



Maharashtra Education Society's

Abasaheb Garware College, Pune.

(Autonomous)

(Affiliated to Savitribai Phule Pune University)

Three Years B.C.A. Degree Program in Computer Application

Faculty of Science and Technology)

Syllabi under NEP

S.Y.B.C.A. (Science)

Choice Based Credit System Syllabus

To be implemented from Academic Year 2025-2026

NEP 2.0
Course Structure of B.C.A. (Science): 2025-26
Second Year

Year	Semester	Course Type	Course Code	Course Title	Credit	No. of hours	
2S	III	Major	CA-201-MJ	Data Structures using C	2	30	
			CA-202-MJ	Relational Database Management Systems	2	30	
			CA-205-MJP	Practical Based on Data Structures and RDBMS	2	4 per batch	
		Minor	ELC-241-MN	Microcontroller Architecture and Interfacing (8051)	2	30	
			ELC-242-MNP	Practical Based on Microcontroller Architecture and Interfacing (8051)	2	4 per batch	
			MTC-241-MN	Discrete Mathematics	2	30	
			MTC-242-MNP	Practical Based on Discrete Mathematics	2	4 per batch	
			VSC	CA-222-VSC	Python Programming	1 + 1	15 +Practical
			FP	CA-232-FP	Project on Web designing		
		IKS	CA-200-IKS	Computing Science in Ancient India	2	30	
	IV	Major	CA-252-MJ	Web Technology – I	2	30	
			CA-255-MJ	Software Engineering	2	30	
			CA-256-MJP	Practical Based on Web Technology I	2	4 per batch	
		Minor	ELC-291-MN	Wireless communication and Introduction to Cloud Technology	2	30	
			ELC-292-MNP	Practical Based on Wireless communication and Introduction to Cloud Technology	2	4 per batch	
			MTC-291-MN	Linear Algebra and Optimization Techniques	2	30	
			MTC-292-MNP	Practical based on Linear Algebra and Optimization Techniques	2	4 per batch	
		VSC	CA-272-VSC	Computer Networks	2	30	
		CEP	CA-281-CEP	Project on Software Engineering			
		SEC	SEC-252-CA	Advanced Excel	1 + 1	15 +Practical	

CA-201-MJ Data Structures using C
Lectures: 30 (Credits-2)

Prerequisites:

- Basic knowledge of algorithms and problem solving
- Knowledge of C Programming Language

Learning Objectives:

1. To understand analysis of algorithms.
2. To learn different applications of array.
3. To understand different types of linked list.
4. To learn use of linear data structures like stack and queue.
5. To acquire knowledge of nonlinear data structures, tree and graph.

Course Outcomes: On completion of this course, students will be able to:

- CO1: Analyze the algorithms on the scale of their performance.
- CO2: Develop searching and sorting techniques to solve real-world computing problems.
- CO3: Apply linked list data structure for developing applications.
- CO4: Understand applications of linear data structures
- CO5: Understand tree and graph operations with their applications.

Unit 1: Introduction to Data Structures and Algorithm Analysis **02(CO1)**

- 1.1 Data object, Data Structure, Abstract Data Type (ADT)
- 1.2 Types of Data Structures
- 1.3 Algorithm analysis - Space complexity, time complexity

Unit 2: Array **03(CO2)**

- 2.1 ADT – Array
- 2.2 Applications of Arrays
 - 2.2.1 Searching - Binary Search
 - 2.2.2 Sorting - Insertion Sort, Merge Sort, Quick Sort

Unit 3: Linked List **08(CO3, CO4)**

- 3.1 Introduction
- 3.2 Types of Linked List – Singly, Doubly, Circular linked list
- 3.3 Operations on Linked List - create, traverse, insert, delete, search
- 3.4 Applications of Linked List – Polynomial representation

Unit 4: Stack **05(CO4)**

- 4.1 Introduction
- 4.2 Implementation of Stack - Static and Dynamic
- 4.3 Operations on Stack – init(), push(), pop(), isEmpty(), isFull(), peek()
- 4.4 Expression types - infix, prefix and postfix expression, evaluation of postfix expression
- 4.5 Applications

Unit 5: Queue**04(C04)**

- 5.1 Introduction
- 5.2 Types of Queues
- 5.3 Static implementation
- 5.4 Operations on Queue - init(), enqueue(), dequeue(), isEmpty(), isFull()
- 5.5 Applications

Unit 6: Non-linear data Structures**08(C05)**

- 6.1 Introduction of Non-linear data Structures - Tree and Graph
- 6.2 Concept and types of Binary trees - skewed tree, strictly binary tree, full binary tree, complete binary tree, expression tree, binary search tree, Heap
- 6.3 Traversal methods of Binary tree – preorder, inorder, postorder
- 6.4 Concept of graph and terminologies
- 6.5 Representation of Graph – Adjacency matrix, Adjacency list
- 6.6 Graph Traversals – Breadth First Search and Depth First Search
- 6.7 Applications – Tree, Graph

Reference Books:

1. Classic Data Structures - D. Samanta, Prentice Hall India Pvt. Ltd.
2. Fundamentals of Data Structures in C - Ellis Horowitz, Sartaj Sahni, Susan Anderson Freed, 2nd Edition, Universities Press.
3. Data Structures using C and C++ - Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum, Pearson Education
4. Data Structures: A Pseudo code approach with C, Richard Gilberg, Behrouz A. Forouzan, Cengage Learning.

CA-202 MJ: Relational Database Management Systems

Lectures: 30 (Credits-2)

Prerequisites:

- Basic Knowledge of RDBMS
- Knowledge of SQL Queries
- Basics of relational database design

Learning Objectives:

1. To understand the concept of database designing
2. To learn the procedural extension of SQL
3. To be familiar with the basic issues of transaction processing
4. To provide knowledge of concurrency control in transactions.
5. To provide understanding of System crash and recovery from crash

Course Outcomes: On completion of this course, students will be able to:

- C01: Design a relational database in desired normal form
- C02: Use database techniques and procedural extensions using PL/SQL
- C03: Understand transaction Management concepts
- C04: Use concepts for concurrency control in transactions
- C05: Use different algorithms for recovery from system crash.

Unit 1: Relational Database Design

04 (C01)

- 1.1 Introduction to Relational-Database Design (undesirable properties of a RDB design)
- 1.2 Concept of Decomposition Desirable Properties of Decomposition (Lossless join, Dependency Preservation)
- 1.3 Functional Dependencies – Basic concept, closure of set of Functional Dependencies, Closure of Attribute set
- 1.4 Concept of normalization, Normal Forms (Definition only) (1NF,2NF and 3NF, BCNF), Examples

Unit 2: Relational Database Design Using PLSQL

10(C02)

- 2.1 Introduction to PLSQL
- 2.2 PL/PgSQL: Datatypes, Language structure
- 2.3 Controlling the program flow, conditional statements, loops
- 2.4 Stored Functions
- 2.5 Handling Errors and Exceptions
- 2.6 Cursors
- 2.7 Triggers
- 2.8 Views

Unit 3: Transaction Concepts and concurrency control **04(C03)**

- 3.1 Describe a transaction, properties of transaction, state of the transaction
- 3.2 Executing transactions concurrently, Anomalies due to Interleaved Execution.
- 3.3 Schedules, types of schedules
- 3.4 Serializability
 - 3.4.1 Precedence graph for Serializability.

Unit 4: Concurrency Control **06(C04)**

- 4.1 Ensuring Serializability by locks, different lock modes,
- 4.2 Two Phase Locking (2PL) and its variations.
- 4.3 Deadlock, Dealing with Deadlock
 - 4.3.1 Deadlock Prevention (wait-die, wound-wait)
 - 4.3.2 Deadlock Detection and Recovery (Wait for graph).
- 4.4 Concurrency control without Locking
 - 4.4.1 Optimistic Concurrency Control
 - 4.4.2 Timestamp Based Concurrency, Thomas Write Rule.
 - 4.4.3 Multiversion Concurrency Control

Unit 5: Crash Recovery **06(C05)**

- 5.1 Failure classification
- 5.2 Buffer Management
 - 5.2.1 Write-Ahead Logging
 - 5.2.2 Stealing Frames and Forcing Pages
- 5.3 Checkpoints
- 5.4 Log based recovery techniques (Deferred and immediate update)
- 5.5 Database backup and recovery from catastrophic failure

Reference Books:

1. Database Management Systems by Raghu Ramakrishnan, Mcgraw-hill higher Education publication
2. Database System Concepts by Henry F. Korth, Abraham Silberschatz, Tata McGraw-Hill Education publication
3. Fundamentals of Database System - by Elmasri Ramez and Navathe Shamkant
4. Practical Postgresql By Joshua D. Drake, John C Worsley O'Reilly publication

CA-205-MJP: Practical Based on Data Structures and RDBMS

No. of Sessions – 12 (Credits – 2)

Assignments of Data Structures using C:

1. Array
 - a. Searching (Binary Search)
 - b. Sorting (Insertion sort, Merge sort)
2. Linked List
 - a. Singly linked list operations
 - b. Doubly linked list operations
 - c. Circular linked list operations
 - d. Polynomial addition
3. Stack
 - a. Stack operations
 - b. Stack applications
4. Queue
 - a. Queue operations
 - b. Queue applications
5. Non linear data structures
 - a. BST creation, traversal techniques
 - b. Graph representation methods

Assignments of Relational Database Management System:

1. Stored Function
 - a. A Simple Stored Function
 - b. A Stored Function that returns a value
 - c. A Stored Function recursive
2. Cursors
 - a. A Simple Cursor
 - b. A Parameterize Cursor
3. Exception Handling
 - a. Simple Exception- Raise Debug Level Messages
 - b. Simple Exception- Raise Notice Level Messages
 - c. Simple Exception- Raise Exception Level Messages
4. Triggers
 - a. Before Triggers (insert, update, delete) (row level and statement level)
 - b. After Triggers (insert, update, delete) (row level and statement level)
5. Views
 - a. Creating Views on tables, and writing queries on the views.

ELC-241-MN: Microcontroller Architecture & Interfacing (8051)

Lectures: 30 (Credits-2)

Prerequisites:

None

Learning Objectives:

1. To study the basics of microcontroller IC 8051.
2. To study the Programming of IC 8051.
3. To study the interfacing techniques of microcontroller IC 8051.
4. To design different application circuits using microcontroller IC 8051.

Course Outcomes: On completion of this course, students will be able to:

- CO1: Develop programs for 8051 microcontroller.
- CO2: Interface I/O peripherals to 8051 microcontroller.
- CO3: Design small microcontroller based projects.
- CO4: Demonstrate the ability to interface and control various hardware components

Unit 1: Basics of Microcontroller & Intel 8051 architecture 08(C01)

- 1.1 Introduction to microcontrollers
- 1.2 Difference between controller and processor
- 1.3 Architecture of 8051
- 1.4 Internal block diagram
- 1.5 Internal RAM organization
- 1.6 SFRS
- 1.7 Pin functions of 8051
- 1.8 I/O port structure & Operation
- 1.9 External Memory Interface

Unit 2: Programming model of 8051 10(C02)

- 2.1 Instruction classification
- 2.2 Instruction set
- 2.3 Addressing Modes - Immediate, register, direct, indirect and relative
- 2.4 Assembler directives (ORG, END)
- 2.5 I/O Bit & Byte programming using assembly language for LED and seven segment display (SSD) interfacing
- 2.6 Introduction to 8051 programming in C

Unit 3: Timer / Counter, Interrupts 07(C03)

- 3.1 Timer / Counter: TMOD, TCON, SCON, SBUF, PCON Registers
- 3.2 Timer modes
- 3.3 Programming for time delay using mode 1 and mode 2
- 3.4 Interrupts
 - 3.4.1 Introduction to interrupt
 - 3.4.2 Interrupt types and their vector addresses
 - 3.4.3 Interrupt enable register and interrupt priority register (IE, IP)

Unit 4: Interfacing, Serial Communication 05 (C04)

- 4.1 Interfacing - DAC, LED, stepper motor, DC Motor, SSD

Reference Books:

1. 8051 microcontroller and Embedded system using assembly and C: Mazidi and McKinley, Pearson publications.
2. The 8051 microcontroller – Architecture, programming and applications: K. Uma Rao and Andhe Pallavi, Pearson publications.
3. Introduction to microcontrollers: Meghana Tikhe Palkar, Nirali Publication.

ELC-242-MNP: Practical based on Microcontroller Architecture & Interfacing (8051)

No of sessions: 12 (2 Credits)

Assignments:

1. Arithmetic programmes using 8051: addition, subtraction, to find largest and smallest number in the given string, to find largest from a string, to find smallest from a string.
2. Interfacing of LED logic
3. Interfacing of TWS
4. Interfacing of LCD (8-bit mode)
5. Interfacing of LCD + keyboard (8-bit mode)
6. Interfacing of 7-segment
7. Interfacing of 7-segment+keyboard
8. Interfacing of DAC 0808
9. Interfacing of DC motor
10. Interfacing of stepper motor
11. Interfacing of relay & buzzer
12. Project equivalent to two experiments

MTC-241-MN: Discrete Mathematics

Lectures: 30 (Credits-2)

Prerequisites:

None

Learning Objectives:

1. To understand the logic behind the validation of statements in programming.
2. To get acquainted with basics of relations and using relation in various cases.
3. To study lattices and relation between binary lattices and digital circuits.
4. To formulate and solve recurrence relations for the given case.

Course Outcomes: On completion of this course, students will be able to:

- CO1: Understand the logic behind the validation of statements in programming.
CO2: Acquainted with basics of relations, types of relations. Students will study implication of transitivity in information flow.
CO3: Study lattices and usage of binary lattices in constructing digital circuit.
CO4: Analyse algorithms in particular using recurrence relations.

Unit 1: Logic

10 (CO1)

- 1.1 Propositional Logic, Propositional Equivalences
- 1.2 Rules of Inference: Argument in propositional Logic, Validity Argument (Direct and Indirect methods), Rules of Inference for Propositional Logic, Building Arguments
- 1.3 Introduction to Predicates and Quantifiers: Predicate, n-Place Predicate or n-ary Predicate, Quantification and Quantifiers, Universal Quantifier, Existential Quantifier, Quantifiers with restricted domains, Logical Equivalences involving Quantifiers

Unit 2: Relations

06(CO2)

- 2.1 Relations, types of relations, equivalence relations
- 2.2 Digraphs of relations, matrix representation and composition of relations
- 2.3 Partial ordering relations, Poset and Hasse diagram
- 2.4 Transitive closure and Warshall's Algorithm

Unit 3: Lattice

04(CO4)

- 3.1 Poset, Hasse diagram
- 3.2 Lattices, Complemented lattice, Bounded lattice and Distributive lattice

Unit 4: Recurrence Relations

10 (CO5)

- 4.1 Recurrence Relations: Introduction, Formation
- 4.2 Linear Recurrence Relations with constant coefficients
- 4.3 Solving Homogeneous Recurrence relations
- 4.4 Non-homogeneous Recurrence relations
- 4.5 Master theorem (Only statement, without proof), Solving recurrence relations using master theorem

Text Books:

1. Discrete Mathematics and its applications, by Kenneth Rosen, Tata McGraw Hill, Seventh Edition.
2. Discrete Mathematical Structures, by Kolman, Busby, Ross, Rehman, Prentice Hall.
3. Elements of Discrete Mathematics, by C. L. Liu, Tata McGraw Hill.

MTC-242-MNP: Practical based on Discrete Mathematics

No. of Sessions – 12 (Credits – 2)

Assignments based on following topics -

1. Problems based on Propositional logic
2. Predicate logic
3. Problems based on equivalence relations, digraph and matrix representation of a relation
4. Problems based on partial order, Hasse digram, Warshall's algorithm
5. Problems based on lattices, distributive lattices, complemented lattices.
6. Problems based on formulation of recurrence relations
7. Problems based on solving homogeneous recurrence relations
8. Problems based on solving recurrence relations using Master theorem

CA-222-VSC: Python Programming

Lectures: 15+Practical (Credits-2)

Prerequisites: None

Learning Objectives:

1. To introduce programming concepts using python.
2. To use various data structures and control structures python.
3. To learn use of built-in functions, modules and develop user defined functions.

Course Outcomes: On completion of this course, students will be able to:

- CO1: Understand basic constructs of python programming.
- CO2: Determine the methods to create and develop Python programs by utilizing the data structures like lists, dictionaries, tuples and sets and control structures.
- CO3: Use built-in functions, modules and develop user defined functions for problem solving.

Unit 1: Introduction to Python

02(C01)

- 1.1 Need, Applications, program structure, interactive mode and script mode
- 1.2 Python Fundamentals – Keywords, Operators, Variables, Indentation, Comments, Input and Output in Python

Unit 2: Data Handling and Control Statements

09(C02)

- 2.1 Data Types – int, float, complex, boolean, none, string
- 2.2 String - declaration, manipulation, special operations, Strings, built-in String methods.
- 2.3 List - Concept, creating and accessing elements, updating & deleting elements of list, traversing a list, reverse, built-in List Operators - Concatenation, Repetition, In, Built-in List functions and methods.
- 2.4 Tuple - Accessing values in Tuples, Tuple Assignment, Tuples as return values, Variable- length argument tuples, Basic tuple operations - Concatenation, Repetition, in, Iteration, Built-in tuple functions, indexing, slicing and matrices.
- 2.5 Set - Concept, creating and accessing elements, Operations on Sets
- 2.6 Dictionary - Creating a Dictionary, Accessing Values in a dictionary, Updating Dictionary, Deleting Elements from Dictionary, Properties of Dictionary keys, Operations in Dictionary, Built-In Dictionary Functions, Built-in Dictionary Methods.
- 2.7 Conditional Statements - if, if-else, nested if-else
- 2.8 Loops - for, while, nested loops, loop control statements (break, continue, pass)

Unit 3: Functions and Modules**04(C03)**

- 3.1 Definitions and Uses, User defined functions, Function Calls
- 3.2 Types of functions, Parameters, Return values
- 3.3 Lambada functions
- 3.4 Recursion
- 3.5 Modules - Importing module, Creating & exploring modules, Math module, Pandas, Numpy

Reference Books:

- 1. An Introduction to Computer Science using Python 3 by Jason Montoyo, Jennifer Campbell, Paul Gries, The pragmatic bookshelf-2013
- 2. James Payne, "Beginning Python": Using Python and Python 3.1, Wrox Publication
- 3. Introduction to Computer Science Using Python- Charles Dierbach, Wiley Publication Learning with Python ", Green Tea Press, 2002
- 4. Introduction to Problem Solving with Python by E balguruswamy, TMH publication- 2016
- 5. Beginning Programming with Python for Dummies Paperback – 2015 by John PaulMueller

Practical Assignments of Python programming

- 1. Assignment on data types, operators
- 2. Assignment on control statements
- 3. Assignment on string, list, set, tuple, dictionary
- 4. Assignment on function
- 5. Assignment on modules

CA-232-FP Project on Web Designing

No. of Sessions – 12 (Credits – 2)

Prerequisites:

- HTML, CSS, JavaScript, jQuery

Learning Objectives:

1. To apply knowledge of HTML, CSS, JavaScript, and jQuery in real-world web development.
2. To build responsive and interactive web interfaces.
3. To develop complete front-end projects using modern web technologies.
4. To learn basic documentation and presentation of software projects.

Course Outcomes: On completion of this course, students will be able to:

- CO1: Develop a functional, structured, and styled multi-page website.
- CO2: Implement interactivity and user interface behavior using JavaScript and jQuery.
- CO3: Apply responsive design using CSS and Bootstrap.
- CO4: Create a basic project report and demonstrate the project effectively.

Project Implementation Guidelines:

1. Students shall choose a relevant web-based project topic in consultation with the subject teacher.
2. The project can be done individually or in pairs.
3. The project work is to be carried out throughout the semester (12 sessions).
4. The focus should be on frontend development only using:
 - HTML5 for structure
 - CSS3 and Bootstrap for styling
 - JavaScript and jQuery for interactivity
5. Each project must have a minimum of 4–5 linked webpages.
6. HTML Forms with validations (via JavaScript) are compulsory.
7. Must include at least:
 - 3+ JavaScript functions
 - 1+ jQuery DOM manipulation or animation
 - 1+ Bootstrap component (e.g., navbar, button, card, grid)
8. Students must demonstrate creativity, clean layout, and user-friendly navigation.
9. A project report must be prepared with the following structure:
 - Title Page
 - Certificate
 - Index
 - Introduction
 - Objective
 - Tools/Technologies Used
 - Screenshots of UI
 - Code Snippets/Features Summary
 - References

10. The report should be spiral bound and properly formatted, and submitted in.
11. The project and report must be certified by the subject teacher and HOD.

Evaluation Scheme:

Description	Evaluation	Marks
Regular Progress, Attendance	Internal	10
Design/Layout & Technology Usage		10
JS/jQuery Implementation + Viva		5
Project Presentation & Report	External	10
Demonstration of the Project		10
Viva		05
Total		50

CS-200-IKS: Computing Science in Ancient India

Lectures: 30 (Credits-2)

Prerequisites: None

Learning Objective:

1. To introduce Vedic mathematical techniques and their relevance to modern computational methods.
2. To understand Nyaya's logical framework and its application in reasoning and AI.
3. To explore the algorithmic structure of Panini's grammar and Chandasastra's binary system in computational linguistics and mathematics.
4. To explore real-world applications of IKS concepts in computational sciences.

Course Outcomes: On completion of this course, students will be able to:

- CO1: Understand the computational foundations of Indian Knowledge Systems by applying Vedic mathematical techniques in problem-solving.
- CO2: Understand Chandasastra's binary system in computational linguistics and mathematics.
- CO3: Recognize the applications of IKS in modern computing fields.
- CO4: Use Nyaya's logical reasoning in AI and decision-making.

Unit 1: Vedic Mathematics & Computational Thinking

08 (C01)

- 1.1. Ganit – Gan – to compute
- 1.2. Need of algorithms/ sophisticated mathematics algorithms in ancient India
- 1.3. Introduction to Vedic Mathematics: Origins and importance in ancient India, Sutras and their logical foundation
- 1.4. Basic Arithmetic using Vedic Methods: Addition, subtraction, multiplication, and division tricks
- 1.5. Algebraic Applications of Vedic Mathematics: Squaring, square roots, cube roots

Unit 2: Panini's Astadhyayi & Chandasāstra

08 (C02)

- 2.1. Introduction to Panini's Astadhyayi: Historical background and linguistic importance
- 2.2. Introduction to Chhanda Shastra by Pingala Acharya
- 2.3. Equivalence to Binary System
- 2.4. Some algorithms in Chhanda Shastra–Prastara, Nashtam, Uddhistam, Sankhya, Lagakriya, Meru Prastara (Pascal triangle)

Unit 3: Cryptography in Ancient India

08 (C03)

- 3.1 Introduction to Cryptography
- 3.2 Rhythmic, Dialectical, Aphoristic
- 3.3 Akshara Sankhya system of Aryabhatta
- 3.4 Akshara Sankhya system of Katpayadi – Cryptography and Hashing function

Unit 4: Introduction to Nyaya (Indian Logic)

06 (C04)

- 4.1. Introduction to Nyaya Philosophy: Introduction to Nyaya (Indian Logic), Overview of Indian philosophical schools, Importance of Nyaya in logical reasoning, Types of reasoning (Anumana, Pramana, etc.)
- 4.2. Nyaya's Four Sources of Knowledge (Pramāṇa): Perception, inference, Equivalence, verbal testimony

References:

1. Computing Science in ancient India by T.N. Rao, Subhash Kak, Meru Publication.
2. Vedic Mathematics, Jagadguru Swami Bharati Krishna Tirtha, Motilal Banarsidass Publishing House, New Delhi.
3. "The Power of Vedic Maths" – Atul Gupta, JAICO publishing
4. Nyaya Theory of Knowledge" – S.C. Vidyabhusana
5. "A Primer of Indian Logic" – Kuppaswami Sastri, Hassell Street Press.2021
6. "The Mathematics of Metre" – Satyanarayana Das

CA-252-MJ Web Technology – I

Lectures: 30 (Credits - 2)

Prerequisites:

- Knowledge of HTML, CSS, JavaScript

Learning Objectives:

1. To understand the fundamentals of web technologies and the client-server communication model.
2. To learn the syntax, structure, and fundamental programming concepts of PHP.
3. To develop skills in using functions and string manipulation techniques in PHP.
4. To explore array structures, operations, and their applications in PHP.
5. To understand file handling and directory management in PHP.

Course Outcomes: On completion of the course, the student will be able to

- C01: Describe the key components of web technologies, including HTTP communication and website types.
- C02: Write basic PHP scripts using variables, data types, operators, and control flow statements.
- C03: Implement functions, variable scope, and string manipulation techniques for efficient program development.
- C04: Use arrays for data storage, retrieval, and processing, including sorting and traversal techniques.
- C05: Perform file and directory operations such as reading, writing, renaming, and managing permissions.

Unit 1: Introduction to Web Technologies

02 (C01)

- 1.1 Qualities of a good website
- 1.2 Web Client-Server and its Communication
- 1.3 HTTP request message, HTTP response message
- 1.4 Types of Websites (Static and Dynamic Websites)

Unit 2: Introduction to PHP

02 (C02)

- 2.1 Lexical structure
- 2.2 Data types
- 2.3 Variables, Operators and Flow-Control Statements
- 2.4 Including code – require and include
- 2.5 Comments

Unit 3: Function and String

10 (C03)

- 3.1 Defining and calling a function
- 3.2 Variable Scope – Global variables and Static variables
- 3.3 Function Parameters – Pass by value, Pass by reference, Default parameters, Variable parameters, Missing parameters
- 3.4 Variable function, Anonymous function
- 3.5 Types of strings in PHP
- 3.6 Printing Strings
- 3.7 Comparing strings

- 3.8 Manipulating and searching strings
- 3.9 Regular expressions

Unit 4: Arrays

10 (C04)

- 4.1 Indexed Vs Associative arrays
- 4.2 Identifying elements of an array
- 4.3 Storing data in arrays
- 4.4 Multidimensional arrays
- 4.5 Extracting multiple values
- 4.6 Converting between arrays and variables
- 4.7 Traversing arrays
- 4.8 Sorting
- 4.9 Action on entire arrays
- 4.10 Using arrays – Sets, Stacks, Queue

Unit 5: Files and Directories

06 (C05)

- 5.1 Opening and Closing, Getting information about file, Read/write to file,
- 5.2 Splitting name and path from file, Rename and delete files
- 5.3 Reading and writing characters in file
- 5.4 Reading entire file
- 5.5 Random access to file data
- 5.6 Getting information on file
- 5.7 Ownership and permissions
- 5.8 Working with Directories

Reference Books

1. Programming PHP By Rasmus Lerdorf and Kevin Tatroe, O'Reilly publication
2. Beginning PHP5, Wrox publication
3. PHP for Beginners, SPD publication
4. Mastering PHP BPB Publication
5. PHP cookbook O'Reilly publication

CA-255-MJ: Software Engineering

Lectures: 30 (Credits-2)

Prerequisites: None

Learning Objectives:

1. To enhance the knowledge of methods and processes involved in software design and development with object-oriented concepts.
2. To get introduced to the concepts of agile software development.
3. To get knowledge of Requirements Engineering process.
4. To achieve clarity of System Analysis and Design and use of UML tools.
5. To acquire knowledge of behavioral and Architectural modeling tools

Course Outcomes: On completion of this course, students will be able to:

- C01: Understand different prescriptive and Generic Process models used for Software Engineering.
- C02: Get basic knowledge of Agile Software Development.
- C03: Carry out the process of Requirement Analysis for a software.
- C04: Perform System Analysis and Design using UML tools
- C05: To use different behavioral and Architectural model tools like Class, Activity, Component, Deployment diagrams for System Design

Unit 1: Introduction To Software Engineering and Process Models

05(C01)

- 1.1 Definition of Software
- 1.2 Nature of Software Engineering
- 1.3 Changing nature of software
- 1.4 Software Process
 - 1.4.1 The Process Framework
 - 1.4.2 Umbrella Activities
- 1.5 Generic Process Model
- 1.6 Prescriptive Process Models
 - 1.6.1 The Waterfall Model
 - 1.6.2 Incremental Process Models
 - 1.6.3 Evolutionary Process Models
 - 1.6.4 Concurrent Models
 - 1.6.5 The Unified Process

Unit 2: Agile Development

05 (C02)

- 2.1 What is Agility?
- 2.2 Agile Process
 - 2.2.1 Agility Principles
 - 2.2.2 The Politics Of Agile Development
 - 2.2.3 Human Factors
- 2.3 Extreme Programming (XP)
 - 2.3.1 XP Values
 - 2.3.2 XP Process
 - 2.3.3 Industrial XP

- 2.4 Scrum
- 2.5 Agile Unified Process (AUP)

Unit 3: Requirements Analysis

04 (C03)

- 3.1 Requirements Engineering
- 3.2 Establishing Groundwork
- 3.3 Requirement Elicitation,
- 3.4 Software requirement specification (SRS)
- 3.5 Building the Analysis Model
 - 3.5.1 Elements of the Analysis Model
 - 3.5.2 Analysis Patterns
 - 3.5.3 Agile Requirements Engineering
- 3.6 Negotiating Requirements
- 3.7 Requirements Monitoring
- 3.8 Validating Requirements

Unit 4: Introduction of UML and Basic and Advanced Structural Modeling

06 (C04)

- 4.1 Overview of UML,
- 4.2 Conceptual Model of UML.
- 4.3 Class Diagram, Advanced classes
- 4.4 Advanced Relationship
- 4.5 Interface
- 4.6 Types and Roles
- 4.7 Packages
- 4.8 Object Diagram

Unit 5: Basic Behavioral Modeling

06 (C05)

- 5.1 Use case diagram
- 5.2 Activity diagram
- 5.3 State diagram
- 5.4 Sequence diagram

Unit 6: Architectural Modeling

04 (C05)

- 6.1 Component Diagram
- 6.2 Deployment Diagram

Reference Books:

1. Software Engineering: A Practitioner's Approach- Roger S. Pressman, McGraw hill (Eighth Edition)
2. Grady Booch, James Rumbaugh, The Unified Modeling Language User/Reference Guide, Second Edition, Publisher: Addison Wesley, ISBN-13: 978-0321267979
3. Ivar Jacobson, Object Oriented Software Engineering, First Edition, Addison Wesley, ISBN-13: 978-0201544350

CA-256-MJP Practical Based on Web Technology I

No. of Sessions: 12 (Credits-2)

Assignments based on Web Technology I

1. Assignments on basics of PHP
2. Assignments on function in PHP
3. Assignments on String manipulations
4. Assignments on Arrays
5. Assignments on File and directory handling

ELC-291-MN: Wireless communication and Introduction to Cloud Technology

Lectures: 30 (Credits-2)

Prerequisites:

Computer Networks

Learning Objectives:

1. To understand the **basic elements** of a communication system.
2. To describe the working of a cellular telephony system.
3. To obtain eigenvalues of a matrix using analytical and numerical methods.
4. To use Simplex methods and its variants to solve linear programming problems.

Course Outcomes: On completion of this course, students will be able to:

CO1: Know basics of communication.

CO2: Become familiar with working of wireless technologies such as Mobile communication, GSM, GPRS, 3G and 4G Cellular Network Technologies for Data Connections.

CO3: Understand working principles of short-range communication application.

CO4: Get introduced to Cloud Technology.

Unit 1: Wireless Communication: Cellular Telephony **08(CO1)**

- 1.1 Introduction to Communication: Elements of Communication system, Electromagnetic spectrum
- 1.2 Types of communication: simplex, half duplex, full duplex, baseband and broadband, Serial communication: asynchronous and synchronous
- 1.3 Information Theory: Signal bandwidth and channel bandwidth, rate of information (data rate, baud rate), channel capacity, Nyquist theorem, Signal to noise ratio, Noise Figure, Shannon theorem
- 1.4 Error handling, hamming code

Unit 2: Wireless Communication: cellular technology **08(CO2)**

- 2.1 Introduction of cellular telephony system: Frequency reuse, handoff strategies, Co-channel and adjacent channel interference, block diagram of mobile handset, Overview of Cellular Telephony generations: 1G to 5G, 3G (W-CDMA, UMTS), 4G(LTE)
- 2.2 GSM: architecture, frame structure, mobility management
- 2.3 GPRS: architecture, application

Unit 3: Short Range Wireless Technologies and Location Tracking **08(CO3)**

- 3.1 Bluetooth: Bluetooth architecture, Bluetooth protocol stack, Bluetooth frame Structure, applications of bluetooth
- 3.2 Zigbee: Architecture, topologies, applications of RFID
- 3.3 RFID: working of RFID system, types of RFID tags, applications Location Tracking
- 3.4 GPS system: components of GPS system (space segment, control segment, user segment), Applications of GPS

Unit 4: Introduction to cloud technology **06 (CO4)**

- 4.1 Introduction to Cloud Storage. Service modes of cloud: IaaS, PaaS, SaaS
- 4.2 Deployment models of cloud: Private cloud, public cloud, hybrid cloud and community cloud
- 4.3 Cloud Technology Applications
- 4.4 Introduction to Amazon Web Services for IoT

Reference Books:

1. Wireless Communications Principles and Practice, Rappaport, Pearson Publication
2. Mobile Communications, Jochen Schiller, Pearson publication
3. Internet of Things: Principles and Paradigms, Rajkumar Buyya and Dastjerdi, MK publishers.

ELC-291-MN: Practical based on Wireless communication and Introduction to Cloud Technology

No. of Sessions – 12 (Credits – 2)

Assignments on following topics:

1. Introduction to Python programming
2. Study of GSM system (Message transmission & Reception)
3. To study working of SIM card in GSM handset
4. Study of GPRS system
5. Study of Zig-bee for one application
6. Study of RFID system
7. Study of Error detection and correction by using Hamming Code technique
8. Study of signal strength measurement.
9. Study of Computer network components: Cables, Connectors, Routers, Switches, Ethernet and related interfacing cards

Compulsory Activity equivalent to two practical: (any one)

1. To study Configuration of IP and MAC address and to study Local Area Network setup
2. To study Amazon web services and Skynet and prepare a report based on the features, advantages, limitations and applications.

MTC-291-MN: Linear Algebra and Optimization Techniques

Lectures: 30 (Credits-2)

Prerequisites:

None

Learning Objectives:

1. To understand various concepts of linear algebra and use it as a platform to model physical problems.
2. To appreciate analytical and numerical solutions of linear equations.
3. To obtain eigenvalues of a matrix using analytical and numerical methods.
4. To use Simplex methods and its variants to solve linear programming problems.

Course Outcomes: On completion of this course, students will be able to:

CO1: Understood matrix as a basic tool of linear algebra and can use various matrix transformations.

CO2: Solve system of linear equations using methods – Gauss Elimination method, Gauss Jordan method, Cramer's Rule.

CO3: Solve system of equations using numerical methods.

CO4: Find eigenvalues of a matrix – both analytic and numerical methods.

CO5: Formulate and solve linear programming problems using Simplex method, Big M method.

Unit 1: Matrices, Vectors, Determinants, Linear Systems **10 (CO1, CO2)**

- 1.1 Overview Matrices, Vectors: Addition and Scalar Multiplication, Matrix Multiplication
- 1.2 Linear Systems of Equations, Gauss Elimination
- 1.3 Linear Independence, Rank of a Matrix
- 1.4 Solutions of Linear Systems: Existence, Uniqueness, Gauss Elimination method
- 1.5 Second- and Third-Order Determinants
- 1.6 Determinants, Cramer's Rule
- 1.7 Inverse of a Matrix, Gauss-Jordan method

Unit 2: Numeric Linear Algebra **06(CO3)**

- 2.1 Linear Systems: Gauss Elimination
- 2.2 Linear Systems: Solution by Iteration (Gauss Seidel method and Gauss Jordan method)

Unit 3: Matrix Eigenvalue Problems **04(CO4)**

- 3.1 The Matrix Eigenvalue Problem: Determining Eigenvalues and Eigenvectors
- 3.2 Symmetric, Skew-Symmetric, and Orthogonal Matrices

Unit 4: Linear Programming (LP) Problems **10 (CO5)**

- 4.1 Two-variable LP Model
- 4.2 Properties of Linear Programming Problems (LPP)
- 4.3 Graphical LP Solution
- 4.4 LP in standard equation form
- 4.5 The Simplex Method
- 4.6 Big M method

Text Books:

1. Erwin Kreyszig, Advanced Engineering Mathematics, Wiley India, 10th Edition, 2011.
2. Hamdy A Taha, Operations Research: An Introduction, Pearson/PHI, 10th Edition, 2018.

Reference Books:

1. Bernard Kolman, David R. Hill, Elementary Linear Algebra with Applications, Ninth Edition, 2008.
2. Ravindran, Phillips and Solberg, Operations Research: Principles and Practice, John Wiley, Second Edition, 2000.

**MTC-292-MNP: Practical based on Linear Algebra and Optimization
Techniques**

No. of Sessions – 12 (Credits – 2)

Following Assignments to be done using Python-

1. Introduction to Python (Emphasis on matrices and Linear Algebra)
2. Determinant, Rank, Solving system of linear equations – inbuilt functions
3. Solving system of equations – Gauss elimination method (using row transformations)
4. Matrix eigenvalues problem – inbuilt functions, power method
5. Gauss Seidel method, Gauss Jordan method
6. Solving LPP using Python using PuLP
7. Solving LPP using Scipy

CA-272-VSC: Computer Networks

Lectures: 30 (Credits-2)

Prerequisites: None

Course Objectives:

1. To provide the fundamental concepts of networking standards and technology.
2. To gain the knowledge of OSI/IP, TCP/IP model and addressing techniques.
3. To learn analog signal and digital signal.
4. To study different techniques for framing, error control, flow control and routing.
5. To learn IP address and the network protocols.

Learning Outcomes: On completion of this course, students will be able to:

- C01: Analyze the requirements for a given organization and select appropriate network architecture, topologies, transmission mediums and technologies.
- C02: Understand different network model and addressing techniques.
- C03: Calculate data rate limits for noisy and noiseless channel.
- C04: Understand different techniques for framing, error control, flow control and routing.
- C05: Understand the IPV4 network protocols.

Unit 1: Introduction to Data Communication

04(C01)

- 1.1 Concept, Characteristics and Components of Data communication
- 1.2 Types of Data flow – Simplex, Half Duplex, Full Duplex
- 1.3 Computer Networks applications –Business Application, Home Application, Mobiles
- 1.4 Broadcast and point-to-point networks
- 1.5 Network Topologies - Bus, Star, Ring, Mesh
- 1.6 Network Types- LAN, MAN, WAN, PAN, Wireless Networks

Unit 2: Network Models

06(C02)

- 2.1 OSI Model – layered architecture, peer-to-peer processes
- 2.2 TCP/IP Model – layers and Protocol Suite
- 2.3 Addressing-Physical, Logical, Port addresses, Specific addresses

Unit 3: Physical Layer

08(C03)

- 3.3 Analog and Digital data, Analog and Digital signals, Digital Signals-Bit rate, Bit length
- 3.4 Baseband Transmission, Broadband Transmission
- 3.5 Transmission Impairments– Attenuation, Distortion and Noise
- 3.6 Data Rate Limits– Noiseless channel: Nyquist's bit rate, noisy channel: Shannon's law
- 3.7 Performance of the Network Bandwidth, Throughput, Latency (Delay), Bandwidth – Delay Product, Jitters
- 3.8 Line Coding Characteristics, Line Coding Schemes–Unipolar -NRZ, Polar-NRZ-I, NRZ-L, RZ, Manchester and Differential Manchester

- 3.9 Transmission Modes, Parallel Transmission and Serial Transmission– Asynchronous, Synchronous and Isochronous
- 3.10 Switching-Circuit Switching, Message Switching and Packet Switching

Unit 4: Data Link Layer

08(C04)

- 4.1 Framing – Concept, Methods – Character Count, Flag bytes with Byte Stuffing, Starting and ending Flags with Bit Stuffing
- 4.2 Elementary data link protocols – For Noiseless channel – Simplex, Stop and wait protocol
- 4.3 Sliding Window Protocols – 1-bit sliding, Go-Back N and Selective Repeat
- 4.4 Random Access Protocols – 1-persistent, p-persistent and non-persistent CSMA/CD, CSMA/CA
- 4.5 Controlled Access - Reservation, Polling and Token Passing

Unit 5: Network Layer

04 (C05)

- 5.1 IPv4 addresses: Address space, Notation, Classful addressing, Classless addressing
- 5.2 IPv4: Datagram, Fragmentation, checksum, options
- 5.3 NAT

Reference Books:

- 1. Data Communications and Networking by Behrouz Forouzan, Fifth Edition, ISBN 978-0-07-337622-6 McGraw Hill.
- 2. Computer Networks, ANDREW S. Tanenbaum, Fifth Edition, ISBN-13: 978-0-13-212695-3, Pearson

CA-281-CEP: Project on Software Engineering

No. of Sessions – 12 (Credits – 2)

Assignments of Mini Project using Software Engineering Techniques:

1. Structural Model:
 - Problem definition, scope of proposed system
 - Requirement Specification
 - ER Diagram
 - Use Case Diagram
 - Class Diagrams
2. Behavioral Model:
 - Sequence Diagram
 - Collaboration Diagram
 - Activity Diagram
 - State Chart Diagram
3. Architectural Model:
 - Component Diagram
 - Deployment Diagram
 - Package Diagram

SEC-252-CA: Advanced Excel

Lectures:15+Practical (Credits-2)

Prerequisites: None

Learning Objectives:

1. Acquire knowledge of data validation, conditional formatting, and charting techniques to improve data visualization.
2. Develop advanced Excel skills to enhance efficiency and reduce risk in data management and analysis.

Course Outcomes: On completion of this course, students will be able to:

- CO1: Learn basic functions of Excel.
- CO2: Learn advanced functions of Excel.
- CO3: Learn data handling in Excel.
- CO4: Creation, management, and formatting pivot tables and pivot charts.

Unit 1: Overview of the Basics of Excel

08(CO1)

- 1.1 Customizing common options in Excel
- 1.2 Basic Formulas like SUM, AVERAGE, COUNT, MAX, MIN
- 1.3 Spreadsheet Basics

Unit 2: Working with Functions

06(CO2)

- 2.1 Writing conditional expressions (using IF)
- 2.2 Using logical functions (AND, OR, NOT)
- 2.3 Using lookup and reference functions (VLOOKUP, HLOOKUP, MATCH, INDEX)
- 2.4 Date and Time Functions
- 2.5 Text Functions
- 2.6 Power Functions (Countif, CountIFS, SumIF, SumifS)

Unit3: Data Handling

08(CO3)

- 3.1 Specifying a valid range of values for a cell
- 3.2 Specifying a list of valid values for a cell
- 3.3 Specifying custom validations based on formula for a cell
- 3.4 Sorting Tables
- 3.5 Using custom sorting
- 3.6 Filtering data for selected view (AutoFilter)

3.7 Data Forms

3.8 Adding Data Using the Data Forms

Unit 4: Pivot Tables

08(C04)

4.1 Formatting and customizing Pivot tables

4.2 Using advanced options of Pivot tables

4.3 Pivot Charts

4.4 Consolidating data from multiple sheets and files using Pivot tables

4.5 Using external data sources

4.6 Using data consolidation feature to consolidate data

4.7 Show Value As (% of Row, % of Column, Running Total, Compare with Specific Field)

4.8 Viewing Subtotal under Pivot

4.9 Creating Slicers (Version 2010 & Above)

Reference Books

1. Mastering Advanced Excel, by published by BPB Publications, ISBN NO: 935551865X, 978-9355518651
2. Advanced Excel with VBA Macros, by Swarup Das, publisher Blue Rose Publishers; 1st edition (6 October 2020), ISBN NO: 9390380316, 978-9390380312.

Practical Assignments of Advanced Excel

1. Assignment on basic functions
2. Assignment on logical functions
3. Assignment on lookup and reference functions
4. Assignment on date and time functions
5. Assignment on text functions
6. Assignment on power functions
7. Assignment on data handling
8. Assignment on pivot tables

Evaluation Pattern

The internal and external evaluation will be 50-50%

All the courses, both theory as well as practical, of major and minor are of two credits each. So, total marks will be 50. Out of 25 marks will be allotted for internal evaluation and 25 marks for external evaluation.

Theory Courses:

- Internal evaluation will be of 25 marks for which 2 continuous evaluation exams will be conducted.
- External evaluation will be of 25 marks

Practical Courses:

- Internal evaluation will be of 25 marks out of which 10 marks will be for assignment submissions done throughout the semester and a test/viva will be conducted for 15 marks.
- External evaluation will be of 25 marks

For Skill Enhancement and Vocational skills, 2 credit course is divided as 1 credit for theory and 1 credit of practical. In that case also internal and external will be of 50-50%. Distribution of 25 internal and 25 external (For total 2 credits) can be decided subject-wise.

Methods of assessment for internal evaluation:

Seminar, objective test, open book test, Quiz, viva, projects, assignments, group discussion, research paper review, case study, industrial visit

Passing percentage:

The student must secure at least 40% marks of that course to earn the full credit.

Examination	Credits	Marks out of	Passing marks (40%)
Internal	2	25	10
External	2	25	10

Note: There is separate passing for internal and external examinations.