ETERNAL UNIVERSITY

(ESTABLISHED UNDER HIMACHAL PRADESH GOVERNMENT ACT NO.3 OF 2009)

BARU SAHIB
HIMACHAL PRADESH

WORLD PEACE THROUGH VALUE BASED EDUCATION

AKAL COLLEGE OF ENGINEERING & TECHNOLOGY

B.C.A. (HONS. WITH RESEARCH) WITH SPECIALISATION IN AI-ML CURRICULUM (SEMESTER III & IV)

APPROVED VIDE ANNEXURE 4.5.3 OF 87TH ACADEMIC COUNCIL MEETING HELD ON 25TH JULY, 2025

TO BE IMPLEMENTED FROM THE ACADEMIC SESSION 2025-26

Dean
Academic Affairs
Eternal University
Baru Sahib (H.P.) 173101

Registrar (Officiating)
Eternal University
Baru Sahib (H.P.) 173101

ETERNAL UNIVERSITY

(Established Under Himachal Pradesh State Act No.3 of 2009)

STUDY SCHEME AND SYLLABUS

FOR

BACHELOR OF COMPUTER APPLICATION (HONS. WITH RESEARCH)

with Specialization in AI-ML

4 YEAR – UG COURSE

2024-2028

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AKAL COLLEGE OF ENGINEERING & TECHNOLOGY

ETERNAL UNIVERSITY

BARU SAHIB, SIRMAUR H.P.

To be implemented from Academic Session 2024-2025

ABOUT THE BCA

Computer Application has been evolving as an important branch of science and technology in the last few years and it has carved out a space for itself like computer science and engineering. Computer application spans theory and more application and it requires thinking both in abstract terms and in concrete terms. Computer science has a wide range of specialties. These include computer architecture, software systems, Graphics, Artificial Intelligence, Mathematical and Statistical Analysis, Data science, Computational Science and Software Engineering.

SALIENT FEATURES OF THE DEPARTMENT

- Research oriented curriculum designed to enable students to acquire all the skills needed to collect and analyze the data.
- The Institute draws upon its strength of highly qualified well-trained faculty, state of art infrastructure and innovative teaching methodology.
- Elective courses that bridge the gap between industry requirements and academia.
- Hands-on experience in most of the courses of computer applications so as to impart practical knowledge in the relevant field.
- To keep the students at par with the emerging technologies prevailing in the market, the institute is furnished with various specialized research labs and software labs.

BCA (BACHELOR OF COMPUTER APPLICATION)

The Program outcomes in BCA are aimed at allowing flexibility and innovation in design and development of course content, in method of imparting training, in teaching learning process and in assessment procedures of the learning outcomes. The emphasis in BCA courses, in the outcome-based curriculum framework, help students learn how to solve problems, accomplish IT tasks, and express creativity, both individually and collaboratively. The proposed framework will help Students learn programming techniques and the syntax of one or more programming languages. After graduating with a 4 years degree, the students are eligible for 1 year MCA (Master in Computer Application) Programme.

VISION

To prepare technically proficient and skillful computer professionals thereby contributing towards building a strong and developed nation.

MISSION

To provide innovative and quality knowledge to students for global competence and excellence. Also, to prepare high quality Professionals for catering the needs of industry.

ELIGIBILITY CRITERIA

10+2 or its equivalent examination in any stream conducted by a recognized Board/ University/Council

DURATION

4 Years

PROGRAM OUTCOMES

PO1: Fundamental Knowledge: Graduates will have a strong foundation in computer science, programming, and information technology, enabling them to understand and apply core principles in diverse areas of computing and technology.

PO2: Problem-Solving Skills: Graduates will be able to analyze complex problems, identify computing requirements, and design and implement effective software solutions using modern tools and techniques.

PO3: Innovation and Research: Graduates will be prepared to engage in innovative practices and research activities, contributing to advancements in the field of computer science and fostering a spirit of inquiry and creativity.

PO4: Ethical and Social Responsibility: Graduates will understand and adhere to professional ethics and responsibilities, recognizing the societal, environmental, and global impacts of their computing solutions.

PO5: Communication and Teamwork: Graduates will develop strong communication skills, both oral and written, and be able to work effectively in diverse teams, exhibiting leadership and collaborative abilities.

PO6: Lifelong Learning: Graduates will recognize the need for continuous learning and self-improvement, staying updated with emerging technologies, methodologies, and best practices in the field of computer science.

PO7: Entrepreneurship and Employability: Graduates will possess the knowledge and skills necessary for successful careers in the IT industry, including the potential for entrepreneurial ventures, by understanding business processes, project management, and the dynamic nature of the technology sector.

PROGRAMME SPECIFIC OUTCOMES (PSO)

PSO1: Apply standard Software Engineering practices and strategies in real-time software project development

PSO2: Design and develop computer programs/computer -based systems in the areas related to AI, algorithms, networking, web design, cloud computing, IoT and data analytics

PSO3: Acquaint with the contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems

PSO4: The ability to apply the knowledge and understanding noted above to the analysis of a given information handling problem.

PSO5: The ability to work independently on a substantial software project and as an effective team member.

SEMESTER-I

S.No.	Course Type	Course Code	Course Title	Hrs./Week			Credits
				L	T	P	
1	DSC-1	0610111011	Foundations of IT	3	0	2	4
2	DSC-2	0610111021	C Programming	3	0	2	4
3	DSC-3	0610111031	Basics of Digital Electronics	3	0	2	4
4	GE-1		One from the pool of GE courses	3	1	0	4
5	SEC-1		One from the pool of SEC courses	1	0	2	2
6	AEC-1		One from the pool of AEC courses	1	0	2	2
7	VAC-1		One from the pool of VAC courses	1	0	2	2
			Total Credits	15	1	12	22

SEMESTER-II

S.No.	Course Type	Course Code	Course Title	Hrs./Week			Credits
				L	T	P	
1	DSC-4	0610121041	Numerical Analysis	3	0	2	4
2	DSC-5	0610121051	Object Oriented Programming with C++	3	0	2	4
3	DSC-6	0610121061	Operating System Principles	3	0	2	4
4	GE-2		One from the pool of GE courses	3	0	2	4
5	SEC-2		One from the pool of SEC courses	1	0	2	2
6	AEC-2		One from the pool of AEC courses	1	0	2	2
7	VAC-2		One from the pool of VAC courses	1	0	2	2
			Total Credits	15	0	14	22

SEMESTER-III

S.No.	Course Type	Course Code	Course Title	Hrs./Week			Credits
				L	T	P	
1	DSC-7	0610131071	Fundamental of DBMS	3	0	2	4
2	DSC-8	0610131081	Data Structures	3	0	2	4
3	DSC-9	0610131091	Fundamentals of Python	3	0	2	4
4	DSE-1/ DSE-2/ DSE-3/ GE-3	0610132011 /061013202 1/06101320 31	Introduction to Web Technologies/ Linux & Shell Scripting/ Java Programming/ GE-3(One From the pool of GE Courses)	3	0	2	4
5	SEC-3		One from the pool of SEC courses	1	0	2	2
6	AEC-3		One from the pool of AEC courses	1	0	2	2
7	VAC-3		One from the pool of VAC courses	1	0	2	2
			Total Credits	15	0	14	22

SEMESTER-IV

S.No.	Course Type	Course Code	Course Title	Hrs./Week			Credits
				L	T	P	
1	DSC-10	0610141101	R Programming	3	0	2	4
2	DSC-11	0610141111	Algorithm Design and Analysis	3	0	2	4
3	DSC-12	0610141120	Introduction to Linear Algebra	3	1	0	4
4	DSE-4/ DSE-5/ DSE-6/ GE-4	0610142041 /061014205 1/06101420 61	Fundamentals of Cyber Security/ Network Programming/ Data Visualization/GE-4 (One From the pool of GE Courses)	3	0	2	4
5	SEC-4		One from the pool of SEC courses	1	0	2	2
6	AEC-4		One from the pool of AEC courses	1	0	2	2
7	VAC-4		One from the pool of VAC courses	1	0	2	2
			Total Credits	15	1	12	22

SEMESTER-V

S.No.	Course Type	Course Code	Course Title	Hrs./Week			Credits
				L	T	P	
1	DSC-13	0610151131	Statistical & Probability for AI	3	0	2	4
2	DSC-14	0610151140	Introduction to AI	3	1	0	4
3	DSC-15	1610151151	Advanced Python	3	0	2	4
4	DSE-7/ DSE-8/ DSE-9	0610152071 /061015208 1/06101520 91	Website Development / Project Based Learning Using Python/Fundamental of Blockchain Technology	3	0	2	4
5	GE-5		One from the pool of GE courses	3	0	2	4
6	SEC-5		One from the pool of SEC courses	1	0	2	2
			Total Credits	16	1	10	22

SEMESTER-VI

S.No.	Course Type	Course Code	Course Title	Hrs./Week			Credits
				L	T	P	
1	DSC-16	0610161161	Introduction to Machine Learning	3	0	2	4
2	DSC-17	0610161171	Fundamentals of Big Data	3	0	2	4
3	DSC-18	0610161180	Cyber Crime and laws	3	1	0	4
4	DSE-10/ DSE-11/ DSE-12	0610162101 /061016211 1/06101621 21	Software Quality Assurance/ Optimization Techniques/ Network Security	3	0	2	4
5	GE-6		One from the pool of GE courses	3	0	2	4
6	SEC-6		One from the pool of SEC courses	1	0	2	2
			Total Credits	16	1	10	22

SEMESTER-VII

S.No.	Course Type	Course Code	Course Title	Hrs./Week			Credits
				L	T	P	
1	DSC-19	0610171191	Advanced Machine Learning	3	0	2	4
2	DSE-13/ DSE-14/ DSE-15	0610172131 /0610172141/0610172151	Advanced R/ Computer Vision/ Quantitative and Statistical Concepts	3	0	2	4
3	DSE-16/ DSE-17/ DSE-18	0610172161 /0610172171/0610172181	Research Methodology/ML Ops/DevOps	3	0	2	4
4	DSE-19/ DSE-20/ DSE-21	0610172191 /0610172201/0610172211	Applied Cryptography/ Introduction to Cloud Computing/ Introduction to Generative AI	3	0	2	4
5	IAPC-1	0610177011	Dissertation on Major Or Dissertation on Minor Or Academic Project/Entrepreneurship	0	0	12	6
			Total Credits	12	0	20	22

SEMESTER-VIII

S.No.	Course Type	Course Code	Course Title	Hrs./Week			Credits
				L	T	P	
1	DSC-20	0610181201	Time Series Analysis and Forecasting	3	0	2	4
2	DSE-22/ DSE-23/ DSE-24	0610182221 /0610182231/0610182241	Pattern Recognition/Deep Learning/Network Simulation	3	0	2	4
3	DSE-25/ DSE-26/ DSE-27	0610182250 /0610182260/0610182270	Entrepreneurship Practices/ Mobile Computing/Distributed Computing Systems	3	1	0	4
4	DSE-28/ DSE-29/ DSE-30	0610182281 /0610182291/0610182301	Natural Language Processing/ Artificial Neural Networks/Soft Computing	3	0	2	4
5	IAPC-2	0610187021	Dissertation on Major Or Dissertation on Minor Or Academic Project/Entrepreneurship	0	0	12	6
			Total Credits	12	1	18	22

OVERALL CREDIT DISTRIBUTION TABLE

SEMESTER	Hrs. / Week			Credits
	L	T	P	
SEMESTER-I	18	1	6	22
SEMESTER-II	18	0	8	22
SEMESTER-III	18	1	6	22
SEMESTER-IV	18	0	8	22
SEMESTER-V	17	1	8	22
SEMESTER-VI	17	0	10	22
SEMESTER-VII	12	0	20	22
SEMESTER-VIII	12	1	18	22

Note – L: Lecture Hour, T: Tutorial Hour, P: Practical Hour, TC: Total Credits, IAE: Internal Assessment Examination, ESE: End Semester Examination.

Generic Electives (GE)
Generic Electives are credited and choice-based. The students make a choice from the pool of GE offered by the Faculty under the University.

Value Added Courses (VAC)
Value Added Courses are credited and choice-based. The students make a choice from the pool of VAC offered by the Faculty under the University.

Ability Enhancement Compulsory Course (AEC)
Ability Enhancement Compulsory Courses are credited and choice-based. The students make a choice from the pool of AEC offered by the Faculty under the University.

Skill Enhancement Courses (SEC)
Skill Enhancement Courses are credited and choice-based. The students make a choice from the pool of SEC offered by the Faculty under the University.

Internships, Apprenticeships, Project, Community outreach program (IAPC)
Internships, Apprenticeships, Community outreach programs, and Project is credited and choice-based. Students can undergo a 6-credit work-based project during semester VII and semester VIII.

SEMESTER-I

Course Type	Course Code	Course Title
DSC-1	0610111011	Foundations of IT
DSC-2	0610111021	C Programming
DSC-3	0610111031	Basics of Digital Electronics
GE-1		One from the pool of GE courses
SEC-1		One from the pool of SEC courses
AEC-1		One from the pool of AEC courses
VAC-1		One from the pool of VAC courses

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610111011
Course Title	Foundations of IT
Semester	I
Type of Course	Discipline Specific Core (DSC)
Credits	3+0+1
Course Prerequisites	–
Course Objective(s)	1. Familiarize students with the basic concepts, terminology, and components of Information Technology (IT), including hardware, software, and networking. 2. Develop proficiency in the use of common software applications such as word processors, spreadsheets, databases, and presentation tools for solving real-world problems. 3. Provide foundational knowledge of programming, covering basic logic, problem-solving, and an introduction to a simple programming language like HTML, CSS, JScript. 4. Explain the basics of computer networking, including the types of networks (LAN, WAN, etc.), network protocols, and how the internet works.
Course Outcome (CO)	CO1: Define and explain basic IT terminology, components of a computer system, and the role of software and hardware in IT. CO2: Demonstrate proficiency in using common office applications (word processing, spreadsheets, presentations) to perform a variety of tasks.

	CO3: Describe the key components of a computer system, including the CPU, memory, storage devices, and peripherals, and understand their functions. CO4: Explain how computer networks operate, the types of networks, and how the internet functions, including an understanding of IP addresses, DNS, and common network protocols.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	1	1	3	2	1	2	3	3	1	3	1	1
CO2	2	3	2	1	3	3	3	3	1	3	2	3
CO3	2	1	1	3	1	2	3	3	3	3	3	1
CO4	2	1	2	2	1	3	2	3	3	1	2	1
Avg.	1.75	1.5	2	2	1.5	2.5	2.75	3	2	2.5	2	1.5
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Overview of Information Technology: Definition and scope of IT, Basic concepts: data, information, knowledge, Evolution and history of IT, Computer Fundamentals: Components of a computer system (CPU, memory, storage), Types of computers: desktop, laptop, mobile devices, Input and output devices: keyboard, mouse, monitor, printer Software Basics: Types of software, system software vs application software Operating systems: functions and examples (Windows, macOS, Linux), Introduction to file management and organization Introduction to Networking, Basics of data communication: concepts and terminology, Local Area Network (LAN) vs Wide Area Network (WAN),Introduction to the Internet and World Wide Web (WWW)	**12**
UNIT II	Word Processing and Presentation Tools, Introduction to word processing software (Microsoft Word, Google Docs) Creating, editing, and formatting documents Creating and delivering presentations (Microsoft PowerPoint, Google Slides)	**12**

	Spreadsheet Basics Using spreadsheet software (Microsoft Excel, Google Sheets) Data entry, formulas, and functions (sum, average, count) Creating simple charts and graphs	
UNIT III	Introduction to Web Development, Basics of HTML and CSS Creating and styling simple web pages. Introduction to web hosting and domain management Digital Media Basics, Basics of digital media (images, audio, video), Introduction to graphic editing software (e.g., Paint, GIMP)	**11**
UNIT IV	Ethical and Legal Issues in IT, Privacy concerns and data protection, Intellectual property rights (copyrights, patents) Ethical use of IT resources and professional conduct IT Career Paths and Skills, Overview of IT career opportunities (e.g., software developer, IT support) Essential skills for IT professionals: communication, problem-solving, teamwork Resources for continuous learning and professional development	**10**
	Practical	**30**
	Total Theory + Practicals	**45 + 30 = 75**

LIST OF PRACTICALS

1. Demonstrate startup/shutdown procedures and basic navigation of operating systems.
2. Practice using common input and output devices (keyboard, mouse, printer).
3. Create a simple document with formatted text, headings, and lists.
4. Edit and format the document using basic formatting tools.
5. Create a budget spreadsheet with income, expenses, and formulas for calculations.
6. Generate a simple chart to visualize budget data.
7. Design a simple database for a personal address book or movie collection.
8. Create queries to retrieve specific information from the database.
9. Write a simple program in Scratch or similar visual programming language.
10. Create an algorithm and represent it using a flowchart.
11. Create a basic webpage using HTML and add styling using CSS.
12. Include text, images, and links on the webpage.
13. Edit a digital image using basic editing tools (crop, resize, adjust brightness).
14. Create a short slideshow using presentation software with images and transitions.
15. Design a mock online store interface with product listings and a shopping cart.
16. Discuss basic security measures for online transactions.
17. Analyze a case study related to ethical issues in IT and propose solutions.
18. Discuss ethical guidelines and professional conduct in the use of IT resources.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr.	Book Title	Author(s)	Edition

No.			
1	Introduction to Information Technology	V. Rajaraman	3rd, 2018
2	IT Essentials: PC Hardware and Software Companion Guide	Cisco Networking Academy	4th, 2010
3	Computer Basics Absolute Beginner's Guide	Michael Miller	8th, 2015

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610111021
Course Title	C Programming
Semester	I
Type of Course	Discipline Specific Core (DSC)
Credits	3+0+1
Course Prerequisites	–
Course Objective(s)	1. To introduce students to the basic concepts and structure of the C programming language, including data types, operators, and expressions. 2. To enable students to apply logical thinking and systematic approaches to solve problems through programming in C. 3. To familiarize students with control structures such as loops, conditional statements, and branching, and their application in real-world programming problems. 4. To teach students the concept of modular programming by dividing complex problems into functions and developing programs with reusable code.
Course Outcome (CO)	CO1: Explain the Core Concepts of C Programming CO2: Develop Efficient Programs Using Control Structures CO3: Implement Functions and Modular Programs CO4: Manipulate Arrays and Pointers Effectively

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	3	1	2	1	3	1	3	3	1
CO2	2	3	2	1	1	1	2	1	3	2	2	3
CO3	3	1	2	2	1	3	1	3	1	1	1	3
CO4	2	1	3	2	1	3	1	2	2	2	2	1
Avg.	2.5	1.75	2	2	1	2.25	1.25	2.25	1.75	2	2	2
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction to Programming Languages, Overview of programming languages, History and features of C language, Structure of a C program, Basic Concepts Keywords, identifiers, and data types, Constants and variables, Operators and expressions, Input and Output Basic input and output functions (printf, scanf), Formatted I/O, Control Structures, Decision-making: if, if-else, nested if, Switch-case statement, Looping: while, do-while, for loops Break, continue, and goto statements	**12**
UNIT II	Functions, Function declaration, definition, and call Types of functions (with and without parameters, with and without return values), Recursive functions Storage classes: auto, extern, static, and register Arrays, One-dimensional arrays: declaration, initialization, and accessing elements, Two-dimensional arrays: declaration, initialization, and accessing elements, Multidimensional arrays Array manipulation (insertion, deletion, searching, sorting)	**11**
UNIT III	Pointers, Introduction to pointers, Pointer arithmetic, Pointers and arrays, Pointers to pointers, Dynamic memory allocation (malloc, calloc, realloc, free) Strings, String handling functions (strlen, strcpy, strcat, strcmp, etc.), Array of strings, Pointers and strings, String manipulation (insertion, deletion, searching, concatenation)	**11**
UNIT IV	Structures and Unions, Defining and declaring structures, Accessing structure members, Arrays of structures, Pointers to structures, Nested structures, Unions and their usage File Handling, File operations: opening, closing, reading, and writing files, File pointers, Modes of file operation (read, write, append), Error handling during file operations, Random access to files (fseek, ftell, rewind)	**11**
	Practical	**30**
	Total Theory + Practicals	**45 + 30 = 75**

LIST OF PRACTICALS

1. Write a program to print "Hello, World!" and get user input for their name, then print a personalized greeting.
2. Write a program to perform and display the result of basic arithmetic operations (addition, subtraction, multiplication, division) using user input.
3. Write a program to find the largest of three numbers using if-else statements.

4. Write a program to display the grade of a student based on marks using switch-case statements.
5. Write a program to print the first 10 natural numbers using a for loop.
6. Write a program to calculate the factorial of a number using a while loop.
7. Write a program to generate the Fibonacci series up to a specified number using a do-while loop.
8. Write a program to calculate the area of a circle, square, and rectangle using separate functions.
9. Write a recursive function to find the GCD (Greatest Common Divisor) of two numbers.
10. Write a program to find the sum and average of elements in a one-dimensional array.
11. Write a program to perform matrix addition and matrix multiplication using two-dimensional arrays.
12. Write a program to reverse a string and check if it is a palindrome.
13. Write a program to count the number of vowels, consonants, digits, and spaces in a given string.
14. Write a program to swap two numbers using pointers.
15. Write a program to find the sum of elements in an array using pointers.
16. Write a program to dynamically allocate memory for an array of integers, accept user input, and find the sum of the elements.
17. Write a program to dynamically allocate memory for a matrix and perform matrix addition.
18. Write a program to create a structure to store student information (name, age, marks) and display it.
19. Write a program to create an array of structures to store information about multiple students and display it.
20. Write a program to demonstrate the use of a union to store different data types in the same memory location.
21. Write a program to read data from a file and display it on the screen.
22. Write a program to write user input to a file.
23. Write a program to append text to an existing file.
24. Write a program to count the number of characters, words, and lines in a text file.
25. Write a program to copy the contents of one file to another.
26. Write a program to concatenate two strings without using the strcat function.
27. Write a program to find the length of a string without using the strlen function.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition
1	Let us C	Yeahwant Kanetkar	17th, 2020
2	A First Course In Programming With C	T Jeyapoovan	1st, 2004
3	Mastering C	Venugopal K	2nd, 2017

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610111031
Course Title	Basics of Digital Electronics
Semester	I
Type of Course	Discipline Specific Core (DSC)
Credits	3+0+1
Course Prerequisites	–
Course Objective(s)	Demonstrate the operation of simple digital gates, identify the symbols, truth table for gates; change binary, hexadecimal, octal numbers to their decimal equivalent and vice versa, demonstrate the operation of a flip-flop. Convert digital into analog and vice versa
Course Outcome (CO)	The students will be able to: CO1. Develop a digital logic CO2. Get knowledge about flip flops. CO3. Understand, analyze and design various combinational and sequential circuits. CO4. Understand how to convert signals.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	2	1	2	1	1	3	2	1	2	3	1	3
CO2	2	2	2	1	2	1	2	2	3	2	2	1
CO3	1	1	1	3	1	3	2	1	2	3	3	3
CO4	1	1	1	3	3	2	1	2	2	1	3	3
Avg.	1.5	1.25	1.5	2	1.75	2.25	1.75	1.5	2.25	2.25	2.25	2.5
1: Weak Correlation, 2: Moderate Correlation, 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Fundamental concepts: Introduction, Digital Signals, Basic Gates and derived Gates: AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR, Boolean Algebra Number System and codes: Introduction to number systems, Decimal, Binary, Octal, Hexadecimal, And Conversation from one number system to another number system. Binary Arithmetic: Addition, Subtraction, Multiplication, Division, Half adder, full adder.1's and 2's complement of Binary Number. Codes : BCD Code, Excess-3 Code, Gray Code Error detecting and correcting codes	**12**
UNIT II	Combinational Logic Design: Standard Representation of logical functions, SOP, POS Forms, K-map Representation of logical functions, and Simplification of logical functions using K-map. Multiplexer, Demultiplexer. Encoder, Decoder	**12**
UNIT III	Flip Flops: 1-Bit Memory Cell, Clocked S-R Flip Flop, J-K Flip Flop, Master Slave Flip Flop, D-type Flip Flop, T-type Flip Flop	**11**
UNIT IV	Sequential Logic Design: Registers, Shift Register, Counter, Synchronous and asynchronous Counter, examples of each	**10**
	Practicals:	30
	Total Theory+Practicals:	**45+30=75**

LIST OF PRACTICALS

1. Implement logic gates (AND, OR, NOT, NAND, NOR, XOR, XNOR) using truth tables.
2. Simulate basic gates using software tools like Logisim or Proteus.
3. Simplify Boolean expressions using laws and theorems of Boolean algebra.
4. Implement and verify simplified Boolean expressions using logic gates.
5. Write a program to convert a decimal number to binary, octal, and hexadecimal formats.
6. Implement algorithms for converting between different number systems manually.
7. Write programs to perform binary addition, subtraction, multiplication, and division.
8. Implement algorithms for 1's and 2's complement of binary numbers and verify with examples.
9. Implement encoding and decoding algorithms for BCD (Binary Coded Decimal) and Excess-3 codes.
10. Design a program to convert between Gray code and binary numbers.

11. Design circuits for given logical functions and implement them using SOP (Sum of Products) and POS (Product of Sums) forms.
12. Simplify logical functions using Karnaugh maps (K-maps) and verify the results.
13. Build and test circuits for 4:1 and 8:1 multiplexers using logic gates.
14. Implement demultiplexer circuits and demonstrate their operation.
15. Design an encoder circuit for a 4-to-2 line encoder and verify its functionality.
16. Implement a decoder circuit for a 2-to-4 line decoder and test its operation.
17. Construct and test flip-flop circuits including S-R flip-flop, J-K flip-flop, D flip-flop, and T flip-flop using logic gates.
18. Simulate flip-flops using software simulations and observe their behavior.
19. Design a 4-bit shift register circuit and demonstrate shifting operations.
20. Build synchronous and asynchronous counters (e.g., up-counter, down-counter) and verify their operation.
21. Implement error detection using parity bit methods (even parity, odd parity) for binary data.
22. Design and test a circuit for error detection using Hamming code.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	

Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition
1	Modern Digital Electronics	R.P. Jain	5th, 2022
2	Microprocessor	B.RAM	8th, 2021
3	Digital Electronics and Logic	B. Somanathan Nair	1st, 2002

SEMESTER-II

Course Type	Course Code	Course Title
DSC-4	0610121041	Numerical Analysis
DSC-5	0610121051	Object Oriented Programming with C++
DSC-6	0610121061	Operating System Principles
GE-I		One from the pool of GE courses
SEC-1		One from the pool of SEC courses
AEC-1		One from the pool of AEC courses
VAC-1		One from the pool of VAC courses

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610121041
Course Title	Numerical Analysis
Semester	II
Type of Course	Discipline Specific Core (DSC)
Credits	3+0+1
Course Prerequisites	Basic Knowledge about Computers
Course Objective(s)	The objective of this course is to introduce the numerical techniques of interpolation in various intervals in real life situations and to acquaint the student with understanding of numerical techniques of differentiation and integration which plays an important role in technical disciplines.
Course Outcome (CO)	The students will be able to: CO1. Apply various interpolation methods and finite difference concepts CO2. Work out numerical differentiation and integration whenever and wherever routine methods are not applicable. CO3. Work numerically on the ordinary differential equations using different methods through the theory of finite differences. CO4. Work numerically on the partial differential equations using different methods through the theory of finite differences.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	1	3	1	2	1	2	3	1	2	1	3	3
CO2	2	3	3	1	2	2	1	2	1	1	3	3
CO3	1	2	2	3	1	3	3	3	2	2	2	1
CO4	2	2	2	2	1	1	2	1	3	3	1	1
Avg.	1.5	2.5	2	2	1.25	2	2.25	1.75	2	1.75	2.25	2
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	**Introduction:** Numbers and their accuracy, Computer Arithmetic, Mathematical preliminaries, Errors and their Computation, General error formula, Error in a series approximation Solution of Algebraic and Transcendental **Equation:** Bisection Method, Iteration method, Method of false position, Newton-Raphson method, Methods of finding complex roots, Muller's method, Rate of convergence of Iterative methods, Polynomial equations.	**12**
UNIT II	**Interpolation:** Finite Differences, Difference tables**, Polynomial Interpolation:** Newton's forward and backward formula**, Central Difference Formulae:** Gauss forward and backward formula, Stirling's, Bessel's, Everett's formula. **Interpolation with unequal intervals:** Lagrange's Interpolation, Newton Divided difference formula, Hermite's Interpolation	**12**
UNIT III	**Solution of differential Equations:** Picard's Method, Euler's Method, Taylor's Method, Runge-Kutta Methods, Predictor Corrector Methods, Automatic Error Monitoring and Stability of solution	**11**
UNIT IV	**Statistical Computation:** Frequency chart, Curve fitting by method of least squares, fitting of straight lines, polynomials, exponential curves etc, Data fitting with Cubic splines, Regression Analysis, Linear and Nonlinear Regression, Multiple regression, Statistical Quality Control methods.	**10**
	Practicals:	30

	Total Theory+Practicals:	**45+30=75**

LIST OF PRACTICALS

1. Representation of Numbers and Accuracy in MATLAB
2. General Error Formulas and Computation in MATLAB
3. Iteration Method (Fixed-Point Iteration): MATLAB Application
4. Finite Differences: MATLAB Calculation and Visualization
5. Picard's Method: MATLAB Execution and Visualization
6. Euler's Method: MATLAB Simulation and Analysis
7. Taylor's Method: MATLAB Implementation
8. Frequency Charts and Histograms: MATLAB Plotting
9. Linear Regression: MATLAB Implementation
10. Statistical Quality Control Methods: MATLAB Techniques

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	

Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition
1	Computer Oriented Numerical Methods	Rajaraman V	4th, 2019
2	Applied Numerical Analyses	Gerald & Whealey	7th, 2007
3	Numerical Methods for Scientific and Engineering Computations	Jain, Iyengar and Jain	7th, 2019

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610121051
Course Title	Object Oriented Programming with C++
Semester	II
Type of Course	Discipline Specific Core (DSC)
Credits	3+0+1
Course Prerequisites	–
Course Objective(s)	To gain experience about structured programming. To help students to understand the implementation of Programming language. To understand various features in Programming Language.
Course Outcome (CO)	The students will be able to: CO1. Understand how **C++** improves C with object-oriented features. CO2. Learn how to write inline functions for efficiency and performance. CO3. Learn the syntax and semantics of the **C++** programming language. CO4. Learn how to design **C++** classes for code reuse.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	2	1	1	1	3	2	1	2	2	2	3	3
CO2	3	3	1	1	1	3	3	2	1	1	2	1
CO3	1	2	3	3	1	3	2	3	3	3	3	3
CO4	3	3	3	1	2	2	3	3	2	2	1	2
Avg.	2.25	2.25	2	1.5	1.75	2.5	2.25	2.5	2	2	2.25	2.25
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	**Introduction to C++ :** Variables, Constants, Data Types and Operators in C++, Construction of C++, Keywords in C++, Input/Output in C++, Precedence of Operators, Characteristics of object oriented languages **Introduction to OOP:** Advantages of OOP, Need of object-oriented programming, Procedure Oriented Vs Object Oriented Programming.	**12**
UNIT II	**Object-Oriented Modeling and Design :** Object Oriented Concepts, Objects and Classes, Characteristics of Objects Identity, Abstraction, Classification, Polymorphism, Inheritance, Object Oriented Models, Object Model, dynamic Model, Functional Model, Links and Associations, Generalization, Grouping Constructs, Metadata, Object design	**12**
UNIT III	**Decision Making and Loops in C++ :** Conditional statement, Switch Statement, Break Statement, Continue Statement, Go to Statement, **Loops in C++**: While, Do-While, For loop.**Functions :** User Defined Functions, library functions, General form of a function, scope rules of functions, function arguments(Call by value, Call by Reference), Recursion Calling Functions with arrays, Returning by reference, Friend Functions, Inline Functions, Structures and Unions in C++, Pointers in C++, Pointers with structure, Pointer with functions.	**11**
UNIT IV	**Objects and classes :** Structure and Classes, Union and Class, friend classes, Scope resolution operator, specifying and using class and object, Constructors, objects and function arguments. **Inheritance:** Base Class, Derived Class, access specifiers, Single Inheritance, Multiple Inheritance, Multilevel Inheritance. **Polymorphism:** Compile time, Run time, Operator Overloading, Function Overloading, Virtual functions, Dynamic Binding, Static Binding	**10**
	Practicals:	30
	Total Theory+Practicals:	**45+30=75**

LIST OF PRACTICALS

1. Write a C++ program to find the sum of individual digits of a positive integer.
2. Write a C++ program to generate the first n terms of the sequence.

3. Write a C++ program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
4. Write a C++ program to find both the largest and smallest number in a list of integers.
5. Write a C++ program to sort a list of numbers in ascending order
6. Write a program Illustrating Class Declarations, Definition, and Accessing Class Members.
7. Program to illustrate default constructor, parameterized constructor and copy constructors
8. Write a Program to Demonstrate the i)Operator Overloading.ii) Function Overloading.
9. Write a Program to Demonstrate Friend Function and Friend Class.
10. Write a Program to Generate Fibonacci Series using Constructors to Initialize the Data Members.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	

Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition
1	The Complete Reference C++	Herbort Schildt	4th, 2017
2	Object Oriented Programming with C++	E. Balaguruswamy	8th, 2020
3	Object Oriented Modeling and Design	James Rambaugh	2nd, 2011

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610121061
Course Title	Operating System Principles
Semester	II
Type of Course	Discipline Specific Core (DSC)
Credits	3+0+1
Course Prerequisites	–
Course Objective(s)	The objective of this course is to help students become familiar with the fundamental concepts of operating systems and provide students with sufficient understanding of operating system design.
Course Outcome (CO)	The students will be able to: CO1. Describe the importance of computer system resources and the role of operating system in their management policies and algorithms CO2. Understand the process management policies and scheduling of processes by CPU CO3. Evaluate the requirement for process synchronization and coordination handled by operating system CO4. Describe and analyze the memory management and its allocation policies.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	2	1	2	1	3	2	3	2	3	2	2	2
CO2	2	3	3	1	3	3	1	3	2	3	1	1
CO3	3	1	2	1	2	3	2	1	3	1	3	2
CO4	3	3	3	2	1	3	1	3	1	2	2	3
Avg.	2.5	2	2.5	1.25	2.25	2.75	1.75	2.25	2.25	2	2	2
1: Weak Correlation, 2: Moderate Correlation, 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	**Introduction:** Importance of Operating system. Basic concepts and terminology, An Operating system Resource manager, Operating Systems functions, Services provided by operating system, Types of operating systems, An Operating system-Process and system calls, Operating system architecture, Processor and user modes, Virtual Machine.	**12**
UNIT II	**Memory management:** Physical and virtual address space, Single Contiguous Allocation, Partitioned Allocation, Fragmentation, Paging, Segmentation, Virtual memory and Demand paging, Page replacement algorithms. **Processor Management**: Process, Process control block, State Model, Non-pre-emptive and preemptive scheduling, Process Scheduling Algorithms, Deadlocks- detection and prevention.	**12**
UNIT III	**Information Management**: Directory structure, File operations, A Simple File System, General Model of a File System, File allocation methods.	**11**
UNIT IV	**OS and Security:** Security breaches, types of attacks, attack prevention methods, security policy and access control, OS design considerations for security, access, policy and access control, OS design considerations for security, access control lists and OS support, internet and network security, Policy mechanism, Program, network and system threats, Authentication.	**10**
	Practical	**30**
	Total Theory + Practicals	**45 + 30 = 75**

LIST OF PRACTICALS

1. Shell Programming
2. Implement all file allocation strategies
3. Sequential b) Indexed c) Linked
4. Implement all File Organization Techniques
5. Implement basic commands of UNIX and LINUX
6. Write a program in shell programming to find the sum of five digit numbers.
7. WAP to implement Bankers algorithm for deadlock avoidance.
8. WAP to implement Bankers algorithm for deadlock prevention.
9. WAP to implement the paging technique of memory management.
10. WAP to implement following CPU scheduling algorithms:
    a. FCFS

b. SJF
c. Priority
d. Round Robin

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition
1	Operating System Concepts	Peterson & Silberschatz	8th, 2019
2	Modern Operating System	Tenenbaum, A.S	4th, 2016

SEMESTER-III

S.No.	Course Type	Course Code	Course Title
1	DSC-7	0610131071	Fundamental of DBMS
2	DSC-8	0610131081	Data Structures
3	DSC-9	0610131091	Fundamentals of Python
4	DSE-1/ DSE-2/ DSE-3/ GE-3	0610132011/0610132021/0610132031	Introduction to Web Technologies/ Linux & Shell Scripting/ Java Programming/ GE-3(One From the pool of GE Courses)
5	SEC-3		One from the pool of SEC courses
6	AEC-3		One from the pool of AEC courses
7	VAC-3		One from the pool of VAC courses

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610131071
Course Title	Fundamental of DBMS
Semester	III
Type of Course	Discipline Specific Core (DSC)
Credits	3+0+1
Course Prerequisites	-
Course Objective(s)	1. Understand the Fundamentals of DBMS 2. Apply Entity-Relationship (ER) Modeling Techniques 3. Work with SQL and Integrity Constraints 4. Manage Data Redundancy and Apply Normalization Techniques 5. Understand and Implement Transaction Management
Course Outcome (CO)	CO1: Understand and explain basic concepts, architecture and roles in a DBMS system. CO2: Design and implement Entity-Relationship diagrams considering keys, constraints and design issues. CO3: Apply SQL for database operations and manage data integrity through constraints. CO4: Analyze and improve database design using Normal Forms to eliminate data redundancy.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	1	2	1	1	3	2	1	2	3	1	3
CO2	2	2	2	1	2	1	2	2	3	2	2	1
CO3	1	1	1	3	1	3	2	1	2	3	3	3
CO4	1	1	1	3	3	2	1	2	2	1	3	3
Avg.	1.5	1.25	1.5	2	1.75	2.25	1.75	1.5	2.25	2.25	2.25	2.5
1: Weak Correlation, 2: Moderate Correlation, 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	**Introduction:** Concept & overview of DBMS, data models, database languages, database administrator, Database Users, Three Schema architecture of DBMS. **Entity-Relationship Model:** Basic concepts, design issues, mapping constraints, keys, entity-relationship diagram.	**11**
UNIT II	**SQL and Integrity Constraints:** Concept of DDL(CREATE, ALTER, DROP, TRUNCATE), DML(SELECT, INSERT, UPDATE, DELETE), DCL(GRANT, REVOKE), set operations(UNION, INTERSECT, EXCEPT), aggregate functions(COUNT, SUM, AVG, MIN, MAX), nested sub queries	**12**
UNIT III	**Data Redundancy and Normalization**: Fundamental dependencies, Normalization for Relational Database, Functional dependencies, Normal Forms (1NF, 2NF, 3NF, BCNF, 4NF & 5NF)	**11**
UNIT IV	**Transaction Management:** Transaction Concepts & State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability. **Failure Recovery and Concurrency Control:** logging-protecting against media failures.	**11**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practical:

1. To implement Data Definition Commands (create, drop).
2. To implement Data Manipulation Commands (insert, delete, update, select)
3. To implement Data Control Commands (Grant, Revoke)
4. Create Table, SQL for Insertion, Deletion, Update and Retrieval using aggregate functions.
5. Perform the following: a. Altering a Table, Dropping/Truncating/Renaming Tables, Backing up / Restoring a Database.
6. Given the table EMPLOYEE (EmpNo, Name, Salary, Designation, DeptID) , write a cursor to select the five highest paid employees from the table.
7. Write Program for Join, Union & intersection etc.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	

Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Database System and Concepts	A Silberschatz, H Korth, S Sudarsha
2	Database Systems	Rob Coronel

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610131081
Course Title	Data Structures
Semester	III
Type of Course	Discipline Specific Core (DSC)
Credits	3+0+1
Course Prerequisites	Basic Knowledge of Computer and C Programming
Course Objective(s)	1. Understand fundamental data structures like arrays, linked lists, stacks, and queues. 2. Implement efficient searching, sorting, and recursion techniques. 3. Analyze algorithm complexity using Big-O notation. 4. Explore tree and graph structures for problem-solving. 5. Develop practical programming skills using C.
Course Outcome (CO)	The students will be able to: CO1. Apply data structures like arrays, linked lists, stacks, and queues in problem-solving. CO2. Implement efficient searching, sorting, and recursion techniques in programming. CO3. Analyse algorithm complexity and optimize performance using Big-O notation. CO4. Utilize tree and graph structures for real-world applications.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	1	2	2	2	1	2	2	2	2	3	3	1
CO2	2	2	2	2	1	2	3	2	2	1	3	2
CO3	1	3	2	1	2	3	1	3	3	1	2	2
CO4	3	3	3	3	2	2	2	2	2	3	2	3
Avg.	1.75	2.5	2.25	2	1.5	2.25	2	2.25	2.25	2	2.5	2
1: Weak Correlation, 2: Moderate Correlation, 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	**Introduction:** Data types, data structures, abstract data types, the running time of a program, the running time and storage cost of algorithms, complexity, Asymptotic Notation **Development of Algorithms:** Notations and Analysis, Storage structures for arrays - sparse matrices - structures	**13**
UNIT II	**Stacks-** Introduction & Definition, Application of Stack, Various Representation of Stack, Operation on stack (Push and Pop), Representation of Arithmetic Expression (Infix, Postfix, Prefix). Evaluation of postfix expressions and their conversions **Queues-** Introduction, Applications of Queue, Various Representations of Queue, Operation on queue. Priority Queues, Circular Queue.	**12**
UNIT III	**Recursion and Link List** **Recursion-** Introduction, Recursion Properties, Applications of Recursion (Factorial, Fibonacci Series etc), Advantages and Disadvantages of Recursion. **Linked List-** Introduction, Application of Linked List, and Representation of Linked List, Operation on Linked List (Insertion, Deletion). Concept of Doubly and Circular Linked List	**12**
UNIT IV	**Trees-** Introduction, Definition of Trees, Binary Tree, Type of Binary Tree, Operation on Binary Tree, Traversal of Binary Tree, Binary Search Tree (BST), Memory Representation of Binary Tree **Sorting & Searching-** Bubble sort, Insertion sort, Selection sort,, Radix sort,Hashing, Linear Search and Binary Search	**8**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

LIST OF PRACTICALS

1. Write a program for insertion sort
2. Write a program to implement Stack and its operation.
3. Write a program for Bubble sort.
4. Write a program to implement Queue and its operation.
5. Write a program to implement Circular Queue and its operation.
6. Write a program to implement a singly linked list for the following operations: Create, Display, searching and traversing.
7. Write a program to implement a singly linked list for the following operations: Insertion and Deletion.
8. Write a program to implement Tree Traversals .
9. Write a program for Linear Search
10. Write a program for Binary Search

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	

Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition
1	Lipschutz Schaum's "Data Structure" Outline Series [TMH]. ISBN-0-07-060168-2	Lipschutz Schaum	Sixth
2	Tenenbaum," Data Structures Using C and C++", Prentice Hall India, New Delhi. ISBN-81317-0328-2	Tenenbaum	Second
3	Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein. Introduction to Algorithms, PHI Learning, 2009. ISBN:978-81-203-4007-7.	Thomas H. Cormen	Third

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610131091
Course Title	Fundamentals of Python
Semester	III
Type of Course	Discipline Specific Core (DSC)
Credits	3+0+1
Course Prerequisites	–
Course Objective(s)	To gain experience in structured programming using Python. To help students understand the basic implementation of programming concepts in Python. To introduce Python's simple syntax and powerful features.
Course Outcome (CO)	The students will be able to: CO1. Understand Python syntax, variables, and data types. CO2. Apply conditional statements and loops to solve basic problems. CO3. Use functions, lists, tuples, and dictionaries in Python programs. CO4. Implement simple object-oriented concepts such as classes and objects in Python.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	2	1	1	1	3	2	1	2	2	2	3	3
CO2	3	3	1	1	1	3	3	2	1	1	2	1
CO3	1	2	3	3	1	3	2	3	3	3	3	3
CO4	3	3	3	1	2	2	3	3	2	2	1	2
Avg.	2.25	2.25	2	1.5	1.75	2.5	2.25	2.5	2	2	2.25	2.25
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	**Introduction to Python, Python Features, Operators, Variables, Computational Thinking and Problem-Solving:** Basics of Python, Features of Python, Variables and Data Types, Arithmetic and Logical Operators, Basic Input and Output. Introduction to Algorithms, Flowcharts, and Pseudocode. Simple problem-solving using sequences and decision-making.	**12**
UNIT II	**Conditional Statements – Boolean Expressions – If/Else Statements – Iteration – Using Functions – Modules – Exception Handling (Basics Only):** Boolean Expressions. Conditional Statements: if, if-else. Loops: for, while, and break, continue. Functions: Definition, Calling Parameters, and Return Values. Using Built-in Functions. Basic Exception Handling with try and except.	**12**
UNIT III	Lists, Tuples, Dictionaries – List Operations – Tuples – Dictionary Operations – **Data Structure Traversal:** Lists: Creation, Access, Basic Methods (append(), remove()). Tuples: Creation and Use. Dictionaries: Creation and Access. Traversing Lists and Dictionaries using Loops.	**11**
UNIT IV	Files, Modules, Reading and Writing Files – Creating Classes – Simple Methods – **Exception Handling (Basics):** File Handling: Reading and Writing Text Files. Modules: Using Standard Modules (e.g., math). Basic Exception Handling with try and except.	**10**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

LIST OF PRACTICALS

1. Write a Python program to display “Hello, World!” and demonstrate basic input and output.
2. Write a Python program to perform arithmetic operations (+, -, *, /) on two numbers input by the user.
3. Write a Python program to demonstrate the use of logical operators (and, or, not).
4. Write a Python program to solve a simple problem using sequence and decision making (e.g., check if a number is positive, negative, or zero).
5. Write a Python program to find the largest of three numbers using if-else statements.
6. Write a Python program to print all even numbers from 1 to 50 using a for loop.
7. Write a Python function to calculate the factorial of a number using a while loop.
8. Write a Python program demonstrating the use of try and except to handle division by zero error.

9. Write a Python program to create a list of 5 elements and perform operations: append, remove, and display the list.
10. 10. Write a Python program to create a tuple and print its elements.
11. Write a Python program to write user input to a text file.
12. Write a Python program to import and use the math module to calculate the square root of a number.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination

End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition
1	Python Programming: An Introduction to Computer Science	John Zelle	3rd, 2017
2	Think Python: How to Think Like a Computer Scientist	Allen B. Downey	2nd, 2015
3	Learning Python	Mark Lutz	5th, 2013
4	Python for Everybody: Exploring Data Using Python 3	Charles R. Severance	1st, 2016
5	Automate the Boring Stuff with Python	Al Sweigart	2nd, 2019

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610132011
Course Title	Introduction to Web Technologies
Semester	III
Type of Course	Discipline Specific Core (DSE)
Credits	3+0+1
Course Prerequisites	Foundations of IT
Course Objective(s)	The aim of the course is to provide knowledge of the web as a tool in presenting information. Each and every product in the e-world now needs a website, this course will make students know about the concept of web design in general.
Course Outcome (CO)	At the end of this course, the student will be able to: CO1: Learn about WWW and search engines. CO2: Understand the domain and assign names to them. CO3: Understand basic web languages and its components. CO4: Perform simple web page designing for practical exposure.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PSO 1**	**PSO 2**	**PSO 3**	**PSO 4**	**PSO 5**
CO1	1	1	3	2	1	2	3	3	1	3	1	1
CO2	2	3	2	1	3	3	3	3	1	3	2	3
CO3	2	1	1	3	1	2	3	3	3	3	3	1
CO4	2	1	2	2	1	3	2	3	3	1	2	1
Avg.	1.75	1.5	2	2	1.5	2.5	2.75	3	2	2.5	2	1.5
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction to Internet and World Wide Web; Evolution and History of World Wide Web; Basic features; Web Browsers; Web Servers; Hypertext Transfer Protocol; URLs; Searching and Web- Casting Techniques; Search Engines and Search Tools	**10**
UNIT II	Web Publishing: Hosting your Site; Internet Service Provider; Planning and designing your Web Site; Steps for developing your Site; Choosing the contents; Home Page; Domain Names; Creating a Website and the Markup Languages (HTML, DHTML)	**11**
UNIT III	Web Development: Introduction to HTML; Hypertext and HTML; HTML Document Features; HTML command Tags; Creating Links; Headers; Text styles; Text Structuring; Text colors and Background; Formatting text; Page layouts;Images; Ordered and Unordered lists; Inserting Graphics; Table Creation and Layouts;	**10**
UNIT IV	Frame Creation and Layouts; Working with Forms and Menus; Working with Radio Buttons; Check Boxes; Text Boxes. CSS: Introduction, Types of style sheets, Style specification formats, Font properties, List properties, Color, Alignment of text, Background images, The and tags, Features of CSS3.	**14**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

IST OF PRACTICALS

1. Write code to generate Simple student bio-data
2. Write code to show Different types of frames
3. Write code which uses Different styles
4. Write a code To read .XML file
5. Write a code for Sorting the values
6. Write the code for Exception handling
7. Write the code to generate Single text field calculator
8. Write the code for Session handling
9. Write the code to use Regular Expression
10. Write the code for Email processing
11. Write the code for the Login database.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition
1	Internet and Web Technologies	Raj Kamal	TataMcGraw-Hill
2	Multimedia and Web Technology	Ramesh Bangia	Firewall Media.

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610132021
Course Title	Linux & Shell Scripting
Semester	III
Type of Course	Discipline Specific Elective (DSE)
Credits	3+0+1
Course Prerequisites	–
Course Objective(s)	1.To comprehend the basic UNIX concepts related to concurrency and control of programs. 2.To Identify and define key terms related to operating system
Course Outcome (CO)	CO1: Understand the UNIX environment, File system and hierarchy CO2: Identify high-level steps such as verifying user input to automate repetitive tasks CO3: Apply shell scripting techniques and standards using filters for pattern matching on plain text data and variety of system log files CO4: Develop effective and interactive scripts using functional blocks, operating system in real time environment

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO5**	**PO6**	**PO7**	**PSO 1**	**PSO 2**	**PSO 3**	**PSO 4**	**PSO 5**
CO1	3	2	1	1	1	2	2	2	2	1	3	1
CO2	2	3	2	1	2	2	3	3	3	2	2	2
CO3	2	3	2	1	2	2	2	3	3	2	3	2
CO4	2	3	3	1	2	2	3	3	3	2	2	3
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	**Introduction:** History - Salient features of LINUX - LINUX system architecture. Features of Linux, Advantages and disadvantages of Linux. Introduction to File System, Memory Management	**10**
UNIT II	**Basic Commands:** login, logout, date, man, pwd, who, whoami, dir, ls, cd,mkdir, rmdir. Use of Wild card characters, Introduction to vi editor. Introduction to environment variables like HOME, PATH, PS1. Types of FAP, use of chmod command. Basic commands like cp, mv, rm, rev, file redirection. cut, paste, find sort commands with example	**12**
UNIT III	**Shell Scripting** **Shell Programming - Basics:** Types of Shells - Shell variables - Shell parameters. Introduction to shell script: execution of it, shell script variable expr, test commands Control structure: if, until, for, continue. Iteration: while, for construct, break, continue, exit commands	**12**
UNIT IV	**Advance Shell Scripting-** Using advanced features of the shell, Input/ Output Redirection The grep filter, More utility programs, The sed filter	**11**
	Practicals:	30
	Total Theory+Practicals:	**45+30=75**

LIST OF PRACTICALS

1. Shell Script that copies multiple files to a directory, Shell Script (small calculator) that adds, subtracts, multiplies and divides the given two integers
2. Write a Shell script to read command line arguments, to execute logical operators, if, while and to evaluate expressions, Write a Shell script to read command line arguments, to execute logical operators, if, while and to evaluate expressions
3. Write a Shell script to read command line arguments, to execute logical operators, if, while and to evaluate expressions, Script on files with their file permissions
4. Scripts on accessing system resources for commands like who, passwd and others, Scripts for performing mathematical operations with using different looping controls, Scripts on usage of command line arguments, Scripts on usage of different environment variables
5. Scripts on usage of various shell variables, Scripts to understand various filters, Scripts to understand the usage of various system and general commands at different levels.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	45

Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition

1	Your UNIX-The Ultimate Guide	Sumitabha Das	Tata McGrawHill, 3rd
2	Beginning Shell Scripting	Eric Foster-Johnson, JohnCWelch, MicahAnderson	Wrox publication
3	A Complete Course on Linux bash shell scripting with real life examples	Imran Afzal	July 2019

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610132031
Course Title	Java Programming
Semester	III
Type of Course	Discipline Specific Elective (DSE)
Credits	3+0+1
Course Prerequisites	–
Course Objective(s)	To understand an integrated development environment to write, compile, run, and test simple object oriented Java programs.
Course Outcome (CO)	At the end of this course, the student will be able to: CO1: Understand the fundamentals of Java programming, including data types, control structures, classes, and objects. CO2: Apply object-oriented programming principles such as inheritance, polymorphism, abstraction, and encapsulation in Java. CO3: Develop GUI-based and event-driven applications using Java libraries and frameworks. CO4: Implement file handling, exception management, and multithreading to build robust and efficient Java applications.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	1	1	3	2	1	2	3	3	1	3	1	1
CO2	2	3	2	1	3	3	3	3	1	3	2	3
CO3	2	1	1	3	1	2	3	3	3	3	3	1
CO4	2	1	2	2	1	3	2	3	3	1	2	1
Avg.	1.75	1.5	2	2	1.5	2.5	2.75	3	2	2.5	2	1.5
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction to Java. Features of java - JDK Environment & tools like(java, javac, appletviewer, javadoc, jdb), Object Oriented Programming Concept- Overview of Programming, Paradigm, Classes, Abstraction, Encapsulation, Inheritance, Polymorphism, Difference between C++ and JAVA, Java Programming Fundamental- Structure of java program, Data types, Variables, Operators, Keywords, Naming Convention, Decision Making (if, switch), Looping(for, while), Type Casting	**11**
UNIT II	Classes and Objects -Creating Classes and objects, Memory allocation for objects, Constructor, Implementation of Inheritance, Simple Multilevel Hierarchical, Implementation of Polymorphism, Method Overloading, Method Overriding, Nested and Inner classes. Arrays, String and Vector -Arrays : Creating an array, Types of Array - One Dimensional arrays, Two Dimensional array. String – ,String Methods, String Buffer class, Vectors, Wrapper classes,	**14**
UNIT III	Abstract Class , Interface and Packages, Modifiers and Access Control, Default, public private protected, Abstract classes and methods, Interfaces, Packages-Packages Concept, Creating user defined packages,Java Built in packages.	**10**
UNIT IV	Exception Handling- Exception types, Using try catch and Multiple catch, Nested try, throw , throws and finally,Creating User defined Exceptions, File Handling- Byte Stream, character stream, file IO Basics, File Operations -Creating file,Reading file, Writing File.	**10**
	Practicals:	30
	Total Theory+Practicals:	**45+30=75**

List of Practical

1. Write a Program in Java to implement the Classes and Objects
2. write a program in Java with Constructors and destructors
3. write a program in Java to implement method overloading
4. write a program in Java to implement Inheritance.
5. write a program in Java to implement Interface
6. write a Program in Java to implement Multi Thread.
7. write a program in Java to implement Packages
8. write a program in java to create Java Applets

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours

Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition

1	Head First Java	Cathy Sierra, Bert Bates	O'Reilly
2	Java for Dummies	Barry Bird	Willey.

SEMESTER-IV

S.No.	Course Type	Course Code	Course Title
1	DSC-10	0610141101	R Programming
2	DSC-11	0610141111	Algorithm Design and Analysis
3	DSC-12	0610141120	Introduction to Linear Algebra
4	DSE-4/ DSE-5/ DSE-6/ GE-4	0610142041/0610142051/0610142061	Fundamentals of Cyber Security/ Network Programming/ Data Visualization/GE-4 (One From the pool of GE Courses)
5	SEC-4		One from the pool of SEC courses
6	AEC-4		One from the pool of AEC courses
7	VAC-4		One from the pool of VAC courses

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610141101
Course Title	R Programming
Semester	IV
Type of Course	Discipline Specific Core (DSC)
Credits	3+0+1
Course Prerequisites	--
Course Objective(s)	This course aims to introduce students to the fundamentals of R programming for statistical computing and data analysis. It equips learners with skills to write R scripts, manipulate data, perform exploratory data analysis, and create visualizations using R'w powerful libraries.
Course Outcome (CO)	At the end of this course, the students will be able to: CO1. Understand and apply the basic syntax, data types, and control structures in R programming. CO2. Perform data import, cleaning, and transformation usingR functions and packages. CO3. Analyze and visualize data using built-in and external R libraries. CO4. Implement statistical and exploratory data analysis for solving real world data problems.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	1	1	3	2	1	2	3	3	1	3	1	1
CO2	2	3	2	1	3	3	3	3	1	3	2	3
CO3	2	1	1	3	1	2	3	3	3	3	3	1
CO4	2	1	2	2	1	3	2	3	3	1	2	1
Avg.	1.75	1.5	2	2	1.5	2.5	2.75	3	2	2.5	2	1.5
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	**Introduction to R**: History, Features, and Installation, RStudio interface and basic commands, Data types in R: numeric, character, logical, complex, raw, Variables and Operators: Assignment, Arithmetic, Relational, Logical, Control Structures: if, else, switch, repeat, while, for	**11**
UNIT II	**Data Structures and Functions in R**:Vectors, Lists, Matrices, Arrays, Factors and Data Frames. Indexing and Subsetting of data structures, Creating and using functions in R, Scope of variables in functions, Apply family of functions: apply(), lapply(), sapply(), tapply()	**14**
UNIT III	**Data Manipulation and File Handling:** Reading and writing data from/to CSV, Excel, and text files, Data import using read.table(), read.csv(), readxl, readr, Data cleaning and preprocessing, Sorting, filtering, merging, and reshaping data frames, Data transformation using dplyr: select(), filter(), mutate(), arrange(), summarize(), group_by()	**10**
UNIT IV	**Data Visualization and Statistical Analysis:** Introduction to data visualization, Base R plotting functions: plot(), hist(), boxplot(), barplot(), Advanced plots using ggplot2, Descriptive statistics and summary functions, Introduction to basic statistical tests: mean, median, mode, SD, t-test, correlation, Mini-project on real-life dataset (optional practical component)	**10**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

LIST OF PRACTICALS

1. Write a program to display basic R commands: arithmetic operations, variable assignments, and use of built-in constants.
2. Write a program that uses control structures:
3. Check if a number is even or odd using if...else.
4. Print numbers from 1 to 10 using a for loop.
5. Write a program to find the factorial of a number using while loop and repeat loop.
6. Write a program to create and manipulate different data structures in R:
7. Create a vector, list, matrix, and data frame.
8. Perform basic operations (indexing, slicing).
9. Write a program to convert a numeric vector into a factor and summarize the factor levels.
10. Write a function in R that calculates the area of a circle and returns the result.
11. Write a program using apply(), lapply(), sapply() to perform row-wise and column-wise operations on a matrix.
12. Write a program to read a CSV file using read.csv() and display summary statistics.
13. Write a program to filter rows in a data frame based on a condition using dplyr::filter() and subset().
14. Write a program to group and summarize data using dplyr::group_by() and summarize().
15. Write a program to merge two data frames and demonstrate inner join, left join, and full join using dplyr.
16. Write a program to sort and arrange a data frame by column values using arrange().
17. Write a program to create a histogram, bar chart, and box plot using base R.
18. Write a program to create a scatter plot and line graph using ggplot2.
19. Write a program to calculate basic statistical measures (mean, median, mode, standard deviation) and perform a correlation test.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	

Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition
1	Hands-On Programming with R (O'Reilly Media)	Garrett Grolemund	
2	R for Data Science	Hadley Wickham and Garrett Grolemund	

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610141111
Course Title	Algorithm Design and Analysis
Semester	IV
Type of Course	Discipline Specific Core (DSC)
Credits	3+0+1
Course Prerequisites	Basic Knowledge of Data Structures and Problem Solving.
Course Objective(s)	To equip students with fundamental algorithmic techniques and efficiency analysis methods, fostering problem-solving skills through the design and application of optimized algorithms for real-world software development tasks.
Course Outcome (CO)	The students will be able to: CO1.Apply algorithmic techniques like divide and conquer, dynamic programming, and greedy algorithms. CO2. Analyze algorithm efficiency using asymptotic notation and complexity classes. CO3. Design optimized algorithms for computational problem-solving. CO4. Implement graph algorithms, searching, sorting, and optimization techniques.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	1	2	2	2	1	2	2	2	2	3	3	1
CO2	2	2	2	2	1	2	3	2	2	1	3	2
CO3	1	3	2	1	2	3	1	3	3	1	2	2
CO4	3	3	3	3	2	2	2	2	2	3	2	3
Avg.	1.75	2.5	2.25	2	1.5	2.25	2	2.25	2.25	2	2.5	2
1: Weak Correlation, 2: Moderate Correlation, 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	**Basics of algorithms:** Algorithms and characteristics, algorithm design paradigms, fundamentals of algorithmic problem solving, fundamental data structures. **Analysis of algorithms:** The efficient algorithm-average, Worst- and best-case analysis, Asymptotic notations, Time Complexity	**9**
UNIT II	**Searching and Sorting:** Searching in list, Linear Search, Binary Search, Analysis of Binary Search, Sorting, Merge sort, Quick Sort, Heap Sort.	**12**
UNIT III	**Graphs:** Graph basic definition, Graph Representation, Adjacency Matrix, Adjacency List, Graph Traversal Algorithm, Depth First Search, Breadth First Search. **Minimum Spanning Tree -** Minimum spanning tree algorithm-Kruskal's Algorithm, PRIM's Algorithm, **Single Shortest Path-** Dijkstra's Algorithm, Bellman ford algorithm	**12**
UNIT IV	**Dynamic Programming:**Memoization, optimal binary search trees, Divide and Conquer. knapsack problem, Traveling salesman problem **Computational Complexity:** NP-hard and NP-complete problems, Cook's theorem (without proof).	**12**
	Practicals:	30
	Total Theory+Practicals:	**45+30=75**

List of Practical

1. Write a program for implementing Merge Sort.
2. Write a program for implementing Quick Sort.
3. Write a program for performing Matrix Multiplication .
4. Write a program for Minimum Spanning Trees using Kruskal's algorithm.
5. Write a program for Minimum Spanning Trees using Prim's algorithm.
6. Write a program for Single Source Shortest Path.
7. Implementation of Stack & Queue.
8. Binary Tree Implementations and traversals.
9. Implementation Breadth First Search.
10. Implementation of Depth First Search.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	45

Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition

1	Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein. Introduction to Algorithms, PHI Learning, 2009. ISBN:978-81-203-4007-7.	Thomas H. Cormen	Third
2	Lipschutz Schaum's "Data Structure" Outline Series [TMH]. ISBN-0-07-060168-2	Lipschutz Schaum	Sixth

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610141120
Course Title	Introduction to Linear Algebra
Semester	IV
Type of Course	Discipline Specific Core (DSC)
Credits	3+1+0
Course Prerequisites	--
Course Objective(s)	This course introduces the fundamental concepts of linear algebra including matrices, systems of linear equations, vector spaces, and eigenvalues. It aims to develop computational skills and theoretical understanding necessary for applications in computer science, data science, and machine learning.
Course Outcome (CO)	After completing this course, the students will be able to : CO1: Understand and apply matrix operations and solve systems of linear equations. CO2: Analyze the properties of vectors and vector spaces with a focus on linear independence, basis, and dimension. CO3: Compute eigenvalues and eigenvectors and understand their significance in linear transformations. CO4: Apply linear algebra concepts to solve problems in computer science and real-world contexts.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	1	2	2	2	1	2	2	2	2	3	3	1
CO2	2	2	2	2	1	2	3	2	2	1	3	2
CO3	1	3	2	1	2	3	1	3	3	1	2	2
CO4	3	3	3	3	2	2	2	2	2	3	2	3
Avg.	1.75	2.5	2.25	2	1.5	2.25	2	2.25	2.25	2	2.5	2
1: Weak Correlation, 2: Moderate Correlation, 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	**Matrices and Linear Systems:** Definition and types of matrices, Matrix operations: addition, scalar multiplication, matrix multiplication, Transpose, symmetric and skew-symmetric matrices, Inverse of a matrix (using elementary row operations and adjoint method), Rank of a matrix and echelon forms, Solving systems of linear equations using Gaussian elimination and matrix inversion	**10**
UNIT II	**Determinants and Their Applications:** Definition and properties of determinants, Minors and cofactors, Laplace expansion, Applications: Cramer's Rule for solving linear systems, Determinant as area/volume, Consistency and inconsistency of systems	**11**
UNIT III	**Vectors and Vector Spaces:** Introduction to vectors in $\mathbb{R}^2$ and $\mathbb{R}^3$, Vector operations: addition, scalar multiplication, dot product, cross product, Linear combination, linear dependence and independence, Vector spaces and subspaces, Basis and dimension of a vector space, Row space, column space, and null space	**10**
UNIT IV	**Linear Transformations and Eigen Concepts:** Linear transformations: definition and examples, Matrix representation of linear transformations, Kernel and image of a transformation, Eigenvalues and eigenvectors. Diagonalization of matrices, Applications in computer science (e.g., data compression, principal component analysis – overview only)	**14**
	Total Theory:	**45**

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	15
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	60

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Introduction to Linear Algebra	Gilbert Strang
2	Linear Algebra and Its Applications	David C. Lay

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610142041
Course Title	Fundamentals of Cyber Security
Semester	IV
Type of Course	Discipline Specific Elective (DSE)
Credits	3+0+1
Course Prerequisites	Basic knowledge of computers, networking, and cyber threats enhances cybersecurity learning.
Course Objective(s)	1. Identify and analyze various cyber threats, vulnerabilities, and attack techniques. 2. Understand and implement network security measures, encryption, and firewalls. 3. Learn about SEIM tools and security principles 4. Learn cryptography, authentication, and access control for data protection.
Course Outcome (CO)	The students will be able to: CO1. Apply cybersecurity principles to protect networks, systems, and data. CO2. Identify and mitigate cyber threats, vulnerabilities, and attacks. CO3. Implement encryption, authentication, and access control techniques. CO4. Understand cyber laws, compliance, and ethical practices in cybersecurity.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	1	2	2	2	1	2	2	2	2	3	3	1
CO2	2	2	2	2	1	2	3	2	2	1	3	2
CO3	1	3	2	1	2	3	1	3	3	1	2	2
CO4	3	3	3	3	2	2	2	2	2	3	2	3
Avg.	1.75	2.5	2.25	2	1.5	2.25	2	2.25	2.25	2	2.5	2
1: Weak Correlation, 2: Moderate Correlation, 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	**Cyber Security** – Introduction to Cyber Security, Importance and value of cyber security, – CIA Triad, Common cybersecurity terminology, Past cybersecurity attacks, Common attacks and their effectiveness, Introduction to the eight CISSP security domains, type of attack, Understand attackers. **Frameworks and Controls–** Introduction, Ethics in Cyber security	**12**
UNIT II	**Common Cyber Security Tools-** SIEM,Packet Sniffer, Playbook **Manage Security Risks-** Security Domains, Risk Management Framework, OWASP principles and security audits **Attacks :** Cyber Attacks and types, Incident Response **Assets, Threats, and Vulnerabilities:** Introduction, Digital and Physical Assets, Risk and asset security-NIST cybersecurity framework,Data Lifecycle, PII-Importance of PII.	**12**
UNIT III	**Encryption Methods:** Fundamentals of Cryptography, Public and Private key Cryptography, Symmetric and Asymmetric encryption, **Authentication, authorization and accounting -** Access controls and authentication systems, SSO and MFA, The mechanisms of authorization, Identity and access management, Identify System Vulnerabilities	**12**
UNIT IV	**Threats to Asset Security**-Social Engineering, Malware, Web based Exploits. **Intrusion Detection and Prevention:** Host -Based Intrusion Detection – Network -Based Intrusion Detection – Distributed or Hybrid, Intrusion Detection **Firewalls**: Need for Firewalls – Firewall, Characteristics – Types of Firewalls	**9**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practical

1. Setting, configuring and managing three password policies in the computer (BIOS,Administrator and Standard User).
2. Setting and configuring two factor authentication in the Mobile phone.
3. Security patch management and updates in Computer and Mobiles.
4. Managing Application permissions in Mobile phones.

5. Installation and configuration of computer Anti-virus.
6. Installation and configuration of Computer Host Firewall.
7. Wi-Fi security management in computer and mobile.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition
1	"Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives", Wiley India Pvt. Ltd.	Sunit Belapure and Nina Godbole	Fifth
2	"Computer Security: Principles and Practice",Prentice Hall, 2008.	William Stallings and Lawrie Brown	2008

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610142051
Course Title	Network Programming
Semester	IV
Type of Course	Discipline Specific Elective (DSE)
Credits	3+0+1
Course Prerequisites	–
Course Objective(s)	The objective of this course is to help students understand the foundational concepts of network communication and socket programming, and enable them to design and implement simple network-based applications using TCP and UDP protocols.
Course Outcome (CO)	The students will be able to: CO1. Describe the basic concepts of computer networks, protocols, addressing schemes, and layered models (OSI and TCP/IP). CO2. Understand the socket programming interface and explain how client-server communication works using TCP and UDP. CO3. Apply socket programming techniques to develop basic client-server applications such as echo server, chat app, and file transfer. CO4. Create and implement end-to-end network applications with error handling, multi-client support, and basic security considerations.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	1	2	1	3	2	3	2	3	2	2	2
CO2	2	3	3	1	3	3	1	3	2	3	1	1
CO3	3	1	2	1	2	3	2	1	3	1	3	2
CO4	3	3	3	2	1	3	1	3	1	2	2	3
Avg.	2.5	2	2.5	1.25	2.25	2.75	1.75	2.25	2.25	2	2	2
1: Weak Correlation, 2: Moderate Correlation, 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Basics of computer networks; Network types: LAN, MAN, WAN; Topologies and network devices; Protocols and standards; OSI and TCP/IP reference models; IP addressing, DNS, and port numbers; Protocols: HTTP, FTP, SMTP, UDP, TCP; Introduction to client-server architecture; Role of network programming.	**12**
UNIT II	**Socket Programming Fundamentals:** Introduction to sockets; Types of sockets: Stream (TCP) and Datagram (UDP); Socket system calls: socket(), bind(), listen(), accept(), connect(), send(), recv(), sendto(), recvfrom(); Address structures; TCP/IP socket programming basics; Error handling in sockets; Network byte order; Blocking and non-blocking sockets; Role of threads and concurrency.	**12**
UNIT III	**TCP and UDP Communication:** Building a TCP client-server model; Single client vs multi-client handling; Use of threading or forking; TCP data transmission and acknowledgment; UDP socket communication; Characteristics and limitations of UDP; Broadcasting and multicasting concepts; Real-time applications using TCP and UDP.	**11**
UNIT IV	**Network Applications and Use Cases:** Developing simple network applications: Echo server, chat application, file transfer, and remote command execution; Web communication: creating an HTTP client; Introduction to DNS resolution through code; Basics of network security considerations in socket communication; Future scope: RESTful APIs, WebSockets, and IoT network programming overview.	**10**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

LIST OF PRACTICALS

1. Write a TCP Client and Server to exchange a "Hello" message.
2. Create a TCP Echo Server that returns whatever message the client sends.
3. Build a TCP Client-Server program to send and receive a short text file.
4. Write a UDP Client and Server to send a small message (e.g., "Good Morning").
5. Modify the TCP server to handle two clients (multi-client handling using basic threading).
6. Develop a simple TCP-based Chat Program (text only).
7. Create a TCP program where the client sends its name and gets a greeting from the server.
8. Perform a DNS lookup using code (get IP from domain name like google.com).
9. Build a TCP Client to send a number and receive its square from the server.

10. Create a simple HTTP Client to fetch and display the homepage of a website (e.g., example.com).

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition
1	Unix Network Programming – Volume 1: The Sockets Networking API	W. Richard Stevens	3rd, 2021
2	Computer Networking: A Top-Down Approach	James F. Kurose, Keith W. Ross	7th, 2016

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	0610142061
Course Title	Data Visualization
Semester	IV
Type of Course	Discipline Specific Elective (DSE)
Credits	3+0+1
Course Prerequisites	–
Course Objective(s)	To equip students with the skills to represent, interpret, and communicate data effectively through visual means using modern tools and techniques for informed decision-making.
Course Outcome (CO)	The students will be able to: CO1. Understand the fundamental concepts, purposes, and ethics of data visualization and identify different types of data and visualization techniques. CO2. Prepare and clean raw data using basic data wrangling techniques and Python libraries such as Pandas and NumPy for visualization purposes. CO3. Create a wide range of static and interactive visualizations using Matplotlib, Seaborn, and Plotly based on the type and structure of data. CO4. Design and develop interactive dashboards and data stories using tools like Tableau or Power BI, adhering to best practices in visual communication.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	1	2	1	3	2	3	2	3	2	2	2
CO2	2	3	3	1	3	3	1	3	2	3	1	1
CO3	3	1	2	1	2	3	2	1	3	1	3	2
CO4	3	3	3	2	1	3	1	3	1	2	2	3
Avg.	2.5	2	2.5	1.25	2.25	2.75	1.75	2.25	2.25	2	2	2
1: Weak Correlation, 2: Moderate Correlation, 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	**Introduction to Data Visualization:** Data Visualization, Why Data Visualization, Types of Data, Levels of Measurement, Elements of Good Visualization, Common Chart Types, Ethics of Visualization, Applications	**9**
UNIT II	**Data Preparation and Cleaning:** Data Sources and File Formats: CSV, Excel, JSON, Introduction to Python Libraries: Pandas and NumPy, Handling Missing Values and Duplicates, Outlier Detection and Correction, Data Transformation: Filtering, Sorting, Grouping, Aggregation, Summary Statistics: Mean, Median, Mode, Standard Deviation, Range	**12**
UNIT III	**Visualization Techniques and Tools:** Introduction to Matplotlib and Seaborn, Plotting with Matplotlib: Line, Bar, Pie, Histogram, Scatter, Box plots, Advanced Plots using Seaborn: Heatmaps, Pairplots, Violin plots, Jointplots, Styling Visuals: Titles, Axis Labels, Legends, Color Palettes, Interactive Visualizations with Plotly, Static vs. Interactive Visualizations	**12**
UNIT IV	**Visualization Platforms:** Concept of Data Storytelling and Narrative Visualization, Designing Dashboard Layouts, Creating Dashboards in Tableau and Power BI, Filters, Parameters, and User Interactions in Dashboards, Case Studies in Business, Health, Education, and Social Media	**12**
	Practicals:	**30**
	Total Theory+Practicals:	**45+30=75**

List of Practicals

1. Environment Setup
   Install Python, Jupyter Notebook, and libraries (Pandas, Matplotlib, Seaborn).
   Introduction to notebook interface and basic operations.
2. Basic Plotting with Matplotlib
   Create line plots, bar charts, pie charts, and histograms.
   Add titles, labels, and legends.
3. Seaborn for Statistical Plots
   Create scatter plots, box plots, and violin plots.
   Introduction to sns.load_dataset() and basic styling.
4. Data Cleaning using Pandas
   Load datasets, detect and handle missing data (isnull(), dropna()), and remove

duplicates.

5. Working with Real-World Datasets
   Load and explore CSV/Excel files.
   Use .describe(), .info(), and sorting/grouping methods.
6. Advanced Plotting with Seaborn
   Customize plots with color palettes, themes, size scaling.
   Use FacetGrid, pairplot, and conditional plotting.
7. Multivariate Data Visualization
   Create and interpret heatmaps and correlation plots (.corr(), sns.heatmap()).
8. Interactive Visualizations with Plotly
   Use Plotly Express to create interactive line, bar, and scatter plots with tooltips and sliders.
9. Introduction to Tableau or Power BI
   Install and navigate the interface.
   Load sample data, create simple visuals (bar, pie, map).
10. Dashboard Building
    Create a multi-sheet interactive dashboard with filters, slicers, and visual KPIs.
    Add titles, text boxes, and interactivity.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	**Contact Hours**
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	**Summative**
Multiple Choice Questions (MCQ)	
Viva-voce	

Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)	Edition
1	Unix Network Programming – Volume 1: The Sockets Networking API	W. Richard Stevens	3rd, 2021
2	Computer Networking: A Top-Down Approach	James F. Kurose, Keith W. Ross	7th, 2016

.