traversing linear arrays; Inserting and deleting elements; Sorting – Selection sort, Bubble sort, Quick sort, Insertion sort; Searching - Sequential Search, Binary search; Iterative and Recursive searching; Multidimensional arrays; Representation of multidimensional arrays; Sparse matrices.	10
<u>UNIT III</u> STACKS	
Basic Concepts – Definition and Representation of stacks; Operations on stacks; Applications of stacks; Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression using stack; Application of stack in function calls.	08
<u>UNIT IV</u> QUEUES & LINKED LIST	
Queues: Basic Concepts – Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues; Linked list: Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly linked list, Header linked list, Circular linked	10



list; Representation of Linked list in Memory; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation; Garbage collection.	
UNIT V TREES & BINARY TREE	
Trees: Definition; Tree terminologies –node, root node, parent node, ancestors of a node, siblings, terminal & non-terminal nodes, degree of a node, level, edge, path, depth; Binary tree: Type of binary trees - strict binary tree, complete binary tree, binary search tree and heap tree; Array and pointer representation of binary tree. Traversal of binary tree; preorder, inorder and postorder traversal;	08

REFERENCES:

Text Books:	
1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures	
Reference Books:	
1.	Tanenbaum: Data structures using C (Pearson Education)
2.	Kamathane: Introduction to Data structures (Pearson Education)
3.	Y. Kanitkar: Data Structures Using C (BPB)
4.	Kottur: Data Structure Using C
5.	Padma Reddy: Data Structure Using C
6.	Sudipa Mukherjee: Data Structures using C – 1000 Problems and Solutions (McGraw Hill Education, 2007))

BLUEPRINT:

Course Code: CA2124

Course Title: DATA STRUCTURES USING C

Unit Number	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	06	9
Unit II	10	10
Unit III	08	14
Unit IV	10	23
Unit V	08	20
Total	42	76
Maximum marks for the course (Excluding bonus questions) = 60		



Semester	II
Course Code	CA2224
Course Title	Object Oriented Programming with JAVA
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

<u>UNIT I</u>	Total Hrs
Introduction to Java: Basics of Java programming, Data types, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, Math class, Arrays in java.	06
<u>UNIT II</u>	
Objects and Classes: Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character, String Buffer, File, this reference.	06
<u>UNIT III</u>	
Inheritance and Polymorphism: Inheritance in java, Super and sub class, Overriding, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java, Package in java, UTIL package.	08
<u>UNIT IV</u>	
Event and GUI programming: Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows, Menus, Dialog Box, Applet and its life cycle, Introduction to swing, Exceptional handling mechanism.	10
<u>UNIT V</u>	
I/O programming: Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files.	06
<u>UNIT VI</u>	
Multithreading and Exception handling in java: Thread life cycle and methods, Runnable interface, Thread synchronization, Exception handling with try catch-finally, Collections in java, Introduction to JavaBeans and Network Programming.	06



REFERENCES:

Text Books: “Introduction to Java Programming” by Daniel Liang Programming with Java, By E Balagurusamy – A Primer, Fourth Edition, Tata McGraw Hill Education Private Limited. Core Java Volume I – Fundamentals, By Cay S. Horstmann, Prentice Hall Object Oriented Programming with Java : Somashekara, M.T., Guru, D.S., Manjunatha, K.S	
Reference Books:	
1.	Java 2 - The Complete Reference – McGraw Hill publication.
2.	Discrete Mathematics and Its Applications, Kenneth H. Rosen: Seventh Edition, 2012.

BLUEPRINT:

Course Code: CA2224

Course Title: Object Oriented Programming with JAVA

Unit Number	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	06	
Unit II	06	
Unit III	08	
Unit IV	10	
Unit V	06	
Unit VI	06	
Total	42	
Maximum marks for the course (Excluding bonus questions) = 60		



Semester	II
Course Code	CA2324
Course Title	Discrete Mathematical Structures
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

<u>UNIT I</u> The Foundations	Total Hrs
Logic and proofs: Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy.	10
<u>UNIT II</u> Counting & Advanced Counting Techniques	
Counting: Basics of counting, Pigeonhole principle, Permutation and combination, Binomial Coefficient and Combination, Generating functions-Generating Permutation and Combination. Advanced Counting Techniques: Applications of Recurrence Relations, Solving Linear Recurrence, Relations, Divide and Conquer Algorithms and Recurrence Relations, Inclusion-Exclusion, Applications of Inclusion-exclusion.	12
<u>UNIT III</u> Induction and Recursion & Relation	
Induction and Recursion: Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Corrections. Relation: Properties of relation, Composition of relation, Closer operation on relation, Equivalence relation and partition. Operation on relation, Representing relation.	12
<u>UNIT IV</u> Graphs	
Graphs: Graphs and Graph models, Special Types of Graphs, Sub graphs, Walk, Path and circuit, Fundamental circuit, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs.	08



REFERENCES:

Text Books: Discrete Mathematics and Its Applications, Kenneth H. Rosen: Seventh Edition, 2012.	
Reference Books:	
1.	Discrete Mathematical Structure, Bernard Kolman, Robert C, Busby, Sharon Ross, 2003.
2.	Graph Theory with Applications to Engg and Comp. Sci: Narsingh Deo-PHI 1986.
3.	Discrete and Combinatorial Mathematics Ralph P. Grimaldi, B. V. Ramatta, Pearson, Education, 5 Edition.
4.	Discrete Mathematical Structures, Trembley and Manobar.

BLUEPRINT:

Course Code: CA2324

Course Title: Discrete Mathematical Structures

Unit Number	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	10	
Unit II	12	
Unit III	12	
Unit IV	08	
Total	42	
Maximum marks for the course (Excluding bonus questions) = 60		



Semester	II
Course Code	CA2424
Course Title	Probability and Statistics using R
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

Course Objective:
This course will enable students to: • To introduce the fundamental concepts of statistics and probability. • To develop skills in data analysis and interpretation. • To apply statistical methods to real-world problems. • To understand the role of probability in decision making. • To provide an introduction to R programming for statistical analysis.

<u>UNIT I</u> Introduction to Statistics	Total Hrs
Definition and scope of statistics, Types of data: Qualitative and Quantitative, Data collection methods, Classification and tabulation of data, Graphical representation of data: Bar charts, Histograms, Pie charts, Measures of central tendency: Mean, Median, Mode, Measures of dispersion: Range, Variance, Standard Deviation	10
<u>UNIT II</u> Sampling and Estimation	
Concepts of population and sample, Sampling methods: Random sampling, Stratified sampling, Cluster sampling, Sampling distributions, Central Limit Theorem, Point estimation and interval estimation, Confidence intervals for mean, proportion, and variance, Practical exercises on sampling methods, Calculation and interpretation of confidence intervals	08
<u>UNIT III</u> Probability Theory	
Introduction to probability and its importance, Sample space and events, Classical definition of probability, Axioms of probability, Conditional probability, Bayes' theorem, Independent and dependent events, Problem-solving sessions on probability calculations, Real-life applications of probability.	08



<u>UNIT IV</u> Random Variables and Probability Distributions	
Definition of random variables: Discrete and Continuous, Probability distribution functions, Expected value and variance of random variables, Common probability distributions: Discrete distributions: Binomial, Poisson. Continuous distributions: Normal, Exponential, Calculation of expected value and variance, Exploration of probability distributions with real-world examples	08
<u>UNIT V</u> Introduction to R Programming	
Introduction to R and RStudio, Basic R syntax and data structures, Importing and exporting data in R, Data manipulation using dplyr and tidyr, Basic statistical analysis in R: Descriptive statistics, Probability distributions, Data visualization using ggplot2, Introduction to statistical modeling in R, Hands-on sessions in RStudio, Practical exercises on data manipulation and analysis, Creating and interpreting visualizations using ggplot2	08

REFERENCES:

1.	"Statistics for Business and Economics" by Paul Newbold, William L. Carlson, Betty Thorne
2.	"Probability and Statistics for Engineers and Scientists" by Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying E. Ye
3.	"Introduction to the Practice of Statistics" by David S. Moore, George P. McCabe, Bruce A. Craig
4.	"Probability and Statistics" by Morris H. DeGroot, Mark J. Schervish
5.	"R for Data Science" by Hadley Wickham and Garrett Grolemund "The Art of R Programming" by Norman Matloff.



BLUEPRINT:

Course Code: CA2424

Course Title: Probability and Statistics using R

Unit Number	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	10	11
Unit II	08	20
Unit III	08	20
Unit IV	08	11
Unit V	08	20
Total	42	82
Maximum marks for the course (Excluding bonus questions) = 60		



Practical I : Course Code and Title : CA2P124 : Data Structures Lab

	Part A:
1.	Program to find GCD using recursive function
2.	Program to display Pascal Triangle using binomial function
3.	Program to generate n Fibonacci numbers using recursive function.
4.	Program to implement Towers of Hanoi.
5.	Program to implement dynamic array, find smallest and largest element of the array.
6.	Program to create two files to store even and odd numbers.
7.	Program to create a file to store student records.
8.	Program to read the names of cities and arrange them alphabetically.
9.	Program to sort the given list using selection sort technique.
10.	Program to sort the given list using bubble sort technique.
	Part B:
1.	Program to sort the given list using insertion sort technique.
2..	Program to sort the given list using quick sort technique.
3.	Program to sort the given list using merge sort technique.
4.	Program to search an element using linear search technique.
5.	Program to search an element using recursive binary search technique.
6.	Program to implement Stack.
7.	Program to convert an infix expression to postfix.
8.	Program to implement a simple queue.
9.	Program to implement linear linked list.
10.	Program to display traversal of a tree.



Practical II -Course Code and Title:CA2P224:JAVA Lab (3 hours/11 sessions)



	Practice Programs
1.	Program to print the following triangle of numbers 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5
2.	Program to simple java application, to print the message, "Welcome to java"
3.	Program to create a user defined exception say Pay out of Bounds.
4.	Program to find the area of a rectangle.
5.	Program to display the month of a year. Months of the year should be held in an array.
6.	Program to demonstrate a division by zero exception
	Part A: Java Fundamentals OOPs in Java
1.	Program to assign two integer values to X and Y. Using the 'if' statement the output of the program should display a message whether X is greater than Y.
2.	Program to list the factorial of the numbers 1 to 10. To calculate the factorialvalue, use while loop. (Hint Fact of 4 = 4*3*2*1)
3.	Program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use function overloading.
4.	Program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the super class. MulDiv should have methods to multiply and divide A main function should access the methods and perform the mathematical operations.
5.	Program with class variable that is available for all instances of a class. Use static variable declaration. Observe the changes that occur in the object's member variable values.
6.	Program a. To find the area and circumference of the circle by accepting the radius from the user. b. To accept a number and find whether the number is Prime or not
7.	Program to create a student class with following attributes; Enrollment No: Name, Mark of sub1, Mark of sub2, mark of sub3, TotalMarks. The three marks must be calculated only when the student passes in all three subjects. The pass mark for each subject is 50. If a candidate fails in any one of the subjects, his total mark must be declared as zero. Using this condition write a constructor for this class. Write separate functions for accepting and displaying student details. In the main method create an array of three student objects and display the details.
8.	In a college first year class are having the following attributes Name of the class (BCA, BCom, BSc), Name of the staff No of the students in the class, Array of students in the class
9.	Define a class called first year with above attributes and define a suitable constructor. Also write a method called best Student () which process a first-year object and return the student with the highest total mark. In the main method define a first-year object and find the best student of this class



10.	Program to define a class called employee with the name and date of appointment. Create ten employee objects as an array and sort them as per their date of appointment. ie, print them as per their seniority.
11.	Create a package 'student. Fulltime. BCA 'in your current working directory a. Create a default class student in the above package with the following attributes: Name, age, sex. b. Have methods for storing as well as displaying
	Part B: Exception Handling & GUI Programming
1.	Program to catch Negative Array Size Exception. This exception is caused when the array is initialized to negative values.
2..	Program to handle Null Pointer Exception and use the "finally" method to display a message to the user.
3.	Program which create and displays a message on the window
4.	Program to draw several shapes in the created window
5.	Program to create an applet and draw grid lines
6.	Program which creates a frame with two buttons father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother similar details of mother also appear.
7.	Create a frame which displays your personal details with respect to a button click.
8.	Create a simple applet which reveals the personal information of yours.
9.	Program to move different shapes according to the arrow key pressed.
10.	Program to create a window when we press M or m the window displays Good Morning, A or a the window displays Good After Noon E or e the window displays Good Evening, N or n the window displays Good Night
11.	Demonstrate the various mouse handling events using suitable example.
12.	Program to create menu bar and pull-down menus.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course



Semester	III
Course Code	CA3125
Course Title	Database Management Systems
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

Course Objective:
This course will enable students to: ● To enable the students to learn the designing of database systems, foundation on the relational model of data and normal forms. ● To understand the concepts of database management system, design simple Database model ● To learn and understand to write queries using SQL, PL/SQL. ● To enable the students to learn the designing of database systems, foundation on the relational model of data and normal forms. ● To understand the concepts of database management system, design simple Database models.

CONTENT	Hours
UNIT-1	
Database Concepts: Database and Database Users: Introduction, Characteristics of the Database, Database Users, Advantages of Using DBMS, When Not to Use a DBMS. Database System Concepts and Architecture: Data Models, Schemas, and Instances, Three-schema Architecture and Data Independence, Database Languages and Interfaces, The Database System Environment, Centralized and Client-Server Architectures, Classification of Database Management Systems.	7
UNIT-2	



Data Models: Data Modeling Using Entity-Relationship Model: Using High-Level Conceptual Data Models for Database Design, An Example Database Application, Entity Types, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, Refining the ER Design, Naming Conventions and Design. File organization and storage, secondary storage devices, type of single level ordered index, multi-level indexes. Vector DB: Introduction to Vector Databases, Fundamentals of Vector Search, Popular Vector Databases and Tools.	7
UNIT-3	
RDBMS AND Normalization: Relational Algebra: Relational Model Concepts, Relational Model Constraints, Unary Relational Operations: SELECT, PROJECT, RENAME. Relational Algebra Operations from SET Theory, Binary Relational Operations: JOIN and DIVISION, Examples of Queries in Relational Algebra. Relational Database Design: Anomalies in a database, functional dependency, trivial functional dependency, normal forms – 1NF, 2NF, 3NF, BCNF.	8
UNIT-4	
SQL and PL/SQL: Structured Query Language: SQL Data Definition and Data Types, Basic Queries in SQL CREATE statement, Specifying Constraints in SQL, Basic Retrieval Queries in SQL – SELECT statement, INSERT, DELETE, and UPDATE Statements in SQL, GRANT and REVOKE statements, Complex SQL Retrieval Queries - Comparisons Involving NULL and Three-Valued Logic, Nested Queries, The GROUP BY and HAVING Clauses. Views (Virtual Tables) in SQL - CREATE VIEW, View Implementation, View Update, and Inline Views. Schema Change Statements in SQL – DROP and ALTER command. Relational SET Operators: UNION – UNION ALL – INTERSECTS – MINUS. SQL Join Operators: Cross Join – Natural Join – Join USING Clause – JOIN ON Clause – Outer Join. PL/SQL: introduction to Cursors and Exceptions	12



UNIT-5	
Transaction Processing: Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, Characterizing schedules based on recoverability, Characterizing schedules based on Serializability, Transaction support in SQL. Concurrency Control in Databases: Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Multiversion Concurrency control techniques, Validation Concurrency control techniques, Granularity of Data items and Multiple Granularity Locking.	8
Self Study: NoSQL Databases(MongoDB,Cassandra,CouchDB etc.	

Text Books

1. Abraham Silberschatz, Henry F.Korth and S.Sudarshan,“Database System Concepts”, McGraw Hill International Publication ,VI Edition
2. Ramez Elmasri and Shamkant B. Navathe. “Fundamentals of Database Systems ”, Pearson publication, VII Edition.
3. Raghu Ramakrishnan and Johannes Gehrke, “Database Management Systems”, McGraw-Hill Education, 3rd Edition.

Reference book:

1. Jeffrey A. Hoffer, Ramesh Venkataraman, and Heikki Topi, “Modern Database Management”, Pearson Publication, 13th Edition.
2. Coronel, Morris, Rob, "Database Systems, Design, Implementation and Management", Ninth Edition
3. Nilesh Shah, "Database Systems Using Oracle", 2nd edition, Pearson Education India, 2016
4. Shio Kumar Singh , “Database Systems “,Pearson publications ,II Edition
5. P.Rizwan Ahmed, RDBMS, Margham Publications, 2016 Web Resource.



BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	7	15
Unit 2	7	18
Unit 3	8	15
Unit 4	12	15
Unit 5	8	15
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) = 60		



Semester	III
Course Code	CA3225
Course Title	Web Development Technologies
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

Course Objective:
This course will enable students to: ● To enable the students to learn the designing of database systems, foundation on the relational model of data and normal forms. ● To understand the concepts of database management system, design simple Database model ● To learn and understand to write queries using SQL, PL/SQL. ● To enable the students to learn the designing of database systems, foundation on the relational model of data and normal forms. ● To understand the concepts of database management system, design simple Database models.

CONTENT	Hours
UNIT 1	
HTML: Introduction to HTML, HTML Syntax, Semantic Markup, Structure of HTML Documents Quick Tour of HTML Elements, HTML5 Semantic Structure Elements, formatting tags, Header elements, font tag, List tags, Image & anchor tags. HTML Tables and Forms: Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, Table and Form Accessibility CSS: Introduction to CSS, CSS Syntax, Location of Styles, Selectors, The Cascade: How Styles Interact, The Box Model, CSS Text Styling	
UNIT 2	
JavaScript: Client-Side Scripting, Introduction to JavaScript- Design Principles, Syntax, JavaScript Objects, Syntax-Variables and Data Types-Statements- Operators-Literals-Functions- Objects-Arrays-Built-in Objects -The Document Object Model (DOM), JavaScript Events Forms.	9
Unit-3	



ReactJS : ES6 Overview Introduction to ES6, ES6 lable, Block scope, let &const, Template literals, Arrow functions, Spread and Rest operators, Destructuring Classes - Inheritance, Static properties and methods, Inheritance. Promises, Modules Introduction to React: Introduction, React version history-React 16 vs React 15-Just React - Hello World- Using create-react-app-Anatomy of react project- Running the app- Debugging first react app.	8
Unit-4	
Templating using JSX: Working with React- create Element- Expressions- Using logical operators- Specifying attributes. Specifying children components- Significance of component architecture. Types of components - Functional, Class based, Pure. Working with state and props: Introduction to state and its significance- Read state and set state- Passing data to component using props-Validating props using prop Types. Rendering lists: Using React key prop-Using map function to iterate on arrays to generate elements.	8
Unit-5	
Event handling in React: Understanding React event system Working with Forms Controlled components-Uncontrolled Components Database Connection: Establishing connection, selecting database, Query operations Using MySQL. Open APIs Introduction to APIs, Understanding Open API Specification (OAS), Creating an OpenAPI Definition, Using Open APIs in Applications Object-Relational Mapping Introduction to Object-Relational Mapping (ORM) frameworks Case study: Connect front-end with any data bases.	9



Text Books:

1.HTML, CSS, and JavaScript All in One, Sams Teach Yourself, 3/e byJulie C. Meloni and Jennifer Kyrnin | 8 May 2020

2.Learning React: Modern Patterns for Developing React Apps, Second Edition,2020

References:

1.MASTERING HTML, CSS & Java Script Web Publishing Paperback – 15 July 2016.

2.Web Design with HTML, CSS, JavaScript and JQuery Set Paperback – 15 August 2014.

3.HTML To React: The Ultimate Guide.

BLUEPRINT

Unit	No. of Hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	9	10
Unit 2	8	10
Unit 3	8	22
Unit 4	8	22
Unit 5	9	14
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) =60		



Semester	III
Course Code	CA3325
Course Title	Artificial Intelligence and Applications.
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

Course Objective:

This course will enable students to:

- To impart artificial intelligence principles, techniques and its history.
- To assess the applicability, strengths, and weaknesses of the basic knowledge representation, problem solving, and learning methods in solving real world problems.
- To develop intelligent systems by assembling solutions to concrete computational problems.

<u>UNIT I</u> Artificial Intelligence and its Issues	Total Hrs
Introduction: What is AI?, The foundations of AI, The history of AI, Applications of AI, Intelligent Agents: Agents and Environments, Good Behaviour, The nature of environments, The structure of agents.	08
<u>UNIT II</u> Problem Solving	
Problem solving by Search, Well defined problems and solutions, Real world problems, Problem space - State space, Uninformed Search Strategies-BFS, DFS, Informed (Heuristic) Search Strategies-A* Search, Heuristic Functions, Local Search algorithms and Optimization Problems-Hill climbing Search, Constraint Satisfaction Problems, Means -ends Analysis, Game playing -The mini-max algorithm, Alpha-Beta Pruning.	10
<u>UNIT III</u> Knowledge Representation and Reasoning	
Logical systems, Knowledge Based systems, Propositional Logic-Resolution, Forward and Backward Chaining, Predicate Logic, First Order Logic-Syntax and Semantics of First Order Logic, Inference in First Order Logic-Unification, Forward Chaining, Backward Chaining, Resolution.	10



<u>UNIT IV</u> Learning Systems	
Logical systems, Knowledge Based systems, Propositional Logic-Resolution, Forward and Backward Chaining, Predicate Logic, First Order Logic-Syntax and Semantics of First Order Logic, Inference in First Order Logic-Unification, Forward Chaining, Backward Chaining, Resolution.	04
<u>UNIT V</u> Introduction to Natural Language Processing and Expert Systems	
Natural Language Processing (NLP), Introduction to NLP and Its Importance, Text Preprocessing: Tokenization, Stemming, Lemmatization, Stopword Removal. Applications: Sentiment Analysis, Chatbots, Translation Systems. Expert Systems Stages in the development of an Expert System - Probability based Expert Systems - Expert System Tools - Difficulties in Developing Expert Systems – Applications of Expert Systems.	08

REFERENCES:

Text Books
Poole, D. and Mackworth, A. T Artificial Intelligence: Foundations of Computational Agents, Third Edition, 2023, Cambridge University Russell, S. and Norvig, Artificial Intelligence: A Modern Approach fourth edition 2020
Reference Books
1. Rich, E., Knight K and Shankar, B. Artificial Intelligence, 3rd edition, Tata Mc Graw Hill. 2. Luger, G.F. Artificial Intelligence - Structures and Strategies for Complex Problem Solving, 6th edition, Pearson. 3. Brachman, R. and Levesque, H. Knowledge Representation and Reasoning, Morgan Kaufmann. 4. Alpaydin, E. Introduction to Machine Learning. 2nd edition, MIT Press. 5. Sutton R.S. and Barto, A.G. Reinforcement Learning: An Introduction, MIT Press. 6. Padhy, N.P.. Artificial Intelligence and Intelligent Systems, Oxford University Press.



BLUEPRINT:

Course Code: CA3325

Course Title: Artificial Intelligence and Applications.

Unit Number	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	10	11
Unit II	08	20
Unit III	08	20
Unit IV	08	11
Unit V	08	20
Total	42	82
Maximum marks for the course (Excluding bonus questions) = 60		



Semester	III
Course Code	CA3425
Course Title	Computer organization and Architecture
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

Course Objective:

This course will enable students to:

- Understand the fundamental principles of digital logic design.
- Learn to design and analyze combinational and sequential circuits.
- Gain knowledge of computer organization and architecture.
- Understand central processor organization and its components.
- Develop an understanding of input-output systems and memory organization.

UNIT I INTRODUCTION TO DIGITAL SYSTEMS AND COMPUTER ARCHITECTURE	Total Hrs
Functional Units of a Computer, bus structure, Von Neumann Architecture, Logic Gates- AND, OR, NOT, NAND, NOR, EXOR, truth tables, Boolean expressions and their simplifications, SOP & POS-min-term, max-term & Karnaugh map . Self-Study: Practical applications of logic circuits in modern computing systems..	07
UNIT II COMBINATIONAL AND SEQUENTIAL CIRCUITS	
Half Adder, Full Adder, Multiplexers, De-multiplexers, Encoders, Decoders Flip flops- RS, JK, D, T, binary Counters, shift registers, memory unit-random-access memories (RAM), read-only memories (ROM). Self-Study: Implementation in real-world digital systems and Timing dependencies and clock signals in sequential circuits..	07
UNIT III BASIC COMPUTER ORGANISATION AND DESIGN	
Instruction codes, computer register, computer instruction, timing and control, instruction cycle, memory reference instructions, input output and interrupt, design of basic computer. Self-Study: Examples of instruction flow in simple operations and Concepts of memory addressing and data retrieval methods.	10



<u>UNIT IV</u> CENTRAL PROCESSOR ORGANIZATION	
Introduction, general register organisation, stack organisation, instruction formats, addressing modes, data transfer and manipulation instructions, shift instructions, program control, RISC and CISC. Self-Study: Use of Microinstructions in Modern Processors and Hazards (Data, Control, Structural) and Their Resolution.	10
<u>UNIT V</u> INPUT-OUTPUT ORGANIZATION AND MEMORY ORGANIZATION	
Peripheral devices, I/O interface, asynchronous data transfer, direct memory access (DMA). Interrupts and its types, input-output processor (IOP). Auxiliary memory, microcomputer, memory hierarchy, associative memory, virtual memory, cache memory, memory management hardware. Self-Study: DMA in graphics processing units (GPUs) for high-speed data transfers in gaming and video rendering..	08

Textbook:

"Computer Systems Architecture" by M. Morris Mano and Charles Kime
Publisher: Pearson Edition: 3rd and 5th Revised Edition

Reference Books:

1. "Computer Organization" by Carl Hamacher, Zvonko Vranesic, and Safwat Zaky Publisher: McGraw-Hill Edition: 5th Edition (2002)
2. "Structured Computer Organization" by Andrew S. Tanenbaum Publisher: Pearson Edition: 6th Edition (2012)
3. "Modern Computer Architecture and Organization" by Jim Ledin Publisher: Wiley Edition: 1st Edition (2020)
4. "Computer Organization and Architecture: Designing for Performance" by William Stallings Publisher: Pearson Edition: 10th Edition (2016)
5. "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy Publisher: Morgan Kaufmann Edition: 5th Edition (2013)
6. "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson Publisher: Morgan Kaufmann Edition: 6th Edition (2019)



Practical I -Course Code and Title:CA3P125: DBMS Lab-(3 hours/11 sessions)



	Part A:
1.	Perform the following: i) Creating a Database, ii) Viewing all databases, iii) Viewing all Tables in a Database, iv) Creating Tables (With and Without Constraints), v) Delete Table vi) Rename Table.
2.	Write SQL Queries involving: i) Date Functions, ii) String Functions and iii) Math Functions.
3.	Create a table STATION to store information about weather observation stations with fields: ID(Primary Key), CITY, STATE, LAT, LONG and populate the table STATION with a few rows: i) Write a SQL query to look at table STATION in undefined order ii) Write a SQL query to select Northern stations (latitude > 39.7) iii) Write a SQL query to select only ID, CITY, and STATE columns iv) Write a SQL query to select only ID, CITY, and STATE columns where Longitude > 45.
4.	Create a table STATION to store information about weather observation stations with fields: ID (Primary Key), CITY, STATE, LAT, LONG. Duplicate ID fields are not allowed. Populate the table STATION with a few rows. Create another table called STATS to store normalized temperature and precipitation data: ID field must match some STATION table ID (so that name and location will be known). Allowable ranges will be enforced for other values. No duplicate ID and MONTH combinations. Temperature is in degrees Fahrenheit. Rainfall is in inches. i) Populate the table STATS with some statistics for January and July. ii) Write a SQL query to look at table STATS in undefined order. iii) Write a SQL query to look at table STATS, picking up location information by joining with table STATION on the ID column. iv) Write a SQL query to look at the table STATS, ordered by month and greatest rainfall, with columns rearranged.



5.	Create a table STATION to store information about weather observation stations with fields: ID(Primary Key), CITY, STATE, LAT, LONG. No duplicate ID fields allowed. Populate the table STATION with a few rows. Create another table called STATS to store normalized temperature and precipitation data. • ID field must match some STATION table ID (so name and location will be known). • Allowable ranges will be enforced for other values. • No duplicate ID and MONTH combinations. • Temperature is in degrees Fahrenheit. • Rainfall is in inches. i) Write a SQL query to look at temperatures for July from table STATS, lowest temperatures first, picking up city name and latitude by joining with table STATION on the ID column ii) Write a SQL query to show MAX and MIN temperatures as well as average rainfall for each station iii) Write a SQL query (with sub query) to show stations with year-round average temperature above 50 degrees. • Rows are selected from the STATION table based on related values in the STATS table.
6.	Create a table called STATS to store normalized temperature and precipitation data. • Allowable ranges will be enforced for other values. • No duplicate ID and MONTH combinations. • Temperature is in degrees Fahrenheit. • Rainfall is in inches. i) Create a view (derived table or persistent query) to convert Fahrenheit to Celsius and inches to centimeters ii) Add new column rainfall_centimeter to the table STATS. iii) Insert values into rainfall_centimeter from the view iv) Delete Column rainfall from table STATS
7.	Create table called STATS to store normalized temperature and precipitation data: • Allowable ranges will be enforced for other values. • No duplicate ID and MONTH combinations. • Temperature is in degrees Fahrenheit. • Rainfall is in inches. i) Write a SQL query to look at table STATS in a metric light (through the new view). ii) Write a SQL metric query restricted to January below-freezing (0 Celsius) data, sorted on rainfall.
8.	Create a table called STATS to store normalized temperature and precipitation data.



	i) Update all rows of table STATS to compensate for faulty rain gauges known to read 0.01 inches low ii) Update one row, ID 44's July temperature reading, to correct a data entry error iii) Make the above changes permanent iv) Undo that update
9.	Create a table STATION to store information about weather observation stations with fields: ID(Primary Key), CITY, STATE, LAT, LONG. No duplicate ID fields allowed. Populate the table STATION with a few rows. Create another table called STATS to store normalized temperature and precipitation data. • ID field must match some STATION table ID (so name and location will be known). • Allowable ranges will be enforced for other values. • No duplicate ID and MONTH combinations. • Temperature is in degrees Fahrenheit. • Rainfall is in inches. i) Delete data from STATION table where longitude is >90 ii) Delete July data from STATS table where longitude is >90 iii) Increase the size for the column CITY with the following information:- COLUMNNAME DATATYPE(SIZE) CITY VARCHAR (25) iv) Modify the column name of LONG to LONGITUDE present in the STATION table and verify the result.
10.	Create Table INSTRUCTOR with the following fields: InstuctID, InstructName, Department, Salary. i) Find instructors whose salary is more than the salary of any employee from department "Physics". ii) Find the instructor name and department name of all instructors working in a department with any instructor whose name contains the letter "S". iii) Find the name and department name, of instructors whose salary is more than all their colleagues' salaries in the same department.
11	Create table INSTRUCTOR with the following fields: InstuctID, InstructName, Department, Salary. Create another table called STUDENT with the following fields: StudentID, StudentName, Department, InstuctID i) Find the names of all instructors whose salary is greater than at least one instructor in the Finance department. ii) Find all instructors whose salary is less than the salary of all instructors in the Computer Science department and whose department name is not Computer Science. iii) Find the student name and department name of all student who study in a department with any student whose name contains the letter "S".



12	Consider the following schema for a Library Database: BOOK (Book_id, Title, Publisher_Name, Pub_Year) BOOK_AUTHORS (Book_id, Author_Name) PUBLISHER (Publisher_Name , Address, Phone) BOOK_COPIES (Book_id, Branch_id, No-of_Copies) BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out, Due_Date) LIBRARY_BRANCH (Branch_id, Branch_Name, Address). i) Draw the E_R Diagram for the Library database. ii) Write SQL query to Retrieve details of all books in the library: id, title, name of publisher, authors, number of copies in each branch. iii) Write an SQL query to compute the total number of books based on publisher name order by year.
13	Consider the schema for Movie Database: ACTOR(Act_id, Act_Name, Act_Gender) DIRECTOR(Dir_id, Dir_Name, Dir_Phone) MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id) MOVIE_CAST(Act_id, Mov_id, Role) RATING(Mov_id, Rev_Stars) i) Draw the E_R Diagram for the Movie database. ii) Write an SQL query to find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title. (Use Inner Join) iii) Write an SQL query to retrieve all the actors and any movies they have acted in (Use left Join) iv) Write an SQL query to retrieve all the directors and any movies they have directed.(Use right outer join) (v) Write an SQL query to retrieve all the directors name and all the movies (use Full join)



Practical II

Course Code and Title :CA3P225 : Web Development Technologies Lab

	Part A:
1.	Create your profile page i.e., educational details, Hobbies, Achievement, My Ideals etc. using HTML lists.
2.	Use table tag to format web page. Also create the Time Table of your class using table
3.	Write HTML code to build a Reservation form.
4.	Write HTML code to build an application form.
5.	Create a home page using CSS concepts to add a menu and various text, image and color properties.
6.	Create Style sheet to set formatting for text tags and embed that style sheet on web pages created for your site.
7.	Develop and demonstrate the usage of inline, internal and external style sheet using CSS.
8.	Write a JavaScript program to determine whether a given year is a leap year in the Gregorian calendar
9.	Write a JavaScript program to convert temperatures to and from Celsius, Fahrenheit.
10.	Write <i>JavaScript</i> to validate the following fields of the Registration page. 1. First Name (Name should contains alphabets and the length should not be less than 6 characters). 2. Password (Password should not be less than 6 characters length).
	Part B:
1.	Prepare Project for Weather Finder App using React Js
2..	Prepare Project for Bill and Discount Calculator using React JS.



Semester	IV
Course Code	CA4125
Course Title	Python Programming
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

Course Objective:

This course will enable students to:

- Gain knowledge of slicing , positive and negative indexing
- Understanding Collection and Implementation of control structures
- Learn to organize code using modules and packages for better readability and maintenance.
- Understand core concepts of OOPS and regex
- Handle errors and exceptions gracefully to build robust programs and integrate with real world scenarios.

<u>UNIT I</u> Basics of Python	Total Hrs
Basics of Python program Printing text, printing the result of a calculation, Strings in Python, The Escape Character, Variables and Types, Numeric Data Types in Python, Numeric Operators, Expressions, Integer division, Operator Precedence. Programming examples for each concept. String and Slice: The String Data Type, Negative Indexing in Strings, Slicing, Slicing with Negative Numbers, using a Step in a Slice, Slicing Backwards, String Operators, String Replacement Fields, String Formatting. Understanding in-built string methods programming examples for each concepts.	08
<u>UNIT II</u> Collections and Control statements	
Lists : Create, Access, slicing Negative indexing ,list methods, Tuple : Create, Access, slicing Negative indexing ,tuple methods, Set : creation and operations , Dictionary : Create add and replace values, operation on dictionaries Programming using list and dictionary in-built functions. Conditional blocks	07



&Flow Control using if, else and elif, for loops in, for loop using ranges, use of while loops in python, Loop manipulation using pass, continue, break and else.	
UNIT III Functions, Packages and File handling	
Functions: parameters and arguments, positional arguments, keyword arguments and variable length arguments, local and global scope of variables, recursive functions lambda function Packages : Organizing python codes using functions, organizing python projects into modules, importing own module as well as external modules, Understanding Packages , importing packages . File handling : Reading files in python, writing files in python, Understanding read and write functions, manipulating file pointer using seek, Programming using file operations..	10
UNIT IV Object Oriented Programming and Regular Expression	
Object Oriented Programming : Concept of class, object and instances, constructor, class attributes and destructors, Inheritance, overlapping and overloading operators, Adding and retrieving dynamic attributes of classes. Regular Expressions : Pattern matching and searching, Power of pattern searching using regex in python, Pattern finding programs using regular expression.	10
UNIT V Exception Handling and Python for Data Analysis	
Avoiding code break using exception handling, Safe guarding file operation using exception handling, Handling and helping developers with error code. Introduction to NumPy, NumPy Arrays and Operations, Mathematical and Statistical Functions in NumPy, Introduction to Pandas, Data Handling with Pandas, Data Manipulation in Pandas, Data Analysis and Visualization with Pandas	08

TEXT BOOKS:

1. Allen B. Downey, "Think Python : How to Think like a Computer Scientist", 3rd Edition, O'Reilly Publishers, 2024.
2. Karl Beecher, "Computational Thinking: A Beginners Guide to Problem Solving and Programming", 1st Edition, BCS Learning & Development Limited, 2017.
3. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython by Wes McKinney , Edition: 2nd Edition (October 24, 2017) Publisher: O'Reilly Media



4. Python Data Analytics: With Pandas, NumPy, and Matplotlib by Fabio Nelli 2nd Edition (2023)
publisher Apress

REFERENCES:

1. "Python Programming: A Complete Guide for Beginners to Master, Python Programming Language" by Brian Draper 2016
2. "Python Crash Course" by Eric Matthes is a comprehensive introduction to Python programming, 3rd Edition,
published 2022 by No Starch Press.
3. "Python Programming for Beginners: Python Programming Language Tutorial" by Joseph Joyner 2014

BLUEPRINT:**Course Code: CA4125****Course Title: Python Programming**

Unit Number	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	10	11
Unit II	08	20
Unit III	08	20
Unit IV	08	11
Unit V	08	20
Total	42	82
Maximum marks for the course (Excluding bonus questions) = 60		



Semester	IV
Course Code	CA4225
Course Title	Computer Communication and Networks
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

Course Objective:

This course will enable students to:

- To build an understanding of the fundamental concepts of computer networking.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to network security concepts
- Preparing the student for entry in advanced courses in computer networking.

<u>UNIT I</u> INTRODUCTION	Total Hrs
Communication Network and services, Approaches to Network Design, Network Functions, Key factors in Communication Network, Network Topology, Applications of networks, Line configuration, Layered Architecture and Applications – Examples of Layering, OSI Reference Model, TCP/IP , Telnet, FTP and IP Utilities Self-study: Digital representation of data	08
<u>UNIT II</u> TRANSMISSION SYSTEMS	
Digital transmission Systems – Twisted Pair, Coaxial Cable, Optical Fiber, Multiplexing – frequency – Division , Time – Division , Wavelength Division Multiplexing, Switching- Circuit , Packet and Message. Hamming distance and coding, Checksum, Error Correction and detection, Cyclic redundancy check. Self-study: Unguided media - Radio wave, Microwave, Infrared Transmission systems.	07



<u>UNIT III</u> PEER –TO-PEER PROTOCOLS	
Peer-to peer Protocols and service models, Data Link Control, ARQ Protocols stop and wait, Go-back-N, Selective Repeat, Transmission efficiency of ARQ Protocols, Other adaptation functions, - Sliding window flow control Timing Recovery in Synchronous Services, Reliable Stream Service.	10
<u>UNIT IV</u> LOCAL AREA NETWORKS AND MEDIUM ACCESS CONTROL PROTOCOLS	
Multiple access communications: Local Area network – LAN Structure, MAC Sub layer, Logical link control layer, Random Access protocols ALOHA, Slotted ALOHA, CSMA, CSMA/CD, Scheduling approaches to medium access control – Reservation Systems, polling , Token passing rings, Channelization – FDMA, TDMA, CDMA. LAN Standard – Ethernet and IEEE 802.3. Self-study: IEEE 802.4 standard	07
<u>UNIT V</u> ROUTING AND SECURITY	
Routing algorithms classification, Routing tables, shortest path routing algorithms, Flooding, Hierarchical routing, Distance vector routing Link state routing, congestion control algorithms. Transport layer: TCP and UDP, IP Address, IPV4, IPV6. Network security, ITU-T security Architecture, Three aspects of security, Security attacks, Security services, Security Mechanisms. Cryptography, Symmetric and asymmetric encryption, Caesar Cipher, Play fair Cipher.	10

Text Books:

Stallings, “Data and Computer Communications”, 7th Edition, Pearson Education, 2012 Reference.

References:

1. Andrew S Tanenbaum, “Computer Networks”, 5th Edition, Pearson Education.
2. Behrouz Forouzan, Introduction to Data Communication & Networking TMH, 2019.



3. Larry & Peterson & Bruce S Davis; Computer networks Second Edition, Morgan Kaufman, 2000.
4. Michael E Whiteman and Herbert J Mattord; “Principles of Information Security”, Vikas Publishing House, New Delhi, 2003.
5. Alfred J. Menezes, Paul. C. Van Oorschot, and Scott A. Vanstone “Handbook of Applied Cryptography”, CRC press, Lib of Congress -2018.

Unit Number	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	10	11
Unit II	08	20
Unit III	08	20
Unit IV	08	11
Unit V	08	20
Total	42	82
Maximum marks for the course (Excluding bonus questions) = 60		



Semester	IV
Course Code	CA4325
Course Title	Operating System Concepts
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

Course Objective:

This course will enable students to:

- Understand the basic concepts and functions of an operating system.
- Learn the different types of operating systems and their structures.
- Study process management techniques and process scheduling algorithms.
- Explore memory management methods and virtual memory systems.
- Understand disk and file system management, including scheduling and protection mechanisms.

<u>UNIT I</u> INTRODUCTION	Total Hrs
Definition, functions, operating system structure, Types of operating system: single user, simple batch –spooling , multiprogramming, time – shared , Real time system, distributed system , parallel operating systems, system calls, Types of system calls.	08
<u>UNIT II</u> PROCESS MANAGEMENT	
Process, Process life cycle, Process control block, operations on a process , Inter-process communication, Process scheduling, Scheduling Criteria and Algorithms: FCFS- SJF- Round Robin- priority scheduling, -multilevel queue, multilevel feedback algorithms.	08
<u>UNIT III</u> PROCESS SYNCHRONIZATION AND DEADLOCK	
PROCESS SYNCHRONIZATION Process synchronization –critical section problems, semaphore, Peterson's Solution, classic problems of synchronization- Reader-Writer Problem, Dining Philosophers Problem DEADLOCK:	10



Deadlock Characterization, Methods for Handling Deadlocks, wait for graph and resource allocation graph, Deadlock Prevention, Avoidance and Detection - bankers Algorithm; Recovery from Deadlock.	
<u>UNIT IV</u> MEMORY MANAGEMENT	
Contiguous Memory Allocation, Swapping, Paging, Segmentation, Paged-segmentation, Virtual Memory management: Demand paging, Page replacement algorithms:-FIFO-LRU-optimal. Allocation of frames, page faults, Thrashing.	08
<u>UNIT V</u>	
DEVICE AND FILE SYSTEM: Disk Structure, need for disk scheduling, Scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN,. FILE SYSTEM: -File concept; File Access methods; Directory structure; File system mounting; File Protection.	08

Text Book:

Operating System Concepts (10th Edition) by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, published by Wiley (2018).

References:

1. Operating System by Madnick and Donoval, McGraw Hill.
2. Operating Systems: Internals and Design Principles, by William Stallings, seventh edition
3. Operating System Concepts by James L Peterson. (2nd Edition)
4. Operating System Design and Implementation by Andrew S Tenenbaum. (3rd Edition).

Unit Number	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	08	10
Unit II	08	09
Unit III	10	19
Unit IV	08	19
Unit V	08	19
Total	42	76
Maximum marks for the course (Excluding bonus questions) = 60		



Semester	IV
Course Code	CA4425
Course Title	SOFTWARE ENGINEERING
Number of teaching hours per week	3
Total number of teaching hours per semester	42
Number of credits	03

Course Objective:

This course will enable students to:

- To inculcate in student's different concepts of software engineering principles
- To analyse the different categories of requirements in a software project, develop the skills necessary to design, develop and execute software projects.
- To comprehend the aspects of software Quality, software maintenance and various testing strategies
- To get an overview of the total management spectrum from the perspective of a software project.

<u>UNIT I</u>	Total Hrs
SOFTWARE PROCESS AND SOFTWARE PROCESS MODELS Introduction: Software process, concepts of project and product, process, Life cycle models: Waterfall model, spiral model, Prototyping Model, Incremental model, RAD model, Agile framework: values and principles of Agile, Agile manifesto, Extreme programming, Scrum Framework	10
<u>UNIT II</u>	
SOFTWARE REQUIREMENTS Software requirements: Functional- non-functional requirements, User requirement, Software Requirement engineering, Feasibility studies, Requirements elicitation and analysis, requirement validation, software requirement specification (SRS document).	06
<u>UNIT III</u>	



SOFTWARE DESIGN AND PROJECT MANAGEMENT Software Design: Design Process, Design strategies, Design characteristics, architectural design: System Structuring, control modelling, Modular decomposition, Data Flow Diagrams. Software Project Management & Cost estimation Technique Software Project Management: Software cost estimation techniques: top down, bottom-up approach, estimation by analogy, COCOMO model, LOC	08
<u>UNIT IV</u>	
SOFTWARE CODING AND SOFTWARE TESTING Software Coding: Introduction, coding techniques, styles, Coding verification techniques, code documentation. Software Testing: Introduction, verification and validation, A strategic approach to software testing, Unit testing, Black-box and white box and integration testing, system testing	08
<u>UNIT V</u>	
SOFTWARE QUALITY AND SOFTWARE MAINTENANCE Software Quality: Introduction, quality metrics, quality assurance: tasks, goals, quality planning, quality control, software reliability and software reuse. Software Maintenance: Introduction, corrective maintenance, adaptive maintenance, perfective maintenance, Preventive maintenance Introduction to DevOps , Advantages of adopting DevOps practices	10

Text books:

1. Roger S. Pressman, “Software Engineering: A Practitioner’s Approach”, Alternate Edition, 7th Edition, McGraw Hill.
2. Ian Somerville, “Software Engineering”, 8th Edition, Pearson Publication Ltd 2012.

Reference books:

1. Software Engineering, an Engineering approach- James F. Peters.

**Practical I****Course Code and Title :CA4P125 : Python programming Lab****(3 hours/11 sessions)**

	Part A:
1.	Write a Python program to create a list of n numbers entered by the user and find the largest and smallest numbers in the list.
2.	Write a Python program to count the number of occurrences of an element in a list. Tuples and Dictionaries
3.	Write a Python program to create a tuple with at least 5 elements of your choice. Implement the following functionalities: a) Print all the elements of the tuple using a loop. b) Allow the user to input a value and check if it exists in the tuple. If it exists, display the index of the value; otherwise, print a message saying the value is not found. c) Display the total number of elements in the tuple.
4.	Write a Python program to create a Dictionary-based Phonebook to store names as keys and phone numbers as values and Implement functionalities to add, search, and delete contacts from the phonebook using dictionary operations.
5.	Write a Python program to count the frequency of each character in a string.
6.	Write a Python program to remove duplicate elements from a list of strings without using built-in methods.
7	Write a Python program to reverse a string using slicing and loop methods.
8.	Write a Python program to implement a stack.
9	Write a Python program to implement a queue.
10	Write a Python module named currency_converter that includes functions to convert amounts between USD, EUR, and INR using predefined exchange rates.
11	Write a Python program to demonstrate the use of default and keyword arguments.
12	Write a Python program to create a module with mathematical operations . Import



	and use the module in a separate script.
13	Create a package with two modules: a) To convert <code>celsius_to_fahrenheit(celsius)</code> b) To convert <code>fahrenheit_to_celsius(fahrenheit)</code>
14	Write a Python program to create a text file, write data into it, use the seek and append methods .
15	Write a Python program to write a list of strings into a file and then read it back.
16	Write a Python program to create a Book class, which will represent a book in the library , create objects of that class to store information about different books.
17	Write a program to define a class Car with a constructor that initializes make, model, year, and price. Add a method to display car details. Then, write a program to create a list of cars and print details of all cars whose price is greater than a given value.
18	Design a simple E-commerce system using Python. You will create a base class called Product and a derived class called ElectronicsProduct that inherits from Product. The base class will store general information about a product, while the derived class will add extra details specific to electronic products.
19	Create a Python program to simulate a vehicle management system with a base class Vehicle. Then create derived classes Car, Motorcycle, and Truck that inherit from Vehicle. The Vehicle class should have basic details like speed and fuel, while each derived class should have its own unique methods such as <code>cargo_capacity</code> for Truck and <code>wheel_type</code> for Motorcycle.
20	Create a Python program to overload a method that can perform mathematical operations on varying numbers of arguments.
21	Create a Complex class that supports basic complex operations. Overload the + and - operators to allow vector objects to be added or subtracted
22	Write a Python program to validate a password entered by the user. The program should ensure the password meets the following criteria: · The password must be at least 8 characters long. · The password must contain at least one uppercase letter. · The password must include at least one digit. · The password must have at least one special character
23	Validate an email address format and handle invalid input.
24	Design and implement a Python program that simulates a simple banking system. The program should allow a user to check their balance, deposit money, and withdraw money.



	Use exception handling to manage the following scenarios: 1. Insufficient Funds: Raise a custom exception <code>InsufficientFundsError</code> if the user tries to withdraw an amount greater than their current balance. 2. Invalid Input: Handle errors when the user enters invalid data (e.g., entering text instead of numbers for deposit or withdrawal amounts).
25	Write a Python program that prompt the user to enter the name of a file. The program should try to open and read the file. If the file does not exist, it should catch the <code>FileNotFoundError</code> and display a message: "File not found"
26	Write a Python program to perform Basic Arithmetic Operations on two NumPy arrays and Applying Universal Functions in NumPy
27	Write a Python program to plot line graphs, histograms, bar charts, and scatter plots using the <code>plot()</code> function in Pandas.

Practical I

Course Code and Title :CA4P225 : Computer Networks Lab

(3 hours/11 sessions)

	Part A:
1.	Implement the functionality of various network devices such as Repeater, Hub, Switch, Router, Digitizer, Bridge, and Gateway.
2.	Implement IP Addressing, Subnetting, and Super netting to configure IP addressing schemes
3.	Design and Connect Computers in a Local Area Network (LAN)
4.	Execute basic commands like <code>ping</code> , <code>tracert</code> , <code>ipconfig</code> , and <code>nslookup</code> to analyze and test network connectivity.
5.	Write a socket program for the implementation of echo.
6.	Write a program for error detecting code using CRC-CCITT (16- bits).
7.	Write a program for error detecting code using 2D Parity check.
8.	Write a program to find the shortest path between vertices using Dijkstra's Algorithm.



9	Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.
10	Write a program for congestion control using leaky bucket algorithm.
11	Write a program for congestion control using Token bucket algorithm.
12	Write a program on datagram socket for client/server to display the messages on client side, typed at the server side.
13	Write a program to implement the Token Passing algorithm.
14	Write a program to encrypt and decrypt a Password
15	Implement the substitution mono alphabetic technique by using Caesar Cipher algorithm.
16	Write a program to implement Play fair Cipher algorithm.

**SEMESTER-V**

Course Code: CA 5126	Course Title: E Commerce
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs.

Course Objectives:

1. To introduce the fundamentals and evolution of E-Commerce.
2. To explain the key components and models of online business systems.
3. To provide insights into building secure and effective E-Commerce platforms.
4. To explore digital marketing tools and supply chain integration in E-Commerce.

Course Outcomes (COs)

1. Describe the evolution, features, and scope of E-Commerce.
2. Explain B2C, B2B, and crowdfunding business models and their strategic components.
3. Apply SWOT analysis and identify threats and management practices in E-Commerce systems.
4. Analyse digital marketing strategies, consumer behaviour, and advertising technologies.
5. Evaluate the role of supply chain and collaborative commerce in enhancing business efficiency.

CONTENT	Hours
UNIT – 1 INTRODUCTION TO E-COMMERCE Introduction, Difference between E-Commerce and Traditional Commerce, M-Commerce and its Role in E-Commerce, Unique features of E-Commerce technology, Types of E-Commerce, Historical Evolution of E-Commerce, Understanding E-Commerce: Organizing themes. <i>Case Study: Pinterest – A Picture is Worth a Thousand Words</i>	6
UNIT – 2 E-COMMERCE BUSINESS MODELS Introduction, Eight Key Elements of a Business Model, Crowdfunding, Types of Crowdfunding, Major Business-to-Consumer (B2C) Business Models: E-Tailer, Community Provider, Content Provider, Portal, Transaction Broker, Market Creator, Service Provider. Major Business-to-Business (B2B) Business Models: E-Distributor, E-Procurement, Exchanges, Industry Consortia, Private Industrial	10



Networks. Impact of E-Commerce in Business: Strategy, Structure and Process. <i>Case Study: Freemium – Duolingo</i>	
UNIT – 3 E-COMMERCE PRESENCE AND SECURITY E-Commerce Presence Map, Timeline, SWOT Analysis, Building an E-Commerce Presence: A Systematic Approach, The E-Commerce Security Environment, Security Threats in E-Commerce, Technology Solutions, Management Policies, Business Procedures and Public Laws, Payment System, E-Commerce Payment System, Electronic Presentment and Payment. <i>Case Study: Bitcoin</i>	10
UNIT – 4 E-COMMERCE MARKETING AND ADVERTISING CONCEPTS Consumer Online: Internet Audience and Consumer Behaviour, Digital Commerce Marketing and Advertising: Traditional Online Marketing and Advertising Tools, Social, Mobile and Local Marketing and Advertising, Internet Marketing Technologies: Web Transaction Logs, Supplementing the Logs, Databases: Data Warehouses, Data Mining and Big Data, Marketing Automation and Customer Relationship Management (CRM) Systems, Understanding the Costs and Benefits of Online Marketing Communications.	8
UNIT – 5 SUPPLY CHAIN MANAGEMENT AND COLLABORATIVE COMMERCE An Overview of B2B E-Commerce, Procurement Process and Supply Chain, Trends in Supply Chain Management and Collaborative Commerce, E-Logistics and Supply Chain Integration, Role of Cloud Computing and IoT in SCM.	8

Textbook

- Kenneth C. Laudon and Carol Guercio Traver: *E-Commerce: Business, Technology, and Society*, Pearson, 2024.

Reference Books

1. Daniel Amor: *E-Business R(Evolution)*, Pearson Education, 2020
2. David Whitely: *E-Commerce: Strategy, Technologies and Applications*, Tata McGraw-Hill, July 2010



Course Code: CA 5226	Course Title: Machine Learning Essentials
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives:

- 1.To understand the basic concepts of Machine Learning.
- 2.To understand and build the supervised and unsupervised learning models.
- 3.To learn and understand the concept of neural networks and deep learning.

Course Outcomes:

- 1.Identify the basic concepts of Machine Learning and Training models.
- 2.Identify and apply the appropriate machine learning techniques for classification.
- 3.Analyze and apply unsupervised learning techniques for solving real time problems.
- 4.Analyze the concept of Neural Network.
5. Implement fundamental machine learning algorithms by utilising core Python libraries— NumPy, Pandas, Matplotlib, and Scikit-learn.

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	*							
CO2	*	*	*					
CO3	*	*	*					
CO4	*	*	*					
CO5	*	*	*	*		*		

Unit 1	Introduction to Machine Learning	8 hours
Fundamentals of Machine Learning – Applications -Types of Machine Learning – Challenges of Machine Learning – Testing and Validating. Training a ML model-End-to-End Machine Learning Project – Working with Real Data – Get the Data – Explore and Visualize the Data – Prepare the Data for Machine Learning Algorithms,Cross-Validation and Resampling Method,Measuring Classifier Performance.		



Unit II	Classification and Regression	8 hours
Supervised Learning-Regression –Linear Regression – Logistic Regression -Support Vector Machine – Naive Bayes – Decision Tree – KNN algorithm .		
Unit III	Unsupervised Learning	8 hours
Introduction to clustering- Categories of Clustering-K-means clustering - Limits of K-means– Hierarchical clustering- agglomerative Clustering-Divisive Clustering-Expected maximization Algorithm. Dimensionality Reduction: Introduction, subset selection, principal component analysis(PCA) .		
Unit IV	Artificial Neural Network	8 hours
Biological to Artificial Neurons – Logic Computations with Neurons – Perceptron - Multilayer Perceptron and Back propagation		
Unit V	Python libraries for machine Learning	10 hours
Numpy: Ndarray, Array attributes Array creation routines, Indexing and slicing, Array Broadcasting, Array manipulation, Mathematical functions, Statistical functions, Search, sort and counting functions, Matrix Library, Linear algebra. Pandas: Series, Data frame, Panel, Basic functionality: axes, dtype, empty, ndim, size, values, head, tail. Descriptive Statistics, Reindexing, iterations, sorting, options and Customization, Indexing and Selecting Data, Statistical Functions, Window Functions, Aggregation, Missing data, Group by, Merging, concatenation, Categorical data, I/O tools: read_csv, read_table. Data Visualization Matplotlib: Barplot, Histograms, Box plots, Area plot, Scatter plot, Pie chart. Scikit: Classifiers: K-nearest, SVM, Naive base, Linear Regression. Clustering: K Means, Spectral, Hierarchical, DBSCAN, OPTICS. Dimensionality Reduction: PCA		
Text Book(s)		
1.	<u>AurelienGeron, “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”, 2019, 2nd Edition,O'Reilly Media, Inc.</u>	
2.	EthemAlpaydin, “Introduction to Machine Learning”, 2020, Fourth Edition, MIT Press.	
Reference Books		
1.	Stephen Marsland, “Machine Learning: An Algorithmic Perspective “,2014, Second Edition”, CRC Press.	
2.	Tom M. Mitchell,”Machine Learning”,2021,by McGraw-Hill.	



Course Code: CA 5326	Course Title: Cloud Computing
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hours

Course Objectives

1. **Understand the fundamentals of cloud computing**, including its history, vision, core characteristics, deployment models, and the differences between traditional on-premises and cloud-based environments.
2. **Analyze cloud computing architecture and service models (IaaS, PaaS, SaaS)**, including their evolution, integration strategies, and the challenges associated with deploying and managing these services.
3. **Explore virtualization technologies** as the backbone of cloud computing, understanding their characteristics, techniques, advantages, limitations, and role in resource provisioning and migration.
4. **Examine energy-efficient and market-oriented approaches** to cloud computing, including green cloud architectures, federated clouds, and economic models for cloud service delivery.
5. **Gain practical exposure to leading cloud platforms**, particularly Amazon Web Services (AWS), focusing on key services in compute, networking, storage, and security, while also encouraging comparative analysis with other platforms like Microsoft Azure.

Course Outcomes (COs):

CO1 : Understand the fundamental concepts, characteristics, and deployment models of cloud computing, and differentiate between traditional on-premises computing and cloud environments.

CO2 : Analyze cloud computing architecture and service models (IaaS, PaaS, SaaS), and evaluate tools and technologies used for cloud resource provisioning and management.

CO3 : Explain virtualization technologies, classify various virtualization techniques, and assess their role and impact on cloud computing.



CO4 : Evaluate green and market-oriented cloud computing architectures, including energy-efficient strategies and federated cloud environments.

CO5 : Explore real-world cloud platforms with a focus on Amazon Web Services (AWS) and gain awareness of other platforms such as Microsoft Azure through self-study.

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	*							
CO2	*	*						
CO3	*							
CO4	*	*	*					
CO5	*	*	*	*		*		

Content	Hours
Unit – 1	
Introduction to Cloud Computing Introduction to Cloud Computing, History of Cloud Computing, Vision of cloud computing, On-Premises vs. Cloud, Characteristics and Benefits of Cloud Computing, Cloud Infrastructure Management, Challenges and Risks, Building cloud computing environments, Cloud Deployment Models: public, private, hybrid and community. Self-Study : Comparison of cloud deployment models	8
Unit-2	
Cloud Computing Architecture The cloud computing reference model, Introduction to IaaS, PaaS and SaaS, Evolution of SaaS, Challenges of SaaS Paradigm, SaaS Integration Services, Introduction to Platform As a service (PaaS), Integration of Private and Public Cloud, Technologies and Tools for Cloud Computing, Resource Provisioning services, Introduction Infrastructure As a Services (IaaS), Background and Related Work, Virtual Machines Provisioning and Manageability, Virtual Machine Migration Services, VM Provisioning and Migration in Action, XaaS Service Architecture.	8
Unit-3	
Virtualization Introduction to Virtualization Technologies, Characteristics of virtualized environments, Taxonomy of virtualization techniques, Virtualization and cloud computing, Pros and cons of virtualization, Technology examples	8



Unit-4	8
“Green” cloud computing architecture Energy efficiency in clouds, Design Principles for Energy-Efficient Cloud Architecture, green cloud computing architecture, . Market-oriented cloud computing Market-based management of clouds, Market-oriented cloud computing, A reference model for MOCC. Federated clouds/InterCloud, Characterization and definition, Cloud federation stack Self-Study : Cloud Applications	
UNIT 5	10
Cloud Platforms in Industry Amazon web services: Security services, Networking services, Compute services, Lambda, storage services. Self-Study: Difference between Microsoft Azure and Amazon web services.	

Text Book

- Thomas Erl, Eric Barceló Monroy, "Cloud Computing: Concepts, Technology, Security & Architecture", 2nd edition, 2024, Pearson Education
- [Rajkumar Buyya](#) , [Christian Vecchiola](#) , [S.ThamaraiSelvi](#) , “Mastering Cloud Computing: Foundations and Applications Programming”, Morgan Kaufmann Publishers In, 2013
- [Rajkumar Buyya](#) , [James Broberg](#) , [Andrzej Goscinski](#), “Cloud Computing: Principles and Paradigms”, First Edition, 2011, willey

Reference Books

- Thomas Erl, Zaigham Mahmood, and Ricardo Puttini,” Cloud Computing Concepts, Technology & Architecture”, PRENTICE HALL, 2013
- Anthony T .Velte, Toby J.Velte, Robert Elsenpeter, “Cloud Computing: A Practical Approach”, Tata McGraw Hill Edition, Fourth Reprint, 2010
- Barrie Sosinsky (2011) Cloud Computing Bible, Wiley, India
- Bloor R., Kanfman M., Halper F. Judith Hurwitz “Cloud Computing for Dummies”, Wiley India Edition, 2010
- John Rittinghouse& James Ransome, “Cloud Computing Implementation Management and Strategy”, CRC Press, 2010.
- Michael Miller, “Cloud Computing: “Web-Based Applications That Change the Way You Work and Collaborate Online”, Que Publishing, August 2008.



Course Code: CA5426	Course Title: UI/UX Design
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs.

Course Objectives:

1. Understand the fundamentals of UI/UX design principles and methodologies.
2. Learn user research techniques and their applications in creating user-centric designs.
3. Gain proficiency in wireframing, prototyping, and usability testing.
4. Explore visual design principles, typography, and color theory in UI design.
5. Develop interactive design solutions using modern tools and technologies.
6. Analyze and evaluate design solutions for usability and accessibility.

Course outcomes:

On completion of the course the student will be able to:

- Explain the core principles of UI/UX design.
- Conduct user research and apply findings to design decisions.
- Create wireframes, prototypes, and interactive design models.
- Use design tools like Figma, Adobe XD, or Sketch effectively.
- Implement accessible and inclusive design principles.
- Critically assess design solutions for user engagement and functionality.

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	*				*		*	*
CO2	*	*	*	*				*
CO3	*	*	*	*	*	*		*
CO4		*	*	*	*	*	*	
CO5	*		*	*	*	*	*	*



CONTENT	HOURS
UNIT 1: Introduction to UI/UX Design	
Introduction: UI vs. UX: Definitions and Roles. Importance of UI/UX in product development. Design Thinking Process: Empathize, Define, Ideate, Prototype, Test. UI Design Principles: Consistency, Feedback, Visual Hierarchy, Affordance. Introduction to Tools: Adobe XD, Figma, Canva – Overview and Features.	8
UNIT 2: User Research and Analysis	
Importance of User Research in UX. Methods: Surveys, Interviews, Focus Groups, and Observations. Analyzing Data: Personas, Empathy Maps, and Journey Maps. Identifying Pain Points and Insights for Design.	8
UNIT 3: Wireframing and Prototyping	
Wireframing: Low-Fidelity vs. High-Fidelity. Creating Wireframes with Figma and Adobe XD. Interactive Prototypes: Adding Transitions and Animations. Iteration: Collecting Feedback and Refining Prototypes.	10
UNIT 4: Usability and Accessibility	
Usability Testing: Methods, Tools, and Guidelines. Accessibility in Design: WCAG Guidelines and Standards. Evaluating Designs Based on User Feedback. Case Studies: Examples of Good and Poor UI/UX Design.	8
UNIT 5: Advanced UI/UX Topics	
Responsive Design: Mobile-First and Adaptive Layouts. Interaction and Motion Design Basics. Ethical Design Practices: Inclusive and Sustainable Design. Future of UI/UX: AI in Design, Personalization, and Emerging Trends.	8

**Reference Books**

1. Steve Krug, *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*, 3rd Edition, New Riders.
2. Aaron Walter, *Designing for Emotion, A Book Apart*.
3. Don Norman, *The Design of Everyday Things*.
4. Jenifer Tidwell, *Designing Interfaces: Patterns for Effective Interaction Design*.
5. Rex Hartson and PardhaPyla, *The UX Book: Agile UX Design for a Quality User Experience*.

References:

- Maze, *UX/UI Design Tools Collection* - <https://maze.co/collections/ux-ui-design/tools/>
- Skillshare, *The Ultimate Figma Prototyping & Animation Course* - <https://www.skillshare.com/en/classes/the-ultimate-figma-prototyping-and-animation-course/2004892826>
- UX Mastery - <https://uxmastery.com/>
- Smashing Magazine – UX Design - <https://www.smashingmagazine.com/category/uxdesign/>
- UX Collective - <https://uxdesign.cc/>

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	8	12
Unit 2	10	12
Unit 3	8	22
Unit 4	8	22
Unit 5	8	10
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) =60		



Course Code:CA5526	Course Title: Blockchain Technology and Applications
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2Hrs

Course Objectives:

1. To understand the core concepts and architecture of Blockchain technology.
2. To explore consensus mechanisms and distributed ledger systems.
3. To provide insight into smart contracts and decentralized applications (DApps).
4. To analyze real-world blockchain use-cases and security aspects.

Course Outcomes:

At the end of the course, students will be able to:

1. Explain the fundamentals and architecture of Blockchain technology.
2. Describe consensus algorithms and distributed ledger systems.
3. Develop smart contracts and decentralized applications.
4. Analyze the application of blockchain in real-world scenarios.
5. Understand blockchain security and privacy concerns.
6. Explore recent trends and future scope in Blockchain.

CONTENT	Hours
UNIT-1	
INTRODUCTION TO BLOCKCHAIN TECHNOLOGY: Definition and characteristics of Blockchain, History and Evolution, Types of Blockchain – Public, Private, Consortium, Blockchain Architecture, Blocks and Hashes, Merkle Tree, Distributed Ledger, Applications and Limitations of Blockchain. Self-study: Use cases of Blockchain in various domains (Supply Chain, Finance, Healthcare, etc).	7



UNIT-2	
CRYPTOGRAPHY AND CONSENSUS MECHANISMS: Cryptographic principles – Hash functions, Digital Signatures, Public Key Infrastructure (PKI); Consensus Mechanisms – Proof of Work, Proof of Stake, Delegated Proof of Stake, Practical Byzantine Fault Tolerance (PBFT). Self-study: Elliptic Curve Cryptography (ECC) basics	8
UNIT-3	
SMART CONTRACTS AND ETHEREUM PLATFORM: Ethereum Overview, Ether, Gas and Transactions, Ethereum Virtual Machine (EVM), Writing Smart Contracts using Solidity, Deploying Smart Contracts, Decentralized Applications (DApps). Self-study: Tools for Smart Contract Development – Remix IDE, Truffle	10
UNIT-4	
BLOCKCHAIN PLATFORMS AND FRAMEWORKS: Hyperledger Fabric Overview, Architecture, Chain code, Consensus in Fabric; Other platforms – Corda, Quorum, Multichain; Blockchain-as-a-Service (BaaS). Self-study: Hyperledger Composer	7
UNIT-5	
SECURITY, PRIVACY, AND FUTURE TRENDS: Blockchain security issues – 51% attack, Sybil attack, Double-spending, Privacy-enhancing techniques – Zero Knowledge Proofs, Ring Signatures; Current Trends – NFTs, DeFi, DAOs; Blockchain integration with IoT, AI and Big Data. Self-study: Regulation and legal aspects of Blockchain	10



Text Books:

1. Imran Bashir, *Mastering Blockchain*, 3rd Edition, Packt Publishing, 2020.

References:

1. Andreas M. Antonopoulos, *Mastering Bitcoin*, O'Reilly Media, 2017.
2. Arvind Narayanan et al., *Bitcoin and Cryptocurrency Technologies*, Princeton University Press, 2016.
3. Joseph J. Bonneau, *SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies*, IEEE Symposium, 2015.
4. Roger Wattenhofer, *The Science of the Blockchain*, 2016
5. Hyperledger Fabric Documentation - <https://hyperledger-fabric.readthedocs.io>

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	7	15
Unit 2	8	18
Unit 3	10	15
Unit 4	7	15
Unit 5	10	15
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) = 60		



Course Code: CA5P226	Course Title: UI/UX Lab
Course Credits: 01	Hours/Week: 02
Total Contact Hours: 22	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hrs.

Course outcomes:

On completion of the course the student will be able to:

- Identify differences and relationships between UI and UX, and articulate their respective roles in product development.
- Apply user research methods and synthesize findings into actionable design insights.
- Develop low-fidelity and high-fidelity wireframes for web or mobile interfaces using industry-standard tools.
- Create interactive prototypes to demonstrate user flows, transitions, and design solutions.
- Conduct usability testing, evaluate interface accessibility, and propose improvements based on user feedback.
- Explain the importance of ethical, inclusive, and future-oriented UI/UX practices.
- Demonstrate design iteration based on critique and user/testing outcomes.

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	*				*		*	*
CO2	*	*	*	*				*
CO3	*	*	*	*	*	*		*
CO4		*	*	*	*	*	*	
CO5	*		*	*	*	*	*	*



Program List Part A: UI/UX Practicals		
1.	Using Figma, Adobe XD or Sketch, design a modern login screen for a mobile app (logo, fields, call-to-action, clear alignment).	
2.	Design a responsive top navigation bar for a web dashboard (with icons and dropdown menu) in Sketch, Figma, or Marvel.	
3.	Using your chosen tool, create a visually appealing product card for an e-commerce app. Include image, title, price, and an “Add to Cart” button.	
4.	Create a wireframe (low fidelity) for a mobile app home page with at least four sections (e.g., header, featured content, navigation, footer).	
5.	In Figma, Adobe XD, or InVision, design a modal popup for email subscription and create a clickable prototype that opens and closes the modal.	
6.	Design three sizes of primary CTA buttons (default, hover, disabled) for a web application in Sketch or Figma, using standard styles and color palettes.	
7.	Design a set of three custom icons (e.g., home, user, settings) using vector tools in Sketch or Adobe XD.	
8.	Use Zeplin or Abstract to export a screen you designed, generate assets for development, and annotate key measurements and color codes.	
Part B: Advanced and Interactive UI/UX Tasks		
1.	Take a desktop landing page and adapt it for mobile and tablet viewports in Sketch, Figma, or Adobe XD. Adjust layout, spacing, and touch targets.	
2.	Use Figma’s Smart Animate, Framer, or Adobe XD to add a hover animation to a button or a loading spinner to a form.	
3.	Work in pairs using Abstract, Figma, or Sketch’s collaboration tools to simultaneously edit and merge changes to the same project (document the merge results with screenshots).	
4.	Using Figma or Marvel, prototype a three-tab navigation for a mobile app. Each tab should reveal a different content area interactively	
5.	In Framer, Marvel, or InVision, prototype a three-step onboarding	



	process showing transitions between screens with animated effects.	
6.	Create an animated feedback “toast” notification (e.g., “Saved successfully!”) using Sketch, Figma, or Framer. Show appearance, auto-fade, and styling.	
7.	Place your finished mobile screens into a realistic device mockup using Sketch, Figma, or Adobe XD, to present for portfolio or demonstration.	
8.	Upload a completed project to Zeplin or Abstract. Add notes on layout, color, and interaction for developer reference.	

Reference:

1. "Fundamentals of user-centered design: A practical approach" (CRC Press, 2017).
2. Design system documentation: Material Design Guidelines, iOS Human Interface Guidelines.
3. Leading industry blogs: UX Mastery, Smashing Magazine, UX Collective, InVision Blog.

Web Reference:

Leading industry tools, feature overviews, and best-use advice

<https://maze.co/collections/ux-ui-design/tools/>

Lessons on prototyping, animating, and presenting designs in Figma.

<https://www.skillshare.com/en/classes/the-ultimate-figma-prototyping-and-animation-course/2004892826>



Course Code: CA 5P226	Course Title: Machine Learning Lab
Course Credits: 01	Hours/Week: 02
Total Contact Hours: 22	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hrs

Course Objectives:

1. To equip students with the knowledge about python libraries.
2. To equip students with the knowledge about machine learning algorithms.
2. To provide experience in applying machine learning algorithms to practical problems.

Course Outcomes:

1. Understand the library functions in python for data preprocessing
2. Use the python machine learning models .
3. Understand complexity of Machine Learning algorithms and their limitations
4. Apply appropriate algorithms for problem solving.
5. Capable of performing experiments in Machine Learning using real-world data

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	*							
CO2	*							
CO3	*							
CO4	*	*	*					
CO5	*	*	*	*		*		



Program List												
	Python Libraries: Numpy, Pandas, Matplotlib and Scikit											
1.	Creating Arrays and Array Operations: a)Create a 1D array with elements from 0 to 9. b)Create a 2D array with shape (3, 4) filled with random numbers. c)Calculate the mean and standard deviation of a given array. d)Normalize the values of an array (subtract the mean and divide by the standard deviation).											
2	Generating Array, Reshaping and Stacking: a) Create a 1D array of 10 evenly spaced values between 0 and 1. b) Generate a 3x3 identity matrix. c) Reshape a 1D array into a 2D array with shape (2, 5). d) Stack two arrays vertically and horizontally.											
3	Indexing and Slicing & Matrix Operations: a) Extract the third column from a 2D array. b) Reverse the order of elements in a 1D array. c) Create two matrices (2x3 and 3x4) and perform matrix multiplication. d) Find the determinant of a 3x3 matrix.											
4	Statistical Operations, Broadcasting: a) Generate a random 2D array and calculate the mean along each axis. b) Find the minimum and maximum values in a given array. c) Create a 1D array and add a constant value to each element without using a loop. d) Multiply each row of a 2D array by a different constant.											
5	Develop a python program to create pandas data frame from list of data.											
6	Develop a python program to analyze the dataset using pandas and matplotlib library											
7	Develop a program to compute Mean, Median, Mode, Variance and Standard Deviation using Datasets.											
8	Plot histogram for the given dataset. <table><tr><td>10-15</td><td>15-20</td><td>20-25</td><td>25-30</td><td>30-35</td></tr><tr><td>5</td><td>6</td><td>9</td><td>8</td><td>2</td></tr></table>	10-15	15-20	20-25	25-30	30-35	5	6	9	8	2	
10-15	15-20	20-25	25-30	30-35								
5	6	9	8	2								



9	Draw box-and-whisker plot for the data set {3, 7, 8, 5, 12, 14, 21, 13, 18}.																	
10	Draw Line Plot and Bar chart for the following data. <table><tr><td>Elapsed time (s)</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>Speed(m/s)</td><td>0</td><td>3</td><td>7</td><td>12</td><td>20</td><td>30</td><td>45.6</td></tr></table>	Elapsed time (s)	0	1	2	3	4	5	6	Speed(m/s)	0	3	7	12	20	30	45.6	
Elapsed time (s)	0	1	2	3	4	5	6											
Speed(m/s)	0	3	7	12	20	30	45.6											
11	Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file																	
12	Develop a python program to implement Simple linear regression and plot the graph																	
13	Develop a python program to implement single layer perceptron.																	
14	Implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.																	
15	Demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.																	
16	Implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions.																	
17	Apply k means algorithm to cluster a set of data stored in a .CSV file.																	
18	Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.																	



Course Code: CA5S26	Course Title: AWS CLOUD PRACTITIONER ESSENTIALS
Course Credits: 02	Hours/Week: 02
Total Contact Hours: 44	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hrs

- Use the AWS Academy Learner Lab (Sandbox Environment) to complete the lab activities.
- Document each step with brief explanations and relevant screenshots.

1. Using the AWS Management Console, perform the following tasks and answer the questions below:

- Identify the Region and Availability Zone:
 - What is the default region selected when you log in to the AWS Console?
 - List at least one Availability Zone within this region.
- Service Identification:

Explore the AWS Console and identify **any 3 services** under each of the following categories. For each service, write a one-line description of its primary purpose.

 - Compute Services
 - Storage Services
 - Database Services
 - Security Services
 - Networking Services

2. Launch one Windows EC2 instance and one Linux EC2 instance using the AWS Academy Learner Lab environment. Connect to both instances and verify successful access.

3. Launch a web server on an EC2 instance and host a static website without using Amazon S3.

4. Design and implement a solution to upload, store, retrieve, and delete user documents and images using AWS Cloud Storage.

5. Perform the following tasks using AWS EC2 and EBS services:

- Launch a Windows EC2 instance.
- Create an additional EBS volume in the same Availability Zone as the instance.
- Attach the EBS volume to the EC2 instance.
- Log in to the Windows instance using Remote Desktop (RDP), initialize the new volume, create a new partition, and format it.



- v. Assign a drive letter and verify that the new volume is accessible.
 - vi. Finally, detach and delete the EBS volume from the instance via the AWS Management Console.
- 6. Create, configure, and manage an Amazon RDS instance. After successful setup, create a manual snapshot of the RDS instance and demonstrate how to restore a new database instance from the snapshot.
- 7. Set up a virtual machine (VM) using any virtualization platform (e.g., AWS EC2). Install a suitable compiler for a programming language of your choice (e.g., GCC for C/C++, Python interpreter, Java JDK). Then, write and execute a sample program using that compiler. In your response, include the following:
 - i. Name and configuration of the virtual machine
 - ii. Steps to install the compiler
 - iii. Code of the sample program
 - iv. Compilation and execution process
 - v. Screenshot or output of the executed program.

PART-B

AWS Lab Mini Project - Cloud Web Application Builder

In this project, you're challenged to use familiar AWS services to build a solution.

By the end of this project, you should be able to do the following:

- i. Create an architectural diagram to depict various AWS services and their interactions with each other.
- ii. Estimate the cost of using services by using the AWS Pricing Calculator.
- iii. Deploy a functional web application that runs on a single virtual machine and is backed by a relational database.
- iv. Architect a web application to separate layers of the application, such as the web server and database.
- v. Create a virtual network that is configured appropriately to host a web application that is publicly accessible and secure.
- vi. Deploy a web application with the load distributed across multiple web servers.
- vii. Configure the appropriate network security settings for the web servers and database.
- viii. Implement high availability and scalability in the deployed solution.
- ix. Configure access permissions between AWS services.



Course Code: CARM5626	Course Title: RESEARCH METHODOLOGY AND PUBLICATION ETHICS
Course Credits: 02	Hours/Week: 03
Total Contact Hours: 34	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives

1. To familiarize students with fundamental research concepts and various methodologies used in academic and industrial research.
2. To develop skills in designing and conducting systematic research, including data collection, analysis, and interpretation.
3. To equip students with knowledge and tools for ethical and effective publication in peer-reviewed journals and conferences.

Course Outcomes (COs)

1. Understand and apply the fundamental principles of research design, methodology, and data collection techniques in scientific inquiry.
2. Critically evaluate different research methods and select appropriate tools and techniques for conducting ethical and impactful research.
3. Demonstrate the ability to identify research problems, formulate hypotheses, and structure a research proposal with clear objectives and methodology.
4. Recognize and adhere to ethical standards in academic research and publications, including issues of plagiarism, authorship, and peer review processes.
5. Develop effective academic writing and communication skills for the preparation of research papers and publications in peer-reviewed journals.

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	*							
CO2	*	*	*					
CO3	*	*	*					
CO4	*	*	*					
CO5	*	*	*	*		*		



CONTENT	Hours
UNIT-1	
Meaning of Research, Characteristics of Scientific Research, etc. Science and Common Sense, Introduction to Research, Scientific Research, Steps in Scientific Research, Methods of Scientific Research, Criteria of Good Research Types of Research Normative Research, Scientific Research, and Value-Free Research, Social Science Research Vs. Scientific Methods Significance of Research in Social Sciences, Scientific Methods, Defining the Problem of Research, Concepts and Constructs, Meaning and Function of Research Design	6
UNIT-2	

Research Design Important Concepts Relating to Research Design, Sampling Sampling and Its Purposes, Steps in Sample Design, Principles of Sampling, Sample Size Determination, Non-Random Sampling, Random Sampling, Complex Random Sampling, Measurement and Scaling Techniques	5
UNIT-3	
Hypotheses and Synopsis Writing Hypotheses, Difference Between a Proposition, a Hypothesis and a Theory, Basic Concepts Concerning Testing of Hypothesis, Data Preparation – Univariate analysis (frequency tables, bar charts, pie charts, percentages), Bivariate analysis – Cross tabulations and Chi-square test including testing hypothesis of association. Python and AI tools as a statistical tool.	5
UNIT-4	



Ethics: Moral philosophy, nature of moral judgements and reactions. Scientific Conduct: Ethics with respect to science and research, Intellectual honesty and research integrity. Scientific misconducts: Falsification, Fabrication, and Plagiarism (FFP). Redundant publications: duplicate and overlapping publications, salami slicing, selective reporting and misrepresentation of data. Publication Ethics Publication ethics: definition, introduction and importance. Best practices / standards setting initiatives and guidelines: COPE, WAME, etc., Self Study: Publication misconduct: Concept, problems that lead to unethical behaviour and vice versa, types. Violation of publication ethics, authorship and contributorship. Identification of publication misconduct, complaints and appeals, Predatory publishers and journals.	8
--	---

UNIT-5	
Open Access Publishing Open access publications and initiatives, SHERPN/RoMEO online resource to check publisher copyright & self-archiving policies Software tool to identify predatory publications developed by SPPU, Journal finder / journal suggestion tools viz. JANE, Elsevier Journal Finder, Springer Journal Suggester, etc. Publication Misconduct : Subject specific ethical issues, FFP, authorship, Conflicts of interest. Complaints and appeals: examples and fraud from India and abroad. Software tools: Use of plagiarism software like Turnitin, Urkund and other open source software tools.	6

Books and references

- Kothari, C. R. (2004). Research Methodology: Methods and Techniques. India: New Age International (P) Limited.
- Ahuja, R. (2001). Research Methods. India: Rawat Publications.
- Santhosh Kumar Yadav · research ethics, misconduct and the ensuing actions as per international law.



Course Code: CA6126	Course Title: Internet of Things
Course Credits: 3	Hours/Week: 3
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 02Hrs

Course Objectives:

The main objectives of this course are to:

1. About the Internet of Things where various communicating entities are controlled and managed for decision making in the application domain.
2. Enable students to learn the Architecture of IoT and IoT Technologies.
3. Developing IoT applications and Security in IoT, Basic Electronics for IoT, Arduino IDE, Sensors and Actuators Programming NODEMCU using Arduino IDE

Course Outcomes:

- Define the fundamental concepts of IoT, including its architecture, protocols, and applications.
- Explain the working principles of IoT devices, sensors, and actuators, and their role in IoT systems
- Demonstrate the use of IoT hardware platforms (e.g., Arduino, Raspberry Pi) to develop basic applications
- Analyze different communication protocols and networking models used in IoT systems

SYLLABUS

UNIT-1	
Introduction Introduction to IoT: Evolution – Definition & Characteristics – Architecture-Technologies– Developing IoT Applications – Applications – Industrial IoT – Security in IoT.	

7 Hours



UNIT-2	
Basic Electronics for IoT Electric Charge, Resistance, Current and Voltage – Binary Calculations – Logic Chips – Microcontrollers – Multipurpose Computers – Electronic Signals – A/D and D/A Conversion – Pulse Width Modulation.	8 Hours
UNIT-3	
Programming using Arduino Programming Fundamentals with C using Arduino IDE: Installing and Setting up the Arduino IDE – Basic Syntax – Data Types/ Variables/ Constant – Operators – Conditional Statements and Loops – Using Arduino C Library Functions for Serial, delay and other invoking Functions – Strings and Mathematics Library Functions. IoT Application Development using TINKERCAD ● Blinking LED: Create a basic Arduino circuit to blink an LED. ● Breadboard LED Setup: Light up two LEDs using a breadboard and resistor ● Button Control: Turn an LED on/off using a push button and Arduino code. Smart Light System: Use an LDR (light sensor) to automatically turn on an LED in the dark. ● Temperature Alert System: Read temperature using TMP36 sensor and turn on a fan or LED when it's too hot. Motion Detection Alarm: Build a motion sensor alarm using a PIR sensor and buzzer. Sound & Display: Display sensor data (like temperature or light level) on the Serial Monitor and sound an alert using a buzzer. ● Traffic Light Simulation: Program a sequence of red, yellow, and green lights using 3 LEDs. ● RGB LED Color Mixer: Control an RGB LED to display different colors using code.	13 Hours
UNIT-4	
Sensors and Actuators	



Sensors and Actuators: Analog and Digital Sensors–Interfacing temperature sensor, ultrasound Sensor and infrared (IR) sensor with Arduino – Interfacing LED and Buzzer with Arduino.	7 Hours
UNIT-5	7 Hours
IoT Applications IoT Applications for Value Creations Introduction, IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications, Four Aspects in your Business to Master IoT, Value Creation from Big Data and Serialization, IoT for Retailing Industry, IoT for Oil and Gas Industry, Opinions on IoT Application and Value for Industry, Home Management, eHealth.	
Case Study: Working with Wokwi Simulator.	

Text Books

1. ArshdeepBahga, Vijay Madiseti, “Internet of Things : A Hands-On Approach”, 2014. ISBN: 978-0996025515
2. Boris Adryan, Dominik Obermaier, Paul Fremantle, “The Technical Foundations of IoT”, Artech Houser Publishers, 2017.

Reference Books

1. Michael Margolis, “Arduino Cookbook”, O’Reilly, 2011
2. Marco Schwartz, “Internet of Things with ESP8266”, Packt Publishing, 2016.
3. DhivyaBala, “ESP8266 : Step by Step Tutorial for ESP8266 IoT, Arduino NODEMCU Dev. Kit”, 2018.

Chapter	Number of Hours	Total marks for which question as to be asked
Unit 1	7	8
Unit 2	8	8
Unit 3	13	10
Unit 4	7	20
Unit 5	7	20
	42	66



Course Code: CA6226	Course Title: Cyber Law & Cyber Security
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives:

1. To provide an understanding of the main issues related to security in modern networked computer systems.
2. To understand the concepts and foundations of computer security.
3. Basic knowledge about security-relevant decisions in designing IT infrastructures from personal laptops to large-scale infrastructures.

Course Outcomes:

Here the students will be able to understand

1. An extensive, detailed, and critical understanding of the concepts, issues, principles and theories of computer network security.
2. Theoretical and detailed practical knowledge of a range of computers.
3. Network security technologies, as well as network security tools and services.
4. Gain experience of analyzing, designing, implementing and validating solutions to computer network security challenges using common network security tools and formal methods.
5. Demonstrate knowledge of reasoning and knowledge representation for solving cybercrime problems.

CONTENT	Hours
UNIT-1	
Cyber Laws Internet Governance – Challenges and Constraints, Cyber Threats: - Cyber Warfare- Cyber Crime-Cyber Terrorism-Cyber Espionage, State and Private Sector in Cyberspace, Cyber Security Standards. The INDIAN Cyberspace, The Indian Penal code, National Cyber Security Policy 2017 & 2023	7



UNIT-2	
Cyber Security Introduction to Cyber Security. Confidentiality, Integrity and Availability – Triad. Attacks: Threats, Risk Assessment and Analysis, Hackers and types, Information Classification, Policies, Procedure and Guidelines, Vulnerabilities and Risk, Layers of cyber security. Self-study: Best cyber security practices.	8
UNIT-3	
Basics of Cryptography Symmetric and Asymmetric Cryptosystems, Classical Encryption Techniques – Substitution Techniques, Transposition Techniques. Feistel Structure, Data Encryption Standard (DES), Principles of public key cryptosystems-The RSA Algorithm-Key management - Diffie Hellman Key exchange, Hashing, MD5 and SHA-1 Algorithms. Self-study: SHA 512 Algorithm	10
UNIT-4	
Network and Wireless Attacks Network Sniffing, packet analysis, display and capture filters, Ettercap, DNS Poisoning, ARP Poisoning, Denial of services, Vulnerability scanning, Setup network, IDS/IPS, Router attacks, Man-in-the-middle Attack, MAC Filtering, Packet Encryption, Packet Sniffing, Types of authentication, Attacks on WEP, WPA Encryption. Case study: Fake hotspots.	7
UNIT-5	
Network Security IP security architecture, Security protocols, IP Sec, Firewalls, IDPS – Types and technologies. Trusted systems – Electronic payment protocols, SET Network Security Applications, Authentication, Mechanisms: Passwords, Cryptographic authentication protocol, Kerberos, X.509 certificate, Digital Signatures. Self-study: Transport layer secure.(SSL)	10



Text Books:

1. SunitBelapure and Nina Godbole, “Cyber Security : Understanding Cyber-crimes and Legal perspectives”, Wiley India Pvt Ltd,2020.
2. Atul Kahate, “Cryptography and Network Security,” Tata McGraw-Hills (Reprinted 2021).

References:

1. William Stallings; “Cryptography and Network Security: Principles and Practices”, Fifth Edition, Prentice Hall Publication Inc., 2007.
2. Nina Godbole and SunitBelapure; “Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives”, Wiley Publications, 2011.
3. Matt Bishop, “Computer Security Art and Science”, Pearson/PHI, 2002.
4. Michael E Whiteman and Herbert J Mattord; “Principles of Information Security”, Vikas Publishing House, New Delhi, 2003.
5. Alfred J. Menezes, Paul. C. Van Oorschot, and Scott A. Vanstone “Handbook of Applied Cryptography”, CRC press, Lib of Congress -2006

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	7	10
Unit 2	8	10
Unit 3	10	22
Unit 4	7	22
Unit 5	10	14
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) =60		



Course Code: CA6326	Course Title: Mobile Application development
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives:

1. To introduce the fundamentals of Android development and Kotlin programming.
2. To familiarize students with Android Studio and user interface design.
3. To develop Android applications using core components and layouts.
4. To explore data storage techniques and advanced Android features.

Course Outcomes (COs):

By the end of the course, students will be able to:

1. Explain the core concepts of mobile application development and the advantages of using Kotlin for Android.
2. Design intuitive and responsive user interfaces using Android layouts, views, and widgets in Kotlin.
3. Develop mobile applications that manage user interaction, navigation, and lifecycle using Kotlin.
4. Implement secure and efficient data storage solutions using Shared Preferences, internal storage, and SQLite.
5. Integrate media features, location services, and external APIs to enhance mobile application functionality.

CONTENT	Hours
UNIT – 1 INTRODUCTION TO ANDROID & KOTLIN PROGRAMMING Overview of Android and its evolution, Applications of Android, Android OS and versions, Features and Architecture of Android, History of Android, Installation of Android Studio (Latest version), Why Kotlin for Android?, Comparison: Kotlin vs Java, Android Studio vs Eclipse. Kotlin Basics: Variables, Data Types, Operators, Control Structures and Functions, Null Safety, Collections and Lambdas.	8



UNIT – 2 FUNDAMENTALS OF ANDROID STUDIO AND LAYOUTS Project Structure in Android Studio, Creating the first Android Application, Core Android Components – Activity, Services, Broadcast Receivers, Content Providers. Intent and Types of Intent, Android Services and Types of Services, View and ViewGroups, Types of Views, Android Widgets – Information, Collection, Control, Hybrid. Layouts – Constraint Layout, Linear Layout, Relative Layout, Frame Layout, Introduction to Animations.	8
UNIT – 3 ACTIVITIES AND UI ELEMENTS Activity Lifecycle, Introduction to Themes, Styles and Colours, Menus in Android: Options, Context, Popup. Introduction to Fragments, Fragment Lifecycle and Communication.	8
UNIT – 4 DATA STORAGE IN ANDROID Introduction to Shared Preferences, Operating Modes, Editor, Put, Read, Clear, Delete methods. Internal Storage in Android, Reading and Writing Files in Internal Storage, Comparison: Shared Preferences vs Internal Storage.	8
UNIT – 5 ADVANCED ANDROID FEATURES AND MEDIA FUNCTIONS Controlling Audio and Video Playback, UI Components: Action Bar, Alert Dialog, List View, Grid View, Timer. Map Activity using Google Maps API. SQLite Database: Inserting, Updating, Deleting and Retrieving Data.	10

Textbooks

1. Android Programming with Beginners
2. Kotlin for Android Developers
3. Android Programming: The Big Ranch Guide



Course Code: CA6426	Course Title: Data Analytics
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives

This course aims to equip students with a comprehensive understanding of advanced data analytics, covering data preprocessing, exploratory analysis, predictive modeling, and visualization techniques. It emphasizes the use of modern tools and frameworks aligned with current industry practices. Students will develop the ability to derive insights from structured and unstructured datasets using statistical and machine learning approaches.

Course Outcomes (COs):

CO1 : Describe the principles, types, and applications of data analytics in various industries.

CO2 : Analyze and preprocess data using appropriate techniques and tools

CO3 : Build and Evaluate predictive models using machine learning algorithms

CO4 : Visualize and interpret data to derive actionable business insights

CO5 : Design and execute a complete data analytics project from problem definition to deployment.

Content	Hours
Unit – 1	
Foundations of Data Analytics Introduction to Data Analytics and the Data Science Lifecycle Types of Data: Structured, Semi-Structured, and Unstructured, Data Types and Measurement Scales, Types of Analytics: Descriptive, Diagnostic, Predictive, Prescriptive Introduction to Python for Data Analytics Hands-on: Jupyter Notebook, Data Import & Basic Exploration Tools: Python (Pandas, NumPy), Jupyter, Excel	8
Unit-2	
Data Preparation and Exploratory Data Analysis (EDA) • Data Cleaning: Handling Missing Data, Duplicates, Outliers • Data Transformation: Normalization, Encoding, Binning	8



● Feature Engineering & Feature Selection Techniques ● Exploratory Data Analysis: Univariate & Bivariate Analysis\ ● Data Visualization & Interpretation ● Hands-on: Charts, Heatmaps, Histograms, Boxplots Tools: Python (Matplotlib, Seaborn, Plotly), Power BI/Tableau (Intro)	
Unit-3	8
Predictive Analytics and Machine Learning ● Introduction to Machine Learning Models ● Regression Techniques: Linear, Logistic ● Classification Models: Decision Tree, K-Nearest Neighbors, Random Forest ● Model Evaluation Metrics: Accuracy, Precision, Recall, F1, ROC-AUC ● Cross Validation, Overfitting & Regularization ● Hands-on: Model Training, Testing, Evaluation Tools: Scikit-learn, TensorFlow/Keras (Intro), Google Colab	
Unit-4	10
Big Data Analytics ● Introduction to Big Data Concepts ● Hadoop Ecosystem vs. Apache Spark ● Real-time vs. Batch Processing ● PySpark: DataFrames, Transformations, Actions ● Introduction to NoSQL Databases (MongoDB overview) ● Hands-on: Spark (local setup or Databricks/Google Colab integration) Tools: PySpark, MongoDB (Intro)	
UNIT 5	8
Industry Applications, Optimization & Capstone Project ● Industry Use Cases: Fraud Detection, Market Basket Analysis, Customer Segmentation ● Optimization Techniques in Analytics (Linear Programming, Decision Modeling) ● Prescriptive Analytics & Simulation Basics ● Ethical Use of Data & AI Regulations (GDPR, Bias in AI) ● Capstone Project: End-to-End Problem Solving Tools: Full Stack Workflow (Python + Visualization Tool + Report Writing)	



- Grus, J. (2022). *Data Science from Scratch: First Principles with Python* (2nd ed.)

Publisher: O'Reilly Media

- Géron, A. (2023). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed), Publisher: O'Reilly



Course Credits: 02	Hours/Week: 04
Total Contact Hours: 44	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives:

1. Identify real world problems and challenges that need IT based solutions.
2. Demonstrate an ability to work in teams.
3. Improve the team building, communication and management skills of the students.
4. Use all concepts of IT in creating a solution for a problem.
5. Report and present the findings of the study conducted in the preferred domain.

Course Outcomes:

1. Problem definition
2. Analyze Systems requirements.
3. Perform and evaluate feasibility studies like cost-benefit analysis, technical feasibility, time feasibility and operational feasibility for the project.
4. Construct and evaluate data flow diagrams.
5. Generate various reports in project.

Project Guidelines**1) Understanding the significance of Project**

Most of the students are under an impression that if good layout is proposed then it will help them in scoring good marks but the quality of project is analysed by proper write-ups. It provides an opportunity for learners to demonstrate originality and to plan and organize the project work and put the practical approach of all the topics studied in the entire curriculum.

2) Meaning of Project

A project is a study of factual information for comprehending and applying the various concepts of the course into practice. Its main purpose is not to generalize but to study the situation with a practical orientation.

GUIDELINES FOR BCA PROJECT

- The project work should normally include software development.
- The students are expected to work on real-life project. However, it is not mandatory for a student to work on a real-life project.
- Not more than four students are permitted to work on a project.
- The project may be done in the university campus/concern study institute or in an approved sponsoring organization (industry/ research and development laboratories / educational institution / Software Company related to the proposed topic.
- The student can formulate a project problem with the help of her/his lab supervisors / teachers and if approved, the students commence working on it.
- A candidate is required to present the progress of the project work during the semester as per the schedule.

4) PROJECT SYNOPSIS FORMAT



The project proposal should be prepared and approved in consultation with supervisor. The project proposal should clearly state the project objectives and the environment of the proposed project to be undertaken. The project proposal should contain complete details in the following form:

1. Title of the project
2. Introduction and objectives of the project
3. Analysis (DFD, ER diagrams, class diagrams, time line etc. As per the project requirements).
4. A complete structure which includes:
 - Name of modules and their description
 - Database / data structures description
 - Process logic of each module (flow chart)
 - Reports generation. (Report format)
5. Tools / platform, hardware and software requirement specifications
6. Organization/ Company Certificate

5) Technical Specifications of Project Report

Length:

The length of the report should be between 80- 100 pages including the cover page, summary, table of contents, list of figures, list of tables, and acknowledgement.

Script and Page Format:

The report should be typed using a Word Processor on standard A4 (210 mm x 297 mm) paper size. A conventional font, size 12-point for Normal text, 14 for headings, 16 for Chapter heading, and line spacing of 1.5 mm should be used.

Margins:

Left-hand margins should have a width of not less than 38 mm to facilitate binding. The right hand, the top, and the bottom should be 25 mm. Each page must be typed in one side, leaving a wide margin.

Paper and Print Quality Paper and print quality:

Paper and Print Quality Paper and print quality are important for successful legibility. The report can be printed on a standard quality paper, (e.g., photocopy paper)

Pagination:

Positioning of page numbers should be at the bottom of the pages. Pages starting from the summary until the last list of tables should be numbered using Latin numbers (I, II, III, IV, ...). Pages starting from the Introduction until the appendices should be numbered using numbers (1,2, 3,...). Pages with figures and tables or illustrations must be also numbered.

Binding:

The report should be hard bound.

Number of copies: Two copies should be submitted – self copy, College copy (Group).

7) Team size: Maximum of four students.



8) Technical Specifications of Project Report

Length: The length of the report should be between 80- 100 pages including the cover page, summary, table of contents, list of figures, list of tables, and acknowledgement.

Script and Page Format: The report should be typed using a Word

Processor on standard A4 (210 mm x 297 mm) paper size. A conventional font, size 12-point and line spacing of 1.5 mm should be used.

Margins: Left-hand margins should have a width of not less than 38 mm to facilitate binding. The right hand, the top, and the bottom should be 25 mm.

Each page must be typed in one side, leaving a wide margin.

Paper and Print Quality Paper and print quality:

Paper and Print Quality

Paper and print quality are important for successful legibility. The report can be printed on a standard quality paper, (e.g., photocopy paper)

Pagination: Positioning of page numbers should be on top right hand side. Pages starting from the summary until the last list of tables should be numbered using Latin numbers (I, II, III, IV...). Pages starting from the Introduction until the appendices should be numbered using numbers (1,2,3,...). Pages with figures and tables or illustrations must be also numbered.

Binding: The report should be hard bound.

Number of copies: Two copies should be submitted – self copy, College copy (Group).

Project Report Format Includes

1. Cover page
2. Certificate
3. Acknowledgment
4. Content
5. Abstract
6. Introduction
7. Study of Existing System
8. Study of Proposed System.
9. System Requirements
10. Analysis
11. Design
12. Coding
13. Testing strategies
14. Future Enhancement
15. Conclusion
16. Bibliography



Course Code: CA6P126	Course Title: Cyber Security Lab
Course Credits: 01	Hours/Week: 02
Total Contact Hours: 22	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hrs

Course Objectives:

The student should be made

1. Exposed to different Cipher techniques
2. Learn to implement the algorithms DES, RSA, MD5, SHA-1
3. Learn to use security tools Cryptool, GnuPG, KF sensor, VI Strumbler.

Course Outcomes:

Here the students will be able to understand

1. An extensive, detailed and critical understanding of the concepts, issues, principles and theories of computer network security.
2. Theoretical and detailed practical knowledge of a range of computer.
3. Network security technologies as well as network security tools and services.
4. Gain experience of analysing, designing, implementing and validating solutions to computer network security challenges using common network security tools and formal methods.
5. Demonstrate knowledge of reasoning and knowledge representation for solving cybercrime problems.



LIST OF EXPERIMENTS

1. Write a program to encrypt and decrypt a Password.
2. Implement the substitution mono alphabetic technique by using the Caesar Cipher algorithm.
3. Write a Program to implement Play fair Cipher algorithm.
4. Write a program to demonstrate polyalphabetic cipher by using Hill Cipher algorithm.
5. Implement Substitution and transposition technique by using Vigenere Cipher algorithm.
6. Write a program to implement Rail fence – row & Column Transformation technique.
7. Write a program to implement Row and Column transformation technique
8. Demonstrate Data Encryption Standard Algorithm using Cryptool.
9. Write a program to implement the RSA Algorithm using HTML and Java script.
10. Write a program to demonstrate Diffie-Hellman's key exchange.
11. Demonstrate Message digest (MD5) Algorithm
12. Implement the SHA-1 hashing Algorithm
13. Demonstrate how to provide secure data storage, secure data transmission and for creating digital signatures (GnuPG)
14. To ensure Security of any one web browser (Mozilla Firefox/Google Chrome).
15. Working with Sniffers for monitoring network communication using Wireshark.
16. Implement KALI LINUX to demonstrate basic command.

Evaluation Scheme for Lab Examination

Assessment Criteria	Marks
Writing	8
Execution	10
Viva Voice	7
Total	25



Course Code: CA6P226	Course Title: Mobile App Lab
Course Credits: 01	Hours/Week: 02
Total Contact Hours: 22	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hrs

1. Create an Android app to display various Android lifecycle phases. Use both Logcat and Toast to display the state of an Activity. Let the Activity contain TextViews and an ImageView.
2. Create a login application to verify username and password. On successful login, redirect to another activity that displays "Welcome User" in a TextView with a Logout button. On clicking Logout, a dialog should appear with OK and Cancel buttons. On clicking OK, navigate back to the login activity; on Cancel, remain on the same activity.
3. Create an Android app with a first activity containing an EditText and a Send button. On clicking the Send button, use an explicit intent to send the text to a second activity, where it should be displayed in a TextView. Include validation to allow only text input.
4. Create a calculator app that performs addition, subtraction, division, and multiplication on numbers. Use a suitable background or image in the activity. Apply validation to ensure the EditText accepts only numerical values.
5. Create a Spinner application with string values taken from the resource directory res/values/strings.xml. On changing the spinner value, the displayed image (from the drawable directory) should change accordingly.
6. Create an app using a RadioButton group to calculate discount on a shopping bill amount. Use EditText to enter the bill amount and select one of three radio buttons to apply a 20%, 25%, or 30% discount. The result should be displayed in a TextView.
7. Use LinearLayout to create a simple application that reads text from a predefined TextView. On button click, convert the text to uppercase, change its color, font family, and size, and display it in an EditText.
8. Create an application using DatePicker and TimePicker to allow the user to select a date and time. Validate the input to ensure only future dates are allowed and display the selected values in a TextView.
9. Create an application that uses TableLayout with TextViews, EditTexts, and Buttons. Also implement a customized styles.xml file in the res/values directory to style the



TextView content.

10. Create an application to perform Create, Insert, Delete, View, and Update operations using SQLite database. The front end should include registration details.
11. Create an app to display three button controls vertically aligned. On selecting a button, the screen background color should change accordingly.
12. Implement three buttons to Start, Pause, and Stop audio and video playback. Use the raw folder to include the media sources.
13. Develop an app that displays the battery level percentage as soon as the app starts. Implement a BroadcastReceiver and set the corresponding permissions in the manifest file.
14. Implement a current location map using external service through Google Maps API. Also include necessary permissions and configurations in the manifest file.



Course Code: CA6S26	Course Title: Power BI Lab
Course Credits: 2	Hours/Week: 02
Total Contact Hours: 30	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hours

Course Objectives

This lab course aims to provide students with hands-on experience in data analysis and visualization using Power BI. Students will learn to import, clean, and transform data, create data models, and build interactive reports. The course emphasizes the use of DAX for calculated columns and measures to derive business insights. Learners will design dashboards using various visual elements like charts, KPIs, and slicers. The course also prepares students to handle real-world datasets and present their findings through a mini project.

Course Outcomes (COs):

CO No.	Course Outcome
CO1	Demonstrate the use of Power BI Desktop interface and its key components.
CO2	Perform data import, cleaning, and transformation using Power Query.
CO3	Design and model data relationships for creating meaningful reports.
CO4	Create calculated columns and measures using DAX functions.
CO5	Develop interactive dashboards using slicers, filters, cards, KPIs, and visuals.
CO6	Interpret and present data-driven insights through reports and a mini project.



PART A: POWER BI LAB EXERCISES

1. Getting Started with Power BI

Main Task: Load a sample Excel or CSV file (e.g., Superstore, Sales). Identify and explain the three main views in Power BI: Report, Data, Model.

Sub-Questions:

1. Load the file into Power BI Desktop and list all table names detected.
2. Switch to **Data View** and note the data type assigned to each column.
3. Verify whether Power BI correctly assigned data types (change if incorrect).
4. Count the total number of rows and columns for each table.
5. Identify which column could serve as a **primary key**.
6. Explain the purpose of **Report View**, **Data View**, and **Model View**.
7. Capture a screenshot showing the loaded dataset in Data View.

2. Data Cleaning with Power Query

Main Task: Use Power Query to remove nulls, duplicates, and split columns. Transform data types as necessary.

Sub-Questions:

1. Identify all columns with **missing values** and decide how to handle them (remove, replace, etc.).
2. Remove duplicate records from the dataset.
3. Split a column into two parts (e.g., City from "City, State").
4. Change incorrect data types (e.g., text to date, number to decimal).
5. Rename columns for clarity and consistency.
6. Document the transformation steps performed in the **Applied Steps** pane.
7. Save and load the cleaned data back into Power BI.



3. Data Modeling

Main Task: Create relationships between multiple tables (e.g., Products, Orders, Customers) and identify relationship type & cardinality.

Sub-Questions:

1. Identify common fields between the tables to create relationships.
2. Create a relationship between Orders and Products tables.
3. Specify the **relationship type** (One-to-Many, Many-to-One, etc.).
4. Specify the **cross-filter direction** and explain why it's chosen.
5. Check whether the relationship is active or inactive.
6. Create a **lookup table** and relate it to your fact table.
7. Test whether the relationship works by creating a simple visual.

4. Visualizations and Charts

Main Task: Create at least three visualizations: bar chart, pie chart, and line chart. Add meaningful titles and legends.

Sub-Questions:

1. Create a **bar chart** to compare sales by category.
2. Create a **pie chart** to show sales distribution by region.
3. Create a **line chart** to display sales trends over time.
4. Add data labels to each visual for clarity.
5. Change chart colors to match a consistent theme.
6. Add descriptive titles and legends to all visuals.
7. Interpret key insights from each visual in 2–3 sentences.



5. Using Filters and Slicers

Main Task: Add slicers for Year, Region, and Category. Apply visual-level and page-level filters.

Sub-Questions:

1. Add a **Year slicer** and test interactivity with visuals.
2. Add a **Region slicer** and observe changes in charts.
3. Add a **Category slicer** to filter product data.
4. Apply a **page-level filter** for a specific region.
5. Apply a **visual-level filter** to show only top 5 products.
6. Compare results before and after applying slicers.
7. Capture screenshots showing slicers in action

6. Calculated Columns and Measures

Main Task: Use DAX to create a calculated column (e.g., Total Sales = Quantity × Price). Create a measure for Total Revenue.

Sub-Questions:

1. Create a calculated column for Total Sales.
2. Create a measure for Total Revenue using SUMX or SUM.
3. Use the measure in a bar chart to compare revenues by category.
4. Explain the **difference between a calculated column and a measure**.
5. Format the measure to show currency values.
6. Create another measure for **Profit Margin**.
7. Add the measure to a KPI visual.



7. Date and Time Functions

Main Task: Extract Year, Month from date columns. Use DATEDIFF to calculate delivery duration.

Sub-Questions:

1. Extract **Year** from Order Date.
2. Extract **Month Name** from Order Date.
3. Calculate Delivery Duration = DATEDIFF(Order Date, Ship Date, DAY).
4. Find the average delivery duration for all orders.
5. Create a table visual to display Order ID, Order Date, Ship Date, and Delivery Duration.
6. Sort the table by Delivery Duration (descending).
7. Identify the order with the longest delivery time.

8. KPI and Card Visuals

Main Task: Use Card, Multi-row Card, and KPI visual to show performance metrics. Add conditional formatting to highlight low values.

Sub-Questions:

1. Create a Card visual showing Total Revenue.
2. Create a Multi-row Card showing Total Sales, Profit, and Orders.
3. Create a KPI visual for Sales Target vs Actual.
4. Add conditional formatting to highlight negative profits.
5. Change KPI colors to green (good) and red (bad).
6. Explain which KPIs are most important for the dataset.
7. Take a screenshot showing KPIs with formatting applied.



9. Interactive Dashboard

Main Task: Combine visuals into one interactive dashboard. Use sync slicers and arrange visuals aesthetically.

Sub-Questions:

1. Arrange visuals in a logical layout for better readability.
2. Add a title and branding to the dashboard.
3. Sync slicers across multiple pages.
4. Use bookmarks for different views of the dashboard.
5. Add tooltip pages for detailed insights.
6. Test interactivity by clicking different slicer values.
7. Capture a screenshot of the final dashboard.

10. Publish to Power BI Service

Main Task: Publish a report to the Power BI workspace and share with at least one user.

Sub-Questions:

1. Save and publish the report to your Power BI workspace.
2. Note the steps followed to publish.
3. Share the report with a specific email ID.
4. Set access permissions (Viewer, Contributor).
5. Test whether the shared user can access the report.
6. Explain security considerations when sharing.
7. Capture a screenshot of sharing settings.



PART B: POWER BI MINI PROJECT – BUSINESS DASHBOARD BUILDER

Mini-Project: Build a comprehensive, interactive dashboard for a real-world scenario (sales, finance, HR, healthcare, or education).

Sub-Tasks:

1. Select and import a real-world dataset.
2. Perform data cleaning and transformation.
3. Create a star schema model with relationships.
4. Create calculated columns and DAX measures for KPIs.
5. Build interactive visuals (bar, pie, line, map, cards, KPIs).
6. Add slicers, bookmarks, and drillthroughs.
7. Publish to Power BI Service and share with controlled permissions