MCA Curriculum & Syllabus
(2025-27)

Dept. of Computer Application
Maharaja Sriram Chandra Bhanja Deo University
Srirama Chandra Vihar, Takatpur, Baripada, Odisha

https://nou.nic.in

Name of the Programme: Master of Computer Applications (MCA)

(a) Programme's Mission & Objectives:

Mission

The broad objective of the MCA programme is to prepare post graduates for productive careers in software industry, corporate sector, Govt. organizations and academia by providing skill based environment for teaching and research in the core and emerging areas of the discipline. The Programme's thrust is on giving the students a thorough and sound background in theoretical and skill-oriented courses relevant to the latest computer software development. The programme emphasizes the application of software technology to solve mathematical, computing, communications/networking and commercial problems.

This Master's Degree Programme has been designed with a semester approach in mind. The first year courses are aimed at skills development in computers using various technologies abd core courses providing conceptual framework. The second year provides the specialization and the project work.

Objectives

1. Produce knowledgeable and skilled human resources which are employable in IT and ITES.
2. Impart knowledge required for planning, designing and building complex Application Software Systems as well as provide support to automated systems or application.
3. Produce entrepreneurs who can develop customized solutions for small to large Enterprises.
4. To develop academically competent and professionally motivated personnel, equipped with objective, critical thinking, right moral and ethical values that compassionately foster the scientific temper with a sense of social responsibility.
5. To develop students to become globally competent.
6. To inculcate Entrepreneurial skills among students

Programme Educational Objectives

The Master of Computer Applications Programme Educational Objectives aims to:

1. MCA graduates who will have successful careers based on their understanding of formal and practical methods of Application Development using the concepts of computer programming, software and design principles.
2. MCA graduates will demonstrate analytical and design skills including the ability to generate creative solutions and foster team-oriented, professionalism through effective communication in their careers.
3. MCA graduates who will exhibit effective work ethics and be able to adapt to the challenges of a dynamic job environment.

(b) Relevance of the Programme with HEI's Mission and Goals:

The Master of Computer Applications (MCA) programme to be offered is purely relevant and aligned with the goals and missions of the University.

(c) Nature of Prospective target groups of Learners:

After getting an MCA degree, one can enter any of the following roles :-

1. Joining IT firms Software and Web Developments

MCA has endless scopes in software. Those who with have skill and knowledge in programming and programming languages like HTML, CSS, jQuery, Ajax, PHP, MySQL, ASP, JSP, Tomcat, Python, and Perl etc. can choose an IT industry with Web development background.

2. System Analyst

A systems analyst is an information technology professional who specializes in analyzing, designing and implementing information systems. Systems analysts assess the suitability of information systems in terms of their intended outcomes.

3. Network Manager
A Network Manager Manage and maintain the network, as well as network performance monitoring, Identifying, installing and maintaining upgrades to the network

4. Data Scientist
After MCA learner can also make a career in Data Science. This field is emerging day by day.

5. Self Engagement
After MCA learner can also become an entrepreneur.

6. Go for Higher Studies
After MCA it is highly recommended that the learner should go for higher studies, depending upon his background and interest. After MCA the learner can opt followed degrees:

- M.Tech: 2 years course
- Ph.D.: Research course

(d) Appropriateness of the Programme to acquire specific Skills and Competence:
This program is concerned with design, development, and use of computer applications. Application implies incorporation of the operating system, a utility and a programming language. Thus, MCA graduates are able to develop and prepare documents, projects, presentations, design websites and demonstrate skills in running software programs. Besides imparting theoretical knowledge, a lot of stress is laid on hands-on training and overall development of the personality. Students graduating from this program can work gainfully in software services companies, outsourcing companies, and other allied sectors. The program has been designed to impart advanced knowledge in various areas of computer applications. It has been designed in such a way that student can complete it based on their advanced coursework. The structure of this program fosters a highly varied learning orientation in students by balancing core courses, advanced elective courses, and a substantial dissertation or project.

The University provides the opportunity to the human resource, basically to the poor and rural people to get a quality education in Computer Applications, which can fulfill the demand of the society.

Mark Distribution Structure of Theory Papers (100 marks each)

Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Question Pattern

A. Internal Assessment Examinations

- **Quiz: Total Marks: 10**
  - 1st Quiz: End of Unit I
  - 2nd Quiz: End of Unit II
  - Final Quiz Mark:
    - Average of two quizzes in case two quizzes conducted
    - Average of Best two quizzes in case three quizzes are conducted.
- **Mid Term Examination:**
  - End of Unit III covering all 3 units.
  - **Question Pattern: Total Marks: 20**
    - 6 Questions each of 2 marks: 6 x 2 = 12 marks (all questions compulsory)
    - 2 Questions each of 4 marks 2 x 4 = 8 marks (2 out of 3 questions)
- **Assignment: Total Marks: 10**
  - End of Unit IV
  - Assignment Mark: One assignment or Best of two in case two assignments are given.

B. End Term Examination

- **Question Pattern of Theory Examination: Total Marks: 60**
  - 5 Questions each of 2 marks: 5 x 2 = 10 marks (all questions compulsory)
  - 4 Questions each of 5 marks: 4 x 5 = 20 marks (4 out of 5 questions)
  - 3 Questions each of 10 marks: 3 x 10 = 30 marks (3 out of 5)
- **Practical Examinations: Total Marks: 100/50**
  - Practical: 60/30 Marks
  - Viva-Voce: 30/15 Marks
  - Record: 10/5 Marks
- **Project/ Dissertation (In Semester IV): Total Marks: 300**
  - Project: 200 marks
  - Report: 50 marks
  - Viva-Voce: 50 marks

MCA Course Structure – 2025-27

Sem	Paper ID	Paper Title	End Sem Exam	Internal Assessment			Full Marks	Credits	L-T-P
				Internal Exam	Quiz	Assignment			
I	CA-101	Discrete Mathematics	60	20	10	10	100	4	3-1-0
	CA-102	Data Structures using C	60	20	10	10	100	4	3-1-0
	CA-103	Computer System Architecture	60	20	10	10	100	4	3-1-0
	CA-104	Python Programming	60	20	10	10	100	4	3-1-0
	CA-105	Computer Network	60	20	10	10	100	4	3-1-0
	CA-106	Indian Knowledge System	30	20			50	2	2-0-0
	CA-107	Lab-1: Data Structure Using 'C'	50				50	3	0-0-3
	CA-108	Lab-2: Python Programming	50				50	3	0-0-3
		Life Skill Course	NA	NA	NA	NA	50	Non Credit	NA
	Total		**400**	**150**	**50**	**50**	**650**	**28**	
II	CA-201	Theory of Computation	60	20	10	10	100	4	3-1-0
	CA-202	Database Management Systems	60	20	10	10	100	4	3-1-0
	CA-203	Operating System	60	20	10	10	100	4	3-1-0
	CA-204	Design and Analysis of Algorithms	60	20	10	10	100	4	3-1-0
	CA-205	Lab-3: Database Management Systems	50				50	3	0-0-3
	CA-206	Interdisciplinary Computer Programming **(Open Elective)**	60	20	10	10	100	5	4-1-0
	CA-207	Lab-4: Operating Systems	50				50	3	0-0-3
	Total		**370**	**130**	**50**	**50**	**600**	**27**	
III	CA-301	Object Oriented Programming using Java	60	20	10	10	100	4	3-1-0
	CA-302	Computer Security	60	20	10	10	100	4	3-1-0
	CA-303	Machine Learning	60	20	10	10	100	4	3-1-0
	CA-304	**Elective – 1**	60	20	10	10	100	4	3-1-0
	CA-305	**SWAYAM MOOCS**	60	20	10	10	100	3	3-0-0
	CA-306	Lab-5: Programming using Java	50				50	3	0-0-3
	CA-307	Lab-6: Machine Learning	50				50	3	0-0-3
	Total		**370**	**130**	**50**	**50**	**600**	**25**	
IV	CA-401	**Elective – 2**	60	20	10	10	100	4	3-1-0
	CA-402	**Elective – 3**	60	20	10	10	100	4	3-1-0
	CA-403	**Elective – 4**	60	20	10	10	100	4	3-1-0
	CA-404	Project/ Dissertation	300	NA	NA	NA	300	8	0-0-8
	Total		**480**	**60**	**30**	**30**	**600**	**20**	3-1-0
Grand Total							**2450**	**100***	

***** Total credits may vary depending on the credit of chosen SWAYAM MOOCS course

Elective Papers (any one out of 4 paper electives)

	Paper ID	Paper Title
Elective 1	CA-304-A	Combinatorics and Graph Theory
	CA-304-B	Data Warehousing and Data Mining
	CA-304-C	Computer Graphics
	CA-304-D	Internet of Things
Elective 2	CA-401-A	Artificial Intelligence
	CA-401-B	Block Chain Technology
	CA-401-C	Mobile Computing
	CA-401-D	Internet & Web Technology
Elective 3	CA-402-A	Digital Image Processing
	CA-402-B	Cloud Computing
	CA-402-C	Data Science using Python
	CA-402-D	Compiler Design
Elective 4	CA-403-A	Organizational Behavior
	CA-403-B	Web and Text Mining
	CA-403-C	Object Oriented Analysis & Design
	CA-403-D	Bioinformatics

CA-101 Discrete Mathematics

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Mathematical foundations and Logical Skills

Learning Objectives:
The objective of this course is to:

1. To learn the mathematical foundations for Computer Science.
2. Topics covered essential for understanding various courses.

Learning Outcomes:
After successful completion of the course, student shall be able to:

1. Use Logical notation
2. Perform logical proofs
3. Determine equivalent logic expressions
4. Use graphs and trees
5. Apply basic and advanced principles of counting
6. Define sets and sequences

Course Content

Unit 1
Logic: Propositions and logical Operations, Conditional statements; Predicate Calculus First order logic, universal and existential quantifiers; Proof Techniques- methods of proof, Mathematical induction. Matrices: Determinant, Matrices, Solving System of Equations-Eigen Values and Eigen Vectors-Inverse of matrix

Unit 2 Graph Theory: Directed and undirected graphs, basic terminology, paths and circuits, bipartite graph, Eulerian graph.
Trees; definition and properties, rooted trees, binary trees, spanning trees, minimal spanning trees

Unit 3 Algebraic structures and application
Binary operations, groups, Subgroups, Cosets, Lagrange's theorem, Normal subgroup, Homomorphism,

Unit 4 Basic properties of lattices, Distributive and complemented lattices, Boolean algebra, disjunctive and conjunctive normal forms.

Text Books:

1. Truss, "Discrete Mathematics", Pearson
2. Rosen, " Discrete Mathematics", McGraw Hill
3. C.L. Liue, "Elements of Discrete mathematics" , Mc Graw Hill international Student edition.
4. J.K. Mantri & T.K. Tripathy, A Modern Approach to Discrete Mathematics & structure
5. Mott: Discrete Mathematics

CA-102 Data Structures using C

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Reasoning and Logical Skills, C Language

Learning Objectives:
The objective of this course is to:

1. To learn basic Programming Skills
2. To develop algorithms for performing different operations on data structures and implement in C language

Learning Outcomes:
After successful completion of the course, student shall be able to:

1. Write code for an Algorithm
2. Understand the flow of data and instructions in programming
3. Apply various data structures for problem solving

Course Content

Unit 1
Introduction to C language: Identifiers, Keywords, Data Types, Constant and Variables, Statements, Expressions, Operators. Decision and Loop Control Statements, Arrays, Functions, Storage Class, Recursion, Structures and Unions, Pointers, Dynamic Memory Allocation.

Unit 2
Stacks: Representation & Operations, Applications of Stack: Postfix expression evaluation, Infix to Postfix Conversion. Queue: Representation and Operations, Circular Queue. Linked List. Doubly Linked List, Circularly Linked List, Linked List Operations: Insertion, Deletion, Search, Reverse, Traversal.

Unit 3
Trees: Preliminaries, Tree: Representation & Implementation. Binary Search Trees: Representation & Operations - Find, Insert, Delete. Binary Tree Traversals. AVL Trees: Single Rotation, Double Rotation. Introduction to Graph, Graph representation: Adjacency matrix, Adjacency list, Graph Traversals: DFS, BFS.

Unit 4
Search: Sequential and Binary search, Sorting Algorithms: Insertion Sort, Selection Sort, Bubble Sort, Merge Sort, Heap Sort, Quick Sort, Counting Sort. Hashing: Hash Functions, Separate Chaining, Open Addressing - Linear Probing, Quadratic Probing.

References:

1. E. Balaguruswamy" Programming in C", Tata McGraw Hill
2. Brian W. Kernighan , Dennis M. Ritchie, "The C Programming Language (Ansi C Version)", 2nd Edition, PHI
3. H. Schildt, C The Complete Reference", Tata McGraw Hill
4. Y. Kanetkar,"Let us C", BPB Publication.
5. Peter van der Linden, "Expert C Programming: Deep C Secrets", Pearson India
6. Mark Allen Weiss, "Data Structure and Algorithm Analysis in C", 2nd Edition, Pearson Education.
7. Samir Kumar Bandyopadhyay, "Data Structures using C", 1st Edition, Pearson Education
8. Richard F. Gilberg, Behrouz A. Forouzan, "Data Structures: A Pseudocode Approach with C", 2nd Edition, Cengage Learning
9. Nick Parlante, "Linked List Problems", Stanford University

CA-103 Computer System Architecture

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Basic Computer Knowledge

Learning Objectives:
1. Conceptualize the basics of organizational and architectural issues of a digital computer and Classify and compute the performance of machines, Machine Instructions
2. Learn about various data transfer techniques in digital computer and the I/O interfaces

Learning Outcomes:
1. Explain the basics of organizational and architectural issues of a digital computer and Classify and compute the performance of machines, Machine Instructions
2. Describe various data transfer techniques in digital computer and the I/O interfaces

Course Content:

Unit 1
Structure of Computer Hardware: Basic function units and their Operational concepts, Single Bus Structure. Logic circuits: Logic functions, Synthesis if logic expression, Using AND, OR and NOT gates, Minimization of logic expressions using Karnaugh maps, Don‘t care conditions, Synthesis using NAND and NOR gates. Computer Arithmetic: Binary Arithmetic, Addition and Subtraction of signed number, Multiplication of positive number. Signed operand multiplication, Division, Floating point number representation and arithmetic

Unit 2
Basic processing of Instruction: Instruction code, Instruction set, Instruction sequencing, Instruction cycle, Instruction format, Addressing format, Addressing modes, Micro instruction, Data path, Hardwired controlled unit, Micro programmed control unit, Design of Control Unit and ALU.

Unit 3
Memory: Memory Hierarchy, RAM, ROM, Cache memory organization, Mapping techniques, Virtual memory, Mapping technique, Associative memory, Memory Interleaving, Secondary Storage, Flash drives. Design

Unit 4
Input/Output: Accessing I/O devices. Programmed I/O. Memory mapped I/O. Interrupt Driven I/O. Standard I/O interfaces. Synchronous and Asynchronous Data Transfer, DMA Data transfer.

Reference Books:

1. M. Morris Mano , "Computer System Architecture", Pearson Education
2. William Stallings, "Computer Organization and Architecture", Pearson Education
3. V. Rajaraman and T. Radhakrishnan, "Computer Organization and Architecture", PHI
4. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization", 5th Edition, Mc Graw Hill Education
5. A. S. Tanenbaum, "Structured Computer Organization", Pearson Education

CA-104 Python Programming

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Basic Computer Knowledge

Learning Objectives:
The objective of this course is to:
1. Develop a foundational understanding of Python programming, including data types, control flow, and file I/O.
2. Gain proficiency in object-oriented programming, functions, and exception handling in Python.

Learning Outcomes:
After successful completion of the course, student shall be able to:
1. Apply Python programming techniques to solve real-world problems.
2. Utilize Python's data structures, file I/O, and object-oriented features effectively in projects.

Course Content

Unit 1
Python Basics: Identifiers and Keywords, Python types, Operations and Conversions, Built-in Functions, Library Functions. Strings: Accessing String elements, String properties, String Operations. Control Flow Instruction: Decision Control Instruction, Logical Operators, Conditional Operators, Repetition Control Instruction, break and continue, pass statement. Console Input/Output: console input, console output, Formatted printing.

Unit 2
Lists: What are Lists? Accessing List elements, Basic List Operations, List Methods, List Varieties, List Comprehension. Tuples: What are Tuples? Accessing Tuple elements, Tuple Operations, Tuple Varieties, Conversion of List Comprehension to Tuple, Iterators and Iterables, zip() function. Sets: What are Sets? Accessing Set Elements, Set Operations, Set Functions, Mathematical Set Operations, Updating Set Operations, Set Comprehension. Dictionaries: What are Dictionaries? Accessing Dictionary elements, Dictionary Operations, Dictionary Functions, Nested Dictionary, Dictionary Comprehension.

Unit 3
Functions: What are Functions? Types of Arguments, Unpacking arguments, Lambada Functions, Recursive Functions. Modules: The Main module, Multiple modules, variations of import. Classes and Objects, Public and Private members, class declaration and Object Creation, Object Initialization, Class Variables and Methods, Accessing Object and Class Attributes, Calling Functions and Methods, Operator overloading, Iterators, User-defined Iterators, Generators, Generator Expression.

Unit 4
Inheritance: Reuse Mechanisms, Containership, Inheritance: Features, Types, Diamond Problem, Abstract classes, Runtime Polymorphism. Exception Handling: What's Exception? Syntax Errors, try-except, user-defined Exceptions, else Block, finally block. File Input/Output: I/O System, File I/O, Fead/Write operations, File Opening modes, with Keyword.

Reference Books:
1. Yashavant Kanetkar, Aditya Kanetkar, "Let Us Python", BPB
2. R. Nageswara Rao, "Core Python Programming", dreamtech
3. Mark Lutz, "Learning Python:, Orielly
4. John V Guttag. "Introduction to Computation and Programming Using Python", PHI
5. Wesley J. Chun. "Core Python Programming - Second Edition", Prentice Hall
6. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, "Data Structures and Algorithms in Pyhon", Wiley
7. Kenneth A. Lambert, "Fundamentals of Python – First Programs", CENGAGE Publication
8. Luke Sneeringer, "Professional Python", Wrox

CA-105 Computer Network

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Basic knowledge of Computer organization, and programming languages

Learning Objectives:

1. Describe the general principles of data communication.
2. Describe how computer networks are organized with the concept of layered approach.
3. Describe how signals are used to transfer data between nodes.
4. Implement a simple LAN with hubs, bridges and switches.

Learning Outcomes:

1. Describe the basis and structure of an abstract layered protocol model
2. Independently understand basic computer network technology.
3. Identify the different types of network topologies and protocols.
4. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.
5. Identify the different types of network devices and their functions within a network
6. Understand and building the skills of subnetting and routing mechanisms.
7. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Course Content:

Unit 1

Introduction: Data Communication, Networks, Protocols and standards. Point to Point and Multipoint Line Configuration, Network Topologies: Mesh, star Tree Bus, Ring, Transmission Modes: Simplex, half Duplex, Full Duplex, Networks: LAN, WAN, MAN, The OSI model: Function of the layers.

Encoding and Modulating: digital-to-digital conversion: Unipolar, Polar, Bipolar, Analog to digital conversion: PAM, PCM, Digital to analog conversion: ASK, FSK, PSK, QAM and analog-to-analog conversion: AM, FM, PM, Transmission of Digital Data: Parallel and serial transmission, Modems, Guided and unguided transmission media, Transmission impairment.

Unit 2

Multiplexing: Frequency division, Wave division and time division multiplexing, Error Detection and Correction: Type of errors, Redundancy Checks (VRC, LRC, CRC), Data Link Control: Line Discipline, Flow Control, Error Control. Local Area Networks: IEEE 802 standards, Ethernet, Token Bus, Token Ring. Switching: Circuit Switching, Packet Switching, Message Switching.

Unit 3

Integrated Services Digital Network (ISDN): Services, History Subscriber access to ISDN, The ISDN Layers. Frame Relay: Introduction, Frame Relay Operation, Frame Relay Layers, Congestion Control, Leaky Bucket Algorithm, and Traffic Control.

Unit 4

Networking and Internetworking Devices: Repeaters, Bridges, Routers, Gateways. Application Layer: Client-Server Model, FTP, HTTP, World Wide Web.

Text books:

1. Forouzan B.A. , “Data Communication and Networking”, 4th Edition, McGraw Hill Education (India)
2. Black, U., “Computer Networks-Protocols, Standards and interfaces”. 2nd Edition, Prentice Hall

CA-106 Indian Knowledge System

Credits	**2**
Assignment / Seminar Presentation	5
Quiz	5
Internal Assessment / Mid Semester	10
Theory Examination (End Term)	30
Total	**50**
Time Duration of Theory Examination	2 Hours

Prerequisites:
Mathematical foundations and Logical Skills

Learning Objectives:
The objective of this course is to:

1. Explore historical development and contributions of Indian number systems and mathematics.
2. Analyze the relationship between linguistics, computational aspects, and Sanskrit grammar.
3. Understand the Indian scheme of knowledge classification and its underlying principles.

Learning Outcomes:
After successful completion of the course, student shall be able to:

1. Identify and explain the historical context of Indian number systems and mathematics.
2. Analyze the relationship between linguistics, computational aspects, and Sanskrit grammar.
3. Apply the Indian epistemological framework to evaluate knowledge claims and reasoning processes.

Course Content

Unit 1
Overview of Indian Knowledge Systems: Definition and Scope of Indian Knowledge Systems; Historical development, and Significance; Interdisciplinary nature of IKS; Philosophical Foundations of IKS: Vedic, Upanishadic, and other classical texts; Key philosophical concepts: Dharma, Karma, Moksha; Contribution of IKS to global knowledge.

Unit 2
Number Systems: Number systems in India - Historical evidence, Salient aspects of Indian Mathematics, Bhūta-Saṃkhyā system, Kaṭapayādi system, Piṅgala and the Binary system, Binary mathematics and combinatorial problems in Chandaḥ Śāstra.

Unit 3
Linguistics: Introduction to Linguistics, Aṣṭādhyāyī, Phonetics, Word generation, Computational aspects, Mnemonics, Recursive operations, Rule based operations, Sentence formation, Verbs and prefixes, Role of Sanskrit in Natural Language Processing.

Unit 4
Knowledge Framework and classifications: Indian scheme of knowledge, The knowledge triangle, Attributes – the properties of substances and Action – the driver of conjunction and disjunction, Sāmānya; viśēṣa; samavāya, Pramāṇa – the means of valid knowledge, Saṃśaya – ambiguities in existing knowledge, Framework for establishing valid knowledge, Potential fallacies in the reasoning process, Siddhānta: established tenets in a field of study.

Reference Books:

1. Introduction to Indian Knowledge System: Concepts and Applications, B. Mahadevan, Vinayak Rajat Bhat, Nagendra Pavana R.N., PHI.
2. IKS: The Knowledge System of Bharata, Bhag Chand Chauhan, Garuda Prakashan.
3. The Computation Meme: Computational thinking in the Indic tradition, Kanchi Gopinath, Shailaja D Sharma, IISc Press.
4. Indian Knowledge System "Journey From Ancient Scientific & Vedic Theories" To "Modernization of Science & Society", Arpit Srivastava, Book Rivers Publications.
5. Scientific basis of Indian Knowledge System, Saroj Sharma, Bal Krishna Rai, Sunita Joshi Kathuria, Shipra Publications.
6. Indian Knowledge Systems Vol – I & II, Kapoor Kapil, Singh Avadhesh, Indian Institute of Advanced Study, Shimla, H.P.
7. The Wonder that is Sanskrit, Sampad and Vijay. Sri Aurobindo Society, Puducherry.

CA-107 Lab-1: Data Structure Using 'C'

Prerequisites:
Knowledge of Data Structures, C Language
Learning Objectives:
The objective of this course is to:
1. To practice with programming skill and improve the programming logic.
2. To apply various techniques with data such storing, inserting, deleting and traversing of data.
3. Utilization of various data structures such as Linked List Structures, Stack, Queues, Trees and Graphs.

Learning Outcomes:
After successful completion of the course, student shall be able to:
1. Write the code for a large program after overcoming the time and space complexity.
2. Frequent use of various algorithms such as searching, sorting, traversing with data structures.
3. Define the data in an optimal way.

Course Contents:

1. Creating and editing simple C programs, compilation and execution.
2. Program on Expressions, Operators, Simple Arithmetic, Decision and Loop Control Statements.
3. Program demonstrating Single & Multidimensional arrays
4. Program demonstrating Functions, recursion,
5. Program demonstrating structure and union
6. Program demonstrating Pointers and dynamic memory allocation
7. Program demonstrating Array based Stacks: Postfix expression evaluation, Infix to Postfix Conversion
8. Program demonstrating Array based Queues: Queue operations, Circular queue
9. Program demonstrating Linked List, Doubly Linked List, Circularly Linked List, Linked List Operations: Insertion, Deletion, Search, Traversal.
10. Implementing Stacks & Queues using Linked List.
11. Program demonstrating Binary Search Trees: Representation & Operations - Find, Insert, Delete. Binary Tree Traversals: inOrder, preOrder, postOrder
12. Implementation of Binary search.
13. Sorting Implementations: Insertion Sort, Selection Sort, Bubble Sort, Merge Sort, Heap Sort, Quick Sort, Counting Sort.
14. Simple Hash Table implementation

CA-108 Lab-2: Python Programming

Prerequisites:
Basic computer literacy, Logical thinking and basic math

Learning Objectives:
The objective of this course is to:

- Understand Python syntax and control structures
- Work with data structures and functions
- Apply object-oriented programming
- Handle input/output and file operations

Learning Outcomes:
After successful completion of the course, student shall be able to:

- Write and execute Python programs
- Use lists, tuples, sets, and dictionaries
- Develop modular and class-based code
- Perform file I/O and handle exceptions

Course Content:

1. Write a program to display a personalized welcome message by taking the user's name and age as input, calculate the user's birth year, and print all the details in a nicely formatted string.
2. Write a program that takes marks in five subjects as input (in string format), converts them to float, calculates the total, percentage, and grade (A/B/C/Fail based on percentage).
3. Write a program that accepts a sentence from the user, counts vowels, consonants, digits, and special characters, and then prints a report.
4. Write a program to generate a multiplication table for a number (between 1 and 20) entered by the user. Include input validation and display a warning if the number is out of range.
5. Write a billing system that takes item names, prices, and quantities (multiple items), calculates total and GST (18%), and prints a formatted receipt.
6. Write a program that takes n numbers from the user, stores them in a list, removes duplicates, sorts the list in descending order, and displays statistical summary: min, max, average.
7. Write a program to create a 2D list (matrix) of m x n integers taken from the user, compute and display the row-wise and column-wise sums.
8. Write a program that stores names and scores of 5 students using tuples and displays the names of students who scored above average. Use zip() and enumerate() meaningfully.
9. Write a program that takes names and phone numbers from the user to create a phonebook dictionary. Allow the user to search by name, delete a contact, and list all contacts alphabetically.
10. Write a program that implements a recursive function to calculate factorial and a lambda function to calculate cube of numbers in a given list. Also, include a menu to select the operation.
11. Create a module mathutils.py with functions for GCD, LCM, and prime check. Write a main script that imports the module and lets the user choose an operation to perform on input numbers.
12. Write a BankAccount class with methods to deposit, withdraw, check balance, and display account summary. Create multiple account objects and demonstrate operations using user input.
13. Write a program with a base class Shape and derived classes Rectangle, Circle, and Triangle. Use method overriding to compute area for each and demonstrate polymorphic behavior.
14. Write a calculator program that handles division, modulus, and exponentiation. Handle division by zero and invalid input using exceptions. Define a custom exception for negative exponents.
15. Write a student management program that stores student name, roll, and marks in a file. Implement functions to add, view, and search records, using file I/O operations and formatted output.

CA-201 Theory of Computation

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
To have a strong background in the fundamentals of discrete mathematics (symbolic logic, set, induction, number theory, summation, series, combinatorics, graph, recursion, basic proof techniques, etc.), algorithms and data structures.

Learning Objectives:
To explore the theoretical foundations of computer science from the perspective of formal languages and classify machines by their power to recognize languages.

Learning Outcomes:
To understand the basic properties of formal languages and grammars. differentiate regular, context-free and recursively enumerable languages. make grammars to produce strings from a specific language. including decidability and intractability

Course Contents:

Unit 1 Introduction To Finite Automata : Alphabets and languages- Finite Representation of Languages. Deterministic Finite Automata – Non- deterministic Finite Automata – Equivalence of Deterministic and Non-Finite Automata; Conversion Problem Complexity: The Classes P and NP ; NP-Completeness; NP-Hard Problems

Unit 2 Properties of the Languages Accepted by Finite Automata – Finite Automata and Regular Expressions Context free languages: Context –Free Grammar – Regular Languages and Context-Free Grammar Pushdown Automata – Pushdown Automata and Context-Free Grammar – Properties of Context-Free Languages – Closure Properties –Deterministic Pushdown Automata and Context – Free Languages – Top- down and Bottom – Up Parsing.

Unit 3 Turing machines: The Definition of Turing Machine – Computing with Turing Machines — Types of Turing machines some ; Recursive Functions

Unit 4 Uncomputability: The Halting Problem – Turing-Enumerability, Turing – Acceptability, and Turing - Decidability – Unsolved problems about Turing machines and Recursive Functions

Text Book: Elements Of The Theory Of Computation, Harry R Lewis, Cristos h. Papadimitriou, Pearson Education / Prentice-Hall of India Private Limited.

Reference Books:

1. Hopcroft. J.E, J.D.Ullman, "Introduction to Automata Theory, Languages, and Computation", Addison-Wesley
2. Kamala Krithivasan, Rama R., "Introduction to Formal Languages, Automata Theory and Computation", Pearson India
3. Michael Sipser, "Introduction to the Theory of Computation", Cengage Learning

CA-202 Database Management Systems

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Comfortable with the relational model, SQL, and the basic functions of database systems

Learning Objectives:
1. To learn the fundamental elements of database system.
2. To learn the basic concepts of relational database management systems.
3. To learn various SQL commands.

Learning Outcomes:
Defines the basics of the relational data model. Lists the database design process steps and also will be able to design and implement properly structured databases that match the standards based under realistic constraints and conditions. Develops an Entity-Relationship model based on user requirements.

Course Content:
Unit 1
Database systems Architecture-data Abstraction, data independence, Codd's Rule for RDBMS, Data definitions and data manipulation Languages. Data models- Entity Relationship (ER), mapping ER Model, network, relational and Object Oriented data models, Integrity Constraints and data manipulation operations.

Unit 2
Relation Query Languages, Relational Algebra, Tuple and Domain relational calculus, SQL and QBE. Relational database Design: Domain and Data dependency, Armstrong's Axioms, Normal Forms, Dependency Preservation, Lossless design.

Unit 3
Query Processing and Optimization: Evaluation of relational algebra Expressions, Query equivalence, join strategies, Query optimization algorithms, Storage strategies: Indices, B-tree, Hashing, Transaction processing: Recovery and Concurrency control, Locking and Timestamp based schedulers, Multiversion and Optimistic Concurrency control schemes.

Unit 4
Distributed Databases: Introduction, DDBMS architectures, Homogeneous and Heterogeneous Databases, Distributed data storage, Distributed transactions, Commit protocols, Availability, Concurrency control & recovery in Distributed Databases.

Text books:

1. Elmaski & Navathe, "Fundamentals of database systems", 4th edition, Pearson education
2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", 6th Edition, Mcgraw Hill Education
3. C.J. Date, "An introduction to database systems", Pearson Education.
4. Bipin Desai, "An introduction to database system", Galgotia Publications
5. Raghu Ramkrishnan, Johannes Gehrke, "Database Management Systems", 2nd Edition, McGraw Hill International

CA-203 Operating System

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Basic Computer Knowledge

Learning Objectives:
1. To understand overall functionality of Operating System such as Process Management, Memory Management, File Management and Security Issue.
2. To Provide sufficient understanding of operating system design
3. To understand the impact of operating system on application systems design and performance.

Learning Outcomes:
After successful completion of the course, student shall be able to:
1. Exhibit familiarity with the fundamental concepts of operating systems.
2. Apply a mature understanding of operating system design and how it impacts application systems design and performance.
3. Exhibit competence in recognizing operating systems features and issues.

Course Contents:
Unit 1
Introduction to Operating Systems. User View & System View of OS. Operating System Concepts, Interrupts & System Calls. Operating System Services. Processes. Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, Priority, Round-Robin, Multilevel Queue, Multilevel Feedback Queue).

Unit 2
Inter Process Communication. Process Synchronization: Background, The Critical-Section Problem, Semaphores, Counting Semaphores & Binary Semaphores. The Dining-Philosophers Problem. Monitors.

Unit 3
Deadlocks: Basic cause of deadlock, Conditions for deadlock, Resource-Allocation graph. Deadlock Prevention, Deadlock Avoidance with Banker's algorithm. Deadlock Detection.

Unit 4
Memory Management Strategies: Background (Address Binding, Logical vs Physical Address space), Swapping, Contiguous Memory Allocation: Dynamic Memory Allocation(First-fit, Best-fit, Worst-fit), Fragmentation. Paging, Page Tables. Segmentation. Virtual Memory Management: Background (Virtual Memory & Virtual Address Space). Demand Paging, Page faults, Page replacement techniques: FIFO, Optimal, LRU. Frame allocation techniques, Thrashing.

Reference Books:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, “Operating System Concepts”, 8th Edition, Wiley India
2. Harvey M. Deitel, Paul J. Deitel, David R. Choffnes, “Operating Systems”, 3rd Edition, Pearson Education
3. William Stallings, “Operating Systems: Internals and Design Principles”, 6th Edition, PHI Learning / Pearson Education
4. Andrew S. Tanenbaum, “Modern Operating Systems”, 3rd Edition, PHI Learning / Pearson Education

CA-204 Design and Analysis of Algorithms

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Basic knowledge of programming and mathematics. & data structure

Learning Objectives:
To Write rigorous correctness proofs for algorithms. Demonstrate a familiarity with major algorithms and data structures. Apply important algorithmic design paradigms and methods of analysis. Synthesize efficient algorithms in common engineering design situations.

Learning Outcomes:

1. An ability to apply design and development principles in the construction of software systems of varying complexity.
2. An ability to use current techniques, skills, and tools necessary for computing practice

Course Content:

Unit 1
Introduction to analysis and design of algorithm, Growth of functions, Asymptotic notations, Recurrences, Solution of recurrences by substitution, Recurrence tree and the master method. Divide and conquer algorithms (Runtime analysis of merge sort, quick sort and heap sort algorithms), Priority queue, Data structure for disjoint sets.

Unit 2
Dynamic programming approach: Matrix chain multiplication, longest common subsequence. Greedy method: Activity solution problem, Greedy verses dynamic programming, Huffman codes. Concept of backtracking, branch & bound design techniques

Unit 3
Graph algorithms: Minimal spanning tree (Kruskal and Prim's algorithms), Single source shortest paths (Bellman-Ford and Dijkstra's algorithm), Floyd's algorithm, Flow Network, Ford-Fulkerson method, Fast Fourier Transform

Unit 4
Rabin-Karp string matching algorithm, NP-Completeness, Polynomial time solvability, Verification and Reducibility, NP complete problems (without proof), Approximation algorithm for the traveling salesman problem

Reference books :

1. T.H. Cormen, C.E. Leiserson, R.L. Rivest and L.Stein, "Introduction to Algorithms" , Second Edition, PHI Learning
2. E. Horowitz, S. Sahani, S. Rajsekharan, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, 2007
3. Mark Allen Weiss, "Data Structure and Algorithm Analysis in C", 2nd Edition, Pearson Education.
4. J. Kleinbers, E.Tardos, Algorithm design, Pearson Education Inc., New Delhi , 2006
5. R. Johnsonbaugh, M. Schaefer, "Algorithms", Pearson Education Inc., New Delhi , 2004
6. Kenneth A. Berman & Jerome L. Paul, "Algorithms", Revised Edition, 2005, CENGAGE Learning India Pvt. Ltd., New Delhi.
7. Anany V. Levitin, "Introduction to the Design and Analysis of Algorithms", Second Edition, 2007, Pearson Education Inc., New Delhi.
8. Michael T. Goodrich and Roberto Tamassia, "Algorithm Design: Foundations, analysis, and Internet Examples", 2nd Edition, Wiley India Pvt. Ltd., New Delhi

CA-205 Lab-3: Database Management Systems

1. Database Schema for a employee scenario Consider the following tables namely “Departments” ,”Employees” and “Salary”
Departments (DEPTNO , DEPTNAME , DEPTLOC);
Employees (EMPNO, EMPNAME , JOB,DEPTNO,DOJ);
Salary(EMPNO,MONTH/YEAR,SAL)
For the above schema, perform the following:

1. Create all above tables.
2. Rename Employee table as Employee_Master
3. Add column Date_of_Birth to Employee_Master
4. Delete Date_of_Birth for Employee_Master
5. Modify EMPNAME field to accept 100 characters.
6. Insert around 10 records in each of the tables
7. List all the employees with their EMPNO.
8. List all the employee numbers whose salary is above 20000 .
9. List all the employees who have joined in the year 2002.
10. List all the department numbers where department location is Odisha.
11. Truncate the Employee_Master table.

2. Database Schema for a Student Library scenario
Student(Stud_no : integer, Stud_name: string)
Membership(Mem_no: integer, Stud_no: integer)
Book(book_no: integer, book_name:string, author: string)
Iss_rec(iss_no:integer, iss_date: date, Mem_no: integer, book_no: integer)

For the above schema, perform the following:

1. Create the all above tables .
2. Insert around 10 records in each of the tables
3. Add column S_Address to student
4. List all the student names with their student numbers.
5. List all the issues for the current date with Book number.
6. List the book names whose author is CJ DATE.
7. Give number student with book no. 5.
8. Truncate the table Iss_rec.

3. Considering Database Schema for a employee scenario
Departments (DEPTNO , DEPTNAME , DEPTLOC);
Employees (EMPNO, EMPNAME , JOB,DEPTNO,DOJ);
Salary(EMPNO,MONTH/YEAR,SAL)
For the above schema, perform the following:

1. Create the table with all the required constraints.
2. Display all the details of the records whose employee name starts with “A”
3. Display all the details of the records whose employee name does not start with “A”
4. Display the rows whose salary ranges from 15000 to 30000.
5. Calculate the total and average salary amount of the salary table.
6. Count the total records in the emp table.
7. Determine the max and min salary and rename the column as max_salary and min_salary.
8. Display the last day of that month in ¯05-Oct-09 .
9. Find how many job titles are available in the employee table.
10. What is the difference between maximum and minimum salaries of employees in the organization?
11. Display the maximum salary of a group by month_year.

CA-206 Interdisciplinary Computer Programming (Open Elective)

Credits	**5**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisite
Basic math and logical reasoning skills.
Familiarity with computers (no prior programming required)

Learning Objectives:
The objective of this course is to:
1. Use Python libraries (NumPy, pandas, Matplotlib) for data analysis and visualization.
2. Apply SciPy, ChemPy, and Biopython to scientific problems.
3. Analyze social networks and economic data using Python tools

Learning Outcomes:
After successful completion of the course, student shall be able to:
1. Write Python programs to solve interdisciplinary problems.
2. Use Python libraries for data analysis in various fields.
3. Apply computational tools to science (physics, chemistry, biotechnology).
4. Analyze social networks and model economic data.
5. Develop solutions for interdisciplinary projects.

Course Content:

Unit 1: Introduction to Python Programming
Introduction to programming concepts. Python basics: variables, data types, operators, control structures, loops. Functions and modular programming. Input/Output operations. Error handling.

Unit 2: Data Structures and Libraries for Interdisciplinary Applications
Lists, tuples, sets, dictionaries. Introduction to NumPy: arrays and matrix operations. Introduction to pandas: data manipulation and analysis. File handling and working with CSV and Excel files. Data visualization with Matplotlib.

Unit 3: Computational Interdisciplinary Applications - I
SciPy for Physics: Numerical Integration (Rectangular, Trapezoidal, Simpson), Solving Ordinary and Partial Differential Equations, Simulations of Projectile motion and Wave Propagation. ChemPy for Chemistry: Chemical Reactions (Balancing Chemical Equations and Calculating Stoichiometric Coefficients), Solving Equilibrium, Thermodynamics (Calculating Enthalpy, Entropy and Gibbs Free Energy Changes), Reaction Kinetics (Simulating Reaction Rates, Fitting Rate Data).

Unit 4: Computational Interdisciplinary Applications - II. Python in Business Management and Economics: Exploratory data analysis using Pandas and Matplotlib. Visualization for Business Insights: creating dashboards with Seaborn and Plotly. Advanced visualizations like heatmaps and geographical data charts. Economic Modeling (Solow Growth Model, Consumer Choice Model), Regression Analysis, forecasting with ARIMA model and Exponential Smoothing.

Reference Books:

1. Yashavant Kanetkar, Aditya Kanetkar, "Let Us Python", BPB
2. R. Nageswara Rao, “Core Python Programming”, Dreamtech
3. Mark Lutz, "Learning Python:, Oreilly
4. John V Guttag. “Introduction to Computation and Programming Using Python”, PHI
5. Wesley J. Chun. “Core Python Programming - Second Edition”, Prentice Hall
6. Sedgewick/Wayne/Dondero, “Introduction to Programming in Python: An Interdisciplinary Approach”, Pearson
7. Jessen Havill, “Discovering Computer Science: Interdisciplinary Problems, Principles, and Python Programming”, Chapman & Hall / CRC Textbooks.
8. Galit Shmueli, Nitin R. Patel, Peter C. Bruce, Data Mining for Business Analytics: Concepts, Techniques, and Applications in Python, Wiley.

CA-207 Lab-4: Operating Systems

Prerequisites:
Basic understanding of Operating System concepts
Familiarity with command-line interface
Basic knowledge of file and directory structure

Learning Objectives:
Learn basic and advanced Linux shell commands
Develop shell scripts for task automation
Understand process management and file handling

Learning Outcomes:
Execute Linux commands for file, directory, and user management
Write shell scripts using variables, loops, and conditions

Course Contents:
1. Write a shell script to display the current username, home directory, present working directory, and list all hidden files in the home directory.
2. Write a shell script to accept two numbers from the user and perform all arithmetic operations (add, subtract, multiply, divide, modulus).
3. Write a shell script to accept a student's marks in 5 subjects, calculate total, percentage, and display pass/fail status using arithmetic operations.
4. Write a shell script to accept a filename from the user and check whether it exists, readable, writable, and whether it is a regular file or directory.
5. Write a shell script to display the Fibonacci series up to a number n entered by the user, using while or for loop.
6. Write a shell script to count the number of vowels, consonants, digits, and special characters in a string entered by the user.
7. Write a shell script to store names of 5 users in an array and: Display all names, Display length of each name, and Search for a name in the array
8. Write a shell script to accept two filenames as command line arguments and compare whether they are identical, different, or one is missing.
9. Write a shell script to compare two numbers passed as command-line arguments and print the larger, smaller, or if they are equal.
10. Write a shell script to define a function to find the factorial of a number and use it in the main program to display factorials from 1 to 10.
11. Write a shell script to take input of multiple student names and roll numbers, store them in a file, and display the contents line by line with line numbers.
12. Write a shell script to take multiple lines of input from the user, save them in a file, then read the file line by line and print with line numbers.
13. Write a shell script to accept a file containing names and marks, sort students by marks (descending), and display the top 3 scorers.

CA-301 Object Oriented Programming using Java

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Basic Programming Skills

Learning Objectives:
1. Understand the basics of Java, Java toolchains and features of the language
2. Use Java programming language with object-oriented programming principles.

Learning Outcomes:
After successful completion of the course, student shall be able to:
1. Design, create, build, and debug Java applications
2. Write Java programs using object-oriented programming techniques including classes, objects, methods, instance variables, composition, inheritance, and polymorphism.

Course Contents:
Unit 1
Basics of Java. JDK & JRE. Writing, Compiling, and running Java programs. Data Types. Variables. Type conversion and casting. Arrays. Operators. Control Statements. Objects, and Encapsulation. Classes. methods, constructors. Keywords: this, static, final. Access control.

Unit 2
Inheritance in Java. Use of super keyword. Multilevel inheritance. Calling of constructors. Polymorphism: Method overloading and overriding. Abstract class. Runtime Polymorphism through Inheritance. Polymorphism using interface. Generics in Java: Generics, Generic methods and constructors. Primitive Type Wrappers. Packages: Defining a package, Access Protection, Importing packages.

Unit 3
Exception Handling in Java: Understanding Exception handling, Exception types, Uncaught Exceptions, Using try and catch, Multiple catch, Nested try statements, Understanding of throw, throws and finally, Java's Built-in Exceptions, Chaining Exceptions. Collection Framework in Java: Collection Interfaces and Classes, Iterators, Using List interface and classes like ArrayList and LinkedList, Map Interfaces and classes, Using Hashmap, Using Comparators.

Unit 4
I/O in Java: I/O Basics, Streams and Stream Classes: Byte Stream Classes, Character Stream Classes. Reading and Writing Files. Multithreading programming with java: Understanding Java Thread Model, The Thread class and Runnable interface, Thread creation, Synchronization, Interthread communication.

Reference Books:

1. Herbert Schildt, "Java: The Complete Reference", Tenth Edition, Oracle Press
2. E. Balagurusamy, "Programming with Java ", 6th Edition, McGraw-Hill Education
3. Harvey M. Deitel & Paul J. Deitel, "Java How to Program", 8th Edition, PHI
4. Debasish Jana, ―Java and Object-Oriented Programming Paradigm", PHI

CA-305 Software Engineering

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Basic Programming Skills

Learning Objectives: To enable students to design and build software using the knowledge acquired in this course

Learning Outcomes: A student who takes this course should be able to
1. Apply SE principles to software design in future
2. Modularize software development
3. To design reusable components
4. Testing types and levels
5. Using CASE tools

Unit 1
Software Engineering paradigms – Waterfall Life cycle model – Spiral Model – Prototype Model – Evolutionary model -fourth Generation Techniques – Planning – Software Project Scheduling, – Risk analysis and management – Requirements and Specification –Project Plan and SRS

Unit 2
Abstraction – Modularity – Software Architecture – Cohesion – Coupling – Various Design Concepts and notations – Real time and Distributed System Design – Documentation – Dataflow Oriented design – Designing for reuse – Programming standards – C
Study for Design of any Application Project.

Unit 3
Software Testing Fundamentals – Software testing strategies – Black Box Testing – White Box Testing – System Testing – Object Orientation Testing – State based Testing - Testing Tools – Test Case Management – Software Maintenance Organization – Maintenance Report – Types of Maintenance

Unit 4
Scope – Classification of metrics – Measuring Process and Product attributes – Direct and Indirect measures – Cost Estimation - Reliability – Software Quality Assurance – Standards – COCOMO model, Need for SCM – Version Control – SCM process – Software Configuration Items – Taxonomy – CASE. Repository – Features – Software reverse Reengineering

Text Books:
1. Mall Rajib, "Fundamentals of Software Engineering", PHI.
2. Roger S. Pressman, "Software Engineering A practitioner's Approach", 5th Edition, McGraw-Hill Higher Education.

CA-302 Computer Security

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Computer Network concepts, Algorithms, Number The

Learning Objectives:

1. Learn fundamentals of cryptography and its application to network security.
2. Understand vulnerability analysis of network security.
3. Acquire background on hash functions; authentication; firewalls; intrusion detection techniques.

Learning Outcomes:

1. Describe network security services and mechanisms.
2. Symmetrical and Asymmetrical cryptography.
3. Data integrity, Authentication, Digital Signatures.
4. Various network security applications, IPSec, Firewall, IDS, Web security, Email security, and Malicious software etc

Course Content:

Unit 1

The Security Problem in Computing: The meaning of computer Security, Computer Criminals, Methods of Defense; Elementary Cryptography: Substitution Ciphers, Transpositions, Making "Good" Encryption Algorithms, Private-Key Cryptosystems, The Data Encryption Standard, The AES Encryption Algorithm, Public-Key Cryptosystems, Public Key Encryptions, Uses of Encryption

Unit 2

Program Security : Secure Programs, Non-malicious Program Errors, viruses and other malicious code, Targeted Malicious code, controls Against Program Threats, Protection in General-Purpose operating system protected objects and methods of protection memory and address protection, File protection Mechanisms, User Authentication Designing Trusted O.S : Security policies, models of security, trusted O.S. design, Assurance in trusted OS, Implementation examples.

Unit 3

Database Security : Security requirements, Reliability and integrity, Sensitive data, Inference, multilevel database, proposals for multilevel security

Network Security: Threats in Network, Network Security Controls, Firewalls, Intrusion Detection Systems, Secure E-mail.

Unit 4

Administering Security: Security Planning, Risk Analysis, Organizational Security policies, Physical Security; The Economics of Cyber security; Privacy in Computing; Legal and Ethical Issues in Computer Security: Protecting Programs and data, Information and the law, Rights of Employees and Employers, Software failures, Computer Crime.

Reference Books:

1. Charles P. Pfleeger & Shari Lawrence Pfleeger, "Security in Computing", 4th Edition,Pearson Education
2. Josef Pieprzyk, Thomas Hardjono, Jennifer Seberry, "Fundamentals of Computer Security", Springer & Universities Press India
3. Dieter Gollmann, "Computer Security", 2nd Edition, Wiley India
4. William Stallings & Lawrie Brown, "Computer Security: Principles and Practice", 1st Edition, Pearson Education
5. Charlie Kaufman, Radia Perlman & Mike Speciner, "Network Security: Private Communication in a Public World", 2nd Edition, PHI Learning
6. Chuck Easttom, "Computer Security Fundamentals", 1st Edition, Pearson Education
7. Alfred Baasta, "Computer Security", 1st edition, CENGAGE Learning

CA-303 Machine Learning

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisite:
Linear algebra, Statistics, Python Programming.

Learning Objectives

The objective of the course is

1. To understand the basic theory underlying machine learning.
2. To be able to formulate machine learning problems corresponding to different applications.
3. To understand a range of machine learning algorithms along with their strengths and weaknesses.
4. To be able to apply machine learning algorithms to solve problems of moderate complexity.
5. To apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models.

Learning Outcomes

After completing this course, the student will be able to

1. Appreciate the importance of visualization in the data analytics solution
2. Apply structured thinking to unstructured problems
3. Understand a very broad collection of machine learning algorithms and problems
4. Learn algorithmic topics of machine learning and mathematically deep enough to introduce the required theory
5. Develop an appreciation for what is involved in learning from data.
6. Get employability as the prospect of Machine Learning in India, as well as in other parts of the world, is high in comparison to other career fields when it comes to job opportunities.

Unit I:
Introduction: Natural vs. Machine Learning; Types of Learning: Supervised and Unsupervised Learning; Decision Tree Learning: Properties, Top-Down Induction, Entropy, Overfitting; Regression: Linear Regression, Logistic Regression.

Unit II:
Artificial Neural Networks: Perceptron Learning, Multi-layer Perceptron, Backpropagation; Support Vector Machine (SVM): Separation, Classification; Model Evaluation & Performance Metrics: Confusion Matrix, Accuracy, Precision, Recall, F1 Score, Sensitivity, Specificity, ROC-AUC; Error Metrics: Residual Error, Mean Square Error, RMSE, Loss; K-Fold Cross-Validation Technique.

Unit III:
Bayesian Learning: Naïve Bayes Classifier, Generative Models, Inference; Instance-Based Learning: Distance Metrics, K-Nearest Neighbor, Variations; Clustering: K-Means, Hierarchical Agglomerative, DBSCAN.

Unit IV:
Deep Learning: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Applications; Evolutionary Learning: Genetic Algorithm, Particle Swarm Optimization, Ant Colony Optimization, Firefly Algorithm

Reference Books:

1. Ethem Alpaydin, Introduction to Machine Learning, MIT Press/ Prentice Hall of India.
2. K.L.Priddy and Paul E. Keller, Artificial Neural Networks- An Introduction, PHI/ EEE edition. 3. Nils J. Nilsson, An Introduction to Machine Learning.
3. Tom Mitchell, Machine Learning, Mc Graw Hil , 1997.

CA-306 Lab-5: Programming using Java

Prerequisites:
Basic Programming Skills

Learning Objectives:
3. To know compilation, execution and debugging of Java Programs
4. To apply Object Oriented Programming Principles in the Programs

Learning Outcomes:
After successful completion of the course, student shall be able to:
3. Build a basic software using Java programming language
4. Get in-depth experience in classes, objects, methods, instance variables, object composition, inheritance, and polymorphism.

Course Contents:
01. Introduction, Compiling & executing a java program.
02. Program with data types & variables.
03. Program with decision control structures: if, nested if etc.
04. Program with loop control structures: do, while, for etc.
05. Program with classes and objects.
06. Implementing data abstraction & data hiding.
07. Implementing inheritance.
08. Implementing and polymorphism.
09. Implementing packages.
10. Implementing generics.
11. Implementing Collections
12. I/O in Java
13. Using JDBC to Connect to a Database (Oracle/MySQL)
14. Using Apache Tomcat Server
15. Using Servlets

CA-307 Lab-6: Machine Learning

Prerequisites:

- Basic Python programming
- Knowledge of linear algebra and statistics
- Familiarity with data structures and algorithms

Learning Objectives:

- Understand core ML algorithms and concepts
- Implement supervised and unsupervised models
- Evaluate model performance using metrics
- Apply neural networks and deep learning basics

Learning Outcomes:

- Preprocess and analyze datasets for ML
- Build and evaluate classification and regression models
- Apply clustering, SVM, and neural networks
- Use Python libraries like scikit-learn and TensorFlow

Course Content:

1. Write a program to load a CSV dataset (e.g., Iris or Titanic), perform data cleaning (handling missing values), encoding categorical features, scaling, and split it into training and testing sets using train_test_split().
2. Write a program to implement a Decision Tree classifier using scikit-learn on a real dataset (e.g., Iris or Titanic), visualize the tree, calculate accuracy, and explain entropy and information gain.
3. Write a program to predict house prices using multiple linear regression on the Boston Housing dataset. Plot regression lines, visualize residuals, and compute MSE and RMSE.
4. Write a program to classify whether a person has diabetes or not using logistic regression on the Pima Indians Diabetes dataset. Show decision boundary and compute performance metrics.
5. Write a program to implement a Perceptron classifier for binary classification using NumPy (from scratch), then compare it with MLPClassifier from sklearn.neural_network on the Iris dataset.
6. Write a program using scikit-learn's MLPClassifier to classify handwritten digits from the MNIST dataset. Experiment with different activation functions, hidden layers, and solvers.
7. Write a program to apply SVM on a binary classification dataset (e.g., Breast Cancer dataset), try linear and RBF kernels, and plot the decision boundary for 2D features.
8. Write a program to train a classifier on any binary classification dataset and evaluate it using confusion matrix, accuracy, precision, recall, F1-score, and ROC-AUC curve.
9. Write a program to perform 10-fold cross-validation on a regression task using scikit-learn and report average MSE and RMSE. Use the California Housing dataset.
10. Write a program to classify text data (e.g., spam detection) using MultinomialNB and GaussianNB, compare their performance using accuracy and confusion matrix.
11. Write a program to classify Iris flowers using the KNN algorithm for k = 1 to 10 and plot accuracy vs. k. Use both Euclidean and Manhattan distance.
12. Write a program to cluster customer data using K-Means, Agglomerative Clustering, and DBSCAN. Visualize clusters using PCA (for dimensionality reduction).
13. Write a program using TensorFlow/Keras to classify digits using a simple CNN on the MNIST dataset. Include training and validation plots, and test accuracy.
14. Write a program using TensorFlow/Keras to build an RNN for text sentiment classification on a dataset like IMDB. Preprocess text using tokenization and padding.
15. Write a program to optimize the function f(x) = x * sin(x) using a Genetic Algorithm. Demonstrate selection, crossover, mutation, and visualize the fitness landscape over generations.

CA-304-A Combinatorics and Graph Theory

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Basic understanding of discrete mathematics and algorithms.

Learning Objectives:
1. Introduce fundamental concepts of graph theory and combinatorics.
2. Explore graph algorithms, connectivity, planarity, and coloring problems.
3. Understand combinatorial counting techniques and Polya's theorem.

Learning Outcomes:
1. Analyze graphs, spanning trees, and shortest path algorithms.
2. Apply concepts of connectivity, separability, and isomorphism in graphs.
3. Solve problems related to planar graphs and graph coloring.
4. Use combinatorial methods to count labeled and unlabeled trees.
5. Apply Polya's theorem to solve enumeration problems in graphs.

Course Contents:

Unit 1
Graph, Sub Graph, Connected component , spanning trees , Shortest path algorithm, Cut-sets and Cut-Vertices: Cut-Sets, Properties of Cut-sets, All Cut-sets in a Graph, Fundamental Circuits and Cut-Sets.

Unit 2
Connectivity and Separability, Network flows, 1-isomorphism, 2-isomorphism. Combinatorial Vs. Geometric Graphs, Planar Graphs, Kuratowski's Two Graphs, Different Representations of a Planar Graph, Detection of planarity, Geometric Dual, Combinatorial Dual, Matrix Representation of Graphs.

Unit 3
Chromatic Number, Chromatic Partitioning, Chromatic, Polynomial, Vertex Cover, Matchings, Pathcover, Connectivity, Hamiltonicity, Vertex Coloring, Edge Coloring, Four Color Problem and Other Coloring Problems.

Unit 4
Basic combinatorial numbers, recurrences, generating functions, Latin squares, partitions, partially orders sets, Types of Enumeration, Counting Labeled Trees, Counting Unlabeled Trees, Polya's Counting Theorem, Graph Enumeration with Polya's Theorem.

Reference Books:
1. Graph Theory with Applications to Engineering and Computer Science, Narasingh Deo, PHI.
2. Graph Theory and Combinatorics, Dr. D.S. Chandrasekharaiah, Prism, 2005.
3. Introduction to Graph Theory, Chartrand Zhang, TMH, 2006.
4. Graph Theory Modeling, Applications, and Algorithms, Geir Agnarsson & Raymond Geenlaw,Pearson Prentice Hall, 2007.
5. A Walk Through Combinatorics by Miklos Bona, World Scientific Publishing Company, 2002.
6. Concrete Mathematics: A Foundation for Computer Science, by Ronald Graham, Donald Knuth, and Oren Patashnik.
7. A Course in Combinatorics, J. H. van Lint, R. M. Wilson Cambridge University Press, 1993.
8. Combinatorial Methods with Computer Applications Jonathan L. Gross.
9. Graph Theory and Its Applications, Second Edition Jonathan L. Gross, Jay Yellen

CA-304-B Data Warehousing and Data Mining

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Knowledge of DBMS

Learning Objectives:
1. Be familiar with mathematical foundations of data mining tools.
2. Understand and implement classical models and algorithms in data warehouses and data mining.
3. Characterize the kinds of patterns that can be discovered by association rule mining, classification and clustering.
4. Master data mining techniques in various applications like social, scientific and environmental context.
5. Develop skill in selecting the appropriate data mining algorithm for solving practical problems.

Learning Outcomes:
1. Understand the functionality of the various data mining and data warehousing components.
2. Appreciate the strengths and limitations of various data mining and data warehousing models.
3. Explain the analyzing techniques of various data.
4. Describe different methodologies used in data mining and data warehousing.
5. Compare different approaches of data warehousing and data mining with various technologies.

Course Content:
Unit 1
Concept of Data warehousing, 3-tier architecture, multidimensional data model, OLAP, ROLAP, and MOLAP operations. Commercial Importance of Data Warehouse, Data Mart structure, Usage of Data Mart, Security in Data Mart, Data warehouse and Data Mart.

Unit 2
Basic Elements of Data Warehouse & ETL: Source System, Data Staging Area, Presentation Server, data Cleaning, Extraction of Data, Transformation of Data, Loading of Data

Unit 3
Introduction to data mining, knowledge discovery, DBMS vs. Data Mining. Concept hierarchies, Interestingness measures, Data generalization and Summarization-based characterization, Mining Association Rules, Apriori algorithm for finding frequent item-sets, Mining Multilevel Association Rules, Mining distance-based Association Rules, Correlation Analysis.

Unit 4
Classification and prediction: decision tree based classification, Bayesian classification, classification by back propagation, k-nearest neighbor classifier. Cluster analysis: categorization of clustering methods, partitioning methods, k-Means and k-Medoids, hierarchical methods, Density-based clustering (DBSCAN)

Reference Books:

1. Jain Pei, Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques", 3rd Edition, Elsevier/Morgan Kaufmann
2. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, "Introduction to Data Mining", 1st Edition, Pearson India
3. A.K.Pujari, "Data Mining", University Press
4. Ian H. Witten, Eibe Frank, "Data Mining Practical Machine Learning Tools and Techniques", 2nd Edition, Elsevier/Morgan Kaufmann
5. Ralph Kimball , Margy Ross, "The Data Warehouse Toolkit : The Definitive Guide to Dimensional Modeling", 3rd Edition, Wiley India
6. Paulraj Ponniah, "Data Warehousing: Fundamentals for IT Professionals", 2nd Edition, Wiley India

S. Anahory, D. Murray, "Data Warehousing", Pearson

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

CA-304-C Computer Graphics

Prerequisites: Basic geometry and programming skills.

Learning Objectives:

1. The main objective of the course is to introduce students with fundamental concepts and theory of computer graphics.
2. It presents the important drawing algorithm, polygon fitting, clipping and 2D transformation curves and an introduction to 3D transformation.
3. It provides the basics of OpenGL application programming interface which allows students to develop programming skills in CG.

Learning Outcomes:

After successful completion of the course, student shall be able to:

1. Explain the core concepts of computer graphics, including viewing, projection, perspective, modelling and transformation in two and three dimensions.
2. Apply the concepts of colour models, lighting and shading models, textures, ray tracing, hidden surface elimination, anti-aliasing, and rendering.
3. Interpret the mathematical foundation of the concepts of computer graphics.
4. Describe the fundamentals of animation, parametric curves and surfaces, and spotlighting.
5. Identify a typical graphics pipeline and apply graphics programming techniques to design and create computer graphics.
6. Create effective OpenGL programs to solve graphics programming issues, including 3D transformation, objects modelling, colour modelling, lighting, textures, and ray tracing.

Course Content:

Unit 1

Introduction: Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices. Output primitives**:** Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms.

Unit 2

2-D and 3D geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems. 3D viewing and General projection transforms (Parallel & Perspective Projections and their Transformations)

Unit 3

Viewing and Clipping: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Liang-Barsky line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm.

Unit 4

Bezier curve and B-Spline curves, Bezier and B-Spline surfaces. Basic illumination models, polygon rendering methods. Illumination models: Basic Models, Displaying Light Intensities, halftone patterns and Dithering Techniques Visible surface detection methods: Classification, back-face detection, depth-buffer, scan-line, depth sorting. Surface Rendering Methods: Polygon Rendering Methods, Gouraud shading Phong Shading

Text Books:

1. Donald Hearn & M. Pauline Baker, “Computer Graphics with OpenGL”, Third Edition, 2004, Pearson Education, Inc. New Delhi.
2. Ze-Nian Li and Mark S. Drew, “Fundamentals of Multimedia”, First Edition, 2004, PHI Learning Pvt. Ltd., New Delhi.
3. Jennifer Burg, “The Science of Digital Media”, First Edition, 2009, Pearson Education Inc., New Delhi.
4. Francis S. Hill & Stephen M. Kelly, “Computer Graphics using OpenGL”, PHI, 3rd Edition

CA-304-D Internet of Things

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Basic networking, embedded systems, and Python.
Familiarity with cloud and communication protocols.

Learning Objectives:
1. Grasp IoT concepts, technologies, and protocols.
2. Develop IoT systems and apply cloud services.
3. Implement IoT for real-world scenarios.

Learning Outcomes:
1. Design IoT solutions with relevant protocols.
2. Use cloud platforms and develop practical IoT applications.

Course Content:

Unit 1
Introduction, Design and Technologies: Introduction- Definition & Characteristics of IoT –Physical design of IoT-Things in IoT and IoT protocols, Logical Design of IoT- IoT Functional Blocks-IoT Communication Model and IoT Communication APIs - IoT Enabling Technologies - Wireless Sensor Networks -Cloud Computing- Big Data Analytics - Communication Protocols Embedded Systems-IoT Levels & Deployment Templates.

Unit 2
Domain, M2M and System Management : Introduction- Home Automation – Cities - Industry- Health & Lifestyle- M2M-SDN and NFV for IoT - Software Defined Networking - Network Function Virtualization- IoT System Management- Need for IoT Systems Management -Simple Network Management Protocol -Limitations of SNMP - Network Operator-Requirements

Unit 3
Protocols and Developing Internet Of Things: Infrastructure - 6LowPAN- IPv6, Identification - EPC-uCode-URIs,Comms / Transport - Wifi, Bluetooth- LPWAN,Discovery- Physical Web- mDNS-DNS-SD, IoT Platforms Design Methodology - IoT System for Weather Monitoring – IoT System for Agriculture.

Unit 4
Cloud Services for IOT: Introduction to Cloud Storage Models & Communication APIs -WAMP - AutoBahn for IoT-Xively Cloud for IoT -Python Web Application Framework, Case studies – Environment- IoT systems for weather Reporting Bot- Air Pollution Monitoring System-Forest Fire Detection-IoT system for Energy-Smart grid-Renewable Energy Systems.

REFERENCES

1. ArshdeepBahga and Vijay Madisetti, ―Internet of Things - A Hands-on Approach", Universities Press, 2015
2. Dieter Uckelmann et.al, ―Architecting the Internet of Things", Springer, 2011.
3. CunoPfister, ―Getting Started with the Internet of Things", O'Reilly, 2011.
4. Adrian McEwen, Hakim Cassimally, ―Designing the Internet of Things", Wiley, 2014.
5. Honbo Zhou , ―The Internet of Things in the Cloud: A Middleware Perspective ―, CRC Press , 2012
6. Olivier Hersent, David Boswarthick, Omar Elloumi, ―The Internet of Things – Key applications and Protocols", Wiley, 2012.

CA-401-A Artificial Intelligence

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Pre-requisites:
Logic, Data Structures

Objectives:
Understand and define AI; Design and implement AI programs / applications

Learning Outcomes: A student who takes this course should be able to
1. Compare the various heuristic algorithms
2. To apply the above to solve NP hard problems depending upon their suitability
3. Represent knowledge in a retrievable form
4. Design NL interfaces
5. Program in LISP / PROLOG/ Python / MATLAB

Course Contents:

Unit 1

Problems and Search: What is Artificial Intelligence?, The AI Problems, The Underlying Assumption, What is an AI Technique, The Level of the Model, Criteria for Success, Problems, Problem Spaces, and Search: Defining the Problem as a State Space Search, Problem Characteristics.

Heuristic Search Techniques: Generate-and- Test, Hill Climbing, Best- First Search, Problem Reduction, Constraint Satisfaction, A* algorithm.

Unit 2

The Prepositional Calculus : Introduction – Syntax of the Prepositional Calculus – Truth-Assignments – Validity and Satisfiability – Equivalence and Normal Forms – resolution in Prepositional Calculus.

Using Predicate Logic:- Representing Instance and ISA Relationships, Computable Functions and Predicates, Resolution ; Truth Maintenance Systems

Unit 3

Historical notes- human Brain, Al and NN. Learning process: Supervised and unsupervised learning, adaptation, statistical nature of the learning process. Perceptron.

Multilayered network architecture, Back Propagation Algorithm, Approximation properties of RBF networks and comparison with multilayer perceptron.

Unit 4

Introduction to AI Programming: PROLOG, LISP, MATLAB, WEKA Tool

Reference Books:

1. Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Third Edition, 2010, Pearson Education, New Delhi.
2. Joseph Giarratano and Gary Riley, “Expert Systems: Principles and Programming”, Fourth Edition, CENGAGE Learning India Pvt. Ltd., New Delhi.
3. Elaine A. Rich and Kevin Knight, “Artificial Intelligence”, 3rd Edition, 2009, McGraw-Hill Education (India), New Delhi.
4. Nills J. Nilsson, “Artificial Intelligence: A New Synthesis”, 2nd Edition, 2000, Elsevier India Publications, New Delhi.
5. Michael Negnevitsky, “Artificial Intelligence: A Guide to Intelligent Systems”, Second Edition, 2005, Pearson Education, Inc. New Delhi.
6. Dan W. Patterson, “Introduction to Artificial Intelligence and Expert Systems”, 1st Edition, 1996, PHI Learning Pvt. Ltd., New Delhi.
7. Ben Coppin, “Artificial Intelligence Illuminated”, 2005, Narosa Publication, New Delhi.

CA-401-B Block Chain Technology

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisite
Basic understanding of computer networks and cryptography.

Learning Objectives
1. Understand blockchain technology, decentralization, and cryptocurrencies.
2. Explore blockchain platforms, architecture, and applications.

Learning Outcomes
1. Explain blockchain fundamentals, smart contracts, and cryptocurrency concepts.
2. Apply blockchain platforms like Ethereum to design decentralized applications.

Course Content:

Unit 1
Discover Blockchain Technology: Blockchain, Growth of blockchain technology, Distributed systems, History of blockchain and Bitcoin, Types of blockchain. Decentralization: Methods of decentralization, Routes of decentralization, Blockchain and full ecosystem decentralization, Smart contracts, Decentralized organizations and platforms for decentralization.

Unit 2
Blockchain: Architecture, Versions, Variants, Use cases, Life use cases of blockchain, Blockchain vs shared Database, Introduction to cryptocurrencies, Types, Applications. Bitcoins: Introducing Bitcoin, Bitcoin digital keys and addresses, Transactions, Blockchain mining. Alternative Coins. Limitations of Bitcoin.

Unit 3
Concept of Double Spending, Hashing, Proof of work. Bitcoin Network and payments, Bitcoin network, Wallets, Bitcoin payments, Innovation in Bitcoin, Bitcoin Clients and APIs.

Unit 4
Introduction to Blockchain Platforms: Ethereum, Hyperledger, IOTA, EOS, Multichain, Bigchain, etc. Advantages and Disadvantages, Ethereum vs Bitcoin, Design a new blockchain, Potential for disruption, Design a distributed application, Blockchain applications.

Reference Books

1. Mark Gates, “Block chain: Ultimate guide to understanding block chain, bit coin, crypto
1. currencies, smart contracts and the future of money”, Wise Fox Publishing and Mark Gates 2017.
2. Salman Baset, Luc Desrosiers, Nitin Gaur, Petr Novotny, Anthony O'Dowd, Venkatraman
3. Ramakrishna, “Hands-On Block chain with Hyper ledger: Building decentralized applications
4. with Hyperledger Fabric and Composer”, 2018.
3. Bahga, Vijay Madisetti, “Block chain Applications: A Hands-On Approach”, Arshdeep Bahga,
5. Vijay Madisetti publishers 2017.
4. Andreas Antonopoulos, “Mastering Bitcoin: Unlocking Digital Crypto currencies”, O'Reilly
6. Media, Inc. 2014.
5. Melanie Swa, “Block chain ”,O'Reilly Media 2014.

CA-401-C Mobile Computing

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Pre-requisites: Basic computer network

Learning Objectives:
1. learn about the concepts and principles of mobile computing;
2. Explore both theoretical and practical issues of mobile computing;
3. Develop skills of finding solutions and building software for mobile computing applications.

Learning Outcomes: A student who takes this course should be able to
1. Define mobile technologies in terms of hardware, software, and communications.
2. Utilize mobile computing nomenclature to describe and analyze existing mobile computing frameworks and architectures.
3. Evaluate the effectiveness of different mobile computing frameworks.
4. Describe how mobile technology functions to enable other computing technologies

Course Content:

Unit 1

Introduction to Mobile Communications and Computing**:** Introduction to Mobile Computing, novel applications, limitations, and architecture. GSM: Mobile services, System architecture, Radio interface, Protocols, Localization and calling, Handover, Security, and New data services.

Unit 2

Wireless Medium Access Control**:** Motivation for a specialized MAC (Hidden and exposed terminals, Near and far terminals), SDMA, FDMA, TDMA, CDMA. Spreading techniques.
Mobile Network Layer: Mobile IP (Goals, assumptions, entities and terminology, IP packet delivery, agent advertisement and discovery, registration, tunneling and encapsulation, optimizations).

Unit 3

Mobile Transport Layer: Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP. Transaction oriented TCP. Database Issues: Hoarding techniques, caching invalidation mechanisms, client server computing with adaptation. Data Dissemination: Communications asymmetry, classification of new data delivery mechanisms, push based mechanisms, pull-based mechanisms.

Unit 4

Mobile Ad hoc Networks (MANETs): Overview, Properties of a MANET, routing and various routing algorithms, security in MANETs. Protocols and Tools: Wireless Application Protocol-WAP. (Introduction, protocol architecture, and treatment of protocols of all layers), Bluetooth (User scenarios, physical layer, MAC layer, networking).

Reference Books:

1. Jochen Schiller, Mobile Communications, Pearson Education.
2. Stojmenovic and Cacute, Handbook of Wireless Networks and Mobile Computing, Wiley.
3. Reza Behravanfar, Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML, Cambridge University Press.
4. Frank Adelstein, Sandeep K.S Gupta, Golden Richard III, Loren Schwiebert, Fundamentals of Mobile and Pervasive Computing, TMH.
5. Uwe Hansmann, Lothar Merk, Martin S, Nicklous, Thomas Stober, Principles of Mobile Computing, Springer.
6. Martyn Mallick, Mobile and Wireless Design Essentials, Wiley.

CA-401-D Internet & Web Technology

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisite
Basic knowledge of networks and programming.
Familiarity with HTML and web basics.

Learning Objectives
1. Understand Internet architecture, protocols, XML, and JavaScript.
2. Apply JavaScript, JDBC, Servlets, and JSP in web development.

Learning Outcomes
1. Explain TCP/IP, HTTP, XML, and JavaScript concepts.
2. Develop dynamic web applications using JavaScript, JDBC, Servlets, and JSP.

Unit 1
Origin of internet. Key elements of Internet. Internet Architecture. OSI Reference Model. TCP/IP - Overview of TCP/IP, Layers in TCP/IP, Network Access Layer, Internet Layer: Internet Protocol, Transport Layer: User Datagram Protocol, Transmission Control Protocol, Application Layer. HTTP - Overview of HTTP: Basic protocol, HEAD, GET, POST, PUT, DELETE.

Unit 2
XML: Document structure, DTD, Namespaces, XML Schema, Parsing XML documents.
JavaScript - Language basics: Primitive & Composite Datatypes, Variables. Operators, Expressions and Statements. Functions: Global & Local variables, Functions as objects, recursive functions, Objects: Object fundamentals, Generic and user defined objects, Built-in Objects: Boolean, Date, Global, Math, Number, String

Unit 3
Applying JavaScript - The Document Object, The Standard Document Object Model, Event Handling, Applying JavaScript:Controlling windows, Handling Documents, Form Handling

Unit 4
The Java Database Connectivity(JDBC) Overview, JDBC Drivers, Loading JDBC drivers, Using JDBC Classes like Connection, Statement, PreparedStatement ,and ResultSet. Java Servlets: Life cycle of a Servlet; Using the Servlet APIs. JSP Technology: - Architecture & Anatomy of JSP Page, JSP Predefined variables/objects, JSP with MVC Architecture.

Reference Books:

1. Andrew S.Tanenbaum, "Computer Networks", 4th Edition, Pearson Education Asia.
2. Douglas E. Comer, "Internetworking with TCP/IP, Volume 1: Principles, Protocols and Architecture", 5th Edition, PHI Learning.
3. Thomas A. Powell, "HTML & CSS: The Complete Reference", 5th Edition, McGraw-Hill
4. Elisabeth Robson, Eric Freeman, "Head First HTML and CSS", 2nd Edition, O'Reilly
5. Ralph Moseley, "Developing Web Applications", 2008, Wiley India
6. Xavier C, "Web Technology and Design", New Age International
7. Thomas Powell and Fritz Schneider, "JavaScript 2.0: The Complete Reference", 2nd Edition, McGraw-Hill/Osborne
8. ECMAScript Language Specification -Standard ECMA-262 3rd Edition - December 1999
9. David Flannagan, "JavaScript: The Definitive Guide", 2nd Edition, O'Reilly
10. Joe Wigglesworth, Paula McMillan, "Java Programming: Advanced Topics", 3rd Edition, CENGAGE Learning
11. Phillip Hanna , "JSP 2.0: The Complete Reference", 2nd Edition, McGraw Hill Education
12. Kongent S., "Java Server Programming (JEE 6) Black Book", Platinum Edition", Dreamtech / Wiley India.
13. Bryan Basham and Kathy Sierra, "Head First Servlets and JSP", 2nd Edition, O'Reilly
14. Brett McLaughlin, "Java & XML", 2nd Edition, O'Reilly

CA-402-A Digital Image Processing

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisite
Basic knowledge of linear algebra, probability, and calculus.

Learning Objectives
1. Understand the fundamentals of digital images and their properties.
2. Learn image preprocessing techniques and transformations.
3. Explore image segmentation methods and edge detection techniques.
4. Study mathematical morphology for image analysis.
5. Understand image texture analysis and representation techniques.

Learning Outcomes
1. Explain the key concepts of image digitization and properties.
2. Apply image preprocessing methods like brightness and geometric transformations.
3. Perform image segmentation using thresholding and edge detection.
4. Use mathematical morphology techniques for image analysis and segmentation.
5. Analyze and describe images using texture and boundary descriptors.

Course Content:

Unit 1
Introduction: The digitized image and its properties: Applications of image processing, image function, image representation, sampling, quantization, color images, metrics and topological properties of digital images, histograms, image quality, noise image.

Unit 2
Image preprocessing: Pixel brightness transformation, position dependent brightness correction, gray scale transformation; geometric transformation, local preprocessing- image smoothening, edge detectors, zero-crossing, scale in image processing, canny edge detection, parametric edge models, edges in multi spectral images, local preprocessing and adaptive neighborhood pre processing; image restoration.

Unit 3
Image Segmentation: Threshold detection methods, optimal thresholding, multispectral thresholding, thresholding in hierarchical data structures; edge based image segmentation, edge image thresholding, edge relaxation, border tracing, border detection.

Unit 4
Mathematical Morphology: Basic morphological concepts, four morphological principles, binary dilation, erosion, Hit or miss transformation, opening and closing; thinning and skeleton algorithms; Morphological segmentation -particles segmentation and watersheds, particles segmentation.
Image textures: Statistical texture description, methods based on spatial frequencies, cooccurrence matrices, edge frequency, and texture recognition method applications. Image representation and description: Representation, boundary descriptors, regional descriptors

Reference Books:

1. Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 3rd Edition, 2008, Pearson Education, Inc. New Delhi.
2. Milan Sonka, Vaclav Hlavac, Roger Boyle, “Image Processing, Analysis, and Machine Vision”, 3rd Edition, 2008, CENGAGE Learning, New Delhi.
3. William K. Pratt, “Digital Image Processing: PIKS Scientific Inside”, 4th Edition, 2008, Wiley India Pvt. Ltd., New Delhi.
4. Bernd Jähne, “Digital Image Processing”, 6th Revised and Extended Edition, 2006, Springer Science. ISBN 978-3-540-24035-8.
5. Anil K. Jain, “Fundamentals of Digital Image Processing”, 1989, PHI Learning Pvt. Ltd.

CA-402-B Cloud Computing

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Basic knowledge of programming, networks, OS, and distributed systems.

Learning Objectives:
1. Understand cloud computing and virtualization basics. Learn cloud service models and providers.
2. Explore cloud architecture and security challenges. Utilize cloud services for collaboration.

Learning Outcomes:
1. Explain virtualization and cloud concepts. Compare cloud models (public, private, hybrid).
2. Apply cloud architecture and resource management. Implement cloud security measures.
3. Use cloud tools for communication and task management.

Course Contents:
Unit 1 INTRODUCTION TO VIRTUALIZATION: Overview of Centralized, Distributed Computing -Cluster computing, Grid computing. Technologies for Network based systems - Virtual Machines and Virtualization Middleware – Data Center Virtualization for Cloud Computing – Implementation Levels of Virtualization – Virtualization Structures/Tools and Mechanisms – Virtualization of CPU – Memory – I/O Devices.

Unit 2 INTRODUCTION TO CLOUD COMPUTING: System Models for Distributed and Cloud Computing – Software Environments for Distributed Systems and Clouds – Cloud Computing and Service Models – Public – Private – Hybrid Clouds – Infrastructure-as-a-Service (IaaS) – Platform-as-a Service (PaaS) - Software-as-a-Service (SaaS)-Different Service Providers

Unit 3 CLOUD ARCHITECTURE AND RESOURCE MANAGEMENT: Architectural Design of Compute and Storage Clouds – Public Cloud Platforms: – Amazon Web Services(AWS) – Azure, Emerging Cloud Software Environments– Open Stack – Extended Cloud Computing Services – Resource Provisioning and Platform Deployment – Virtual Machine Creation and Management. CLOUD PROGRAMMING: Parallel Computing and Programming Paradigms – Map Reduce – Twister – Iterative Map Reduce – Current technologies – Emerging Cloud software Environment.

Unit 4 SECURITY IN THE CLOUD Security Overview – Cloud Security Challenges – Security -as-a Service – Security Governance – Risk Management – Security Monitoring – Security Architecture Design – Data Security – Application Security – Virtual Machine Security.

Reference Books:

1. Kai Hwang , Geoffrey C Fox, Jack J Dongarra : “Distributed and Cloud Computing – From Parallel Processing to the Internet of Things” , Morgan Kaufmann Publishers – 2012
2. RajkumarBuyya, Christian Vecchiola, and ThamaraiSelvi“Mastering Cloud Computing” McGraw Hill
3. Alex Amies, Harm Sluiman, QiangGuo Tong and Guo Ning Liu: “Developing and Hosting Applications on the cloud”, IBM Press, 2012.
4. George Reese, “Cloud Application Architectures: Building Applications and Infrastructure in the Cloud (Theory in Practice)”, O’Reilly Publications, 2009.
6. Haley Beard, “Cloud Computing Best Practices for Managing and Measuring Processes for On-demand Computing – applications and Data Centers in the Cloud with SLAs”, Emereo Pty Limited, July 2008
7. James E. Smith and Ravi Nair: Virtual: Machines: Versatile Platforms for Systems and Processes, Morgan Kaufmann, ELSEVIER Publication, 2006.
8. John W Rittinghouse and James F Ransome , “Cloud Computing: Implementation – Management – and Security”, CRC Press, 2010.
9. Michael Miller, “Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online”, Pearson Education, 2009.

CA-402-C Data Science using Python

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Reasoning and Logical Skills

Learning Objectives:
The objective of this course is to:

1. To acquire in-depth understanding of the theoretical concepts in statistics, data analysis, data mining, machine learning and other advanced data science techniques.
2. To empower students with tools and techniques for handling, managing, analyzing and interpreting data.
3. Ability to understand the abstract concepts that lead to various data science theories in Mathematics, Statistics and Computer science.
4. To identify, analyze and design solutions for data science problems using fundamental principles of mathematics, Statistics, computing sciences, and relevant domain disciplines.

Learning Outcomes:
After successful completion of the course, student shall be able to:

1. Understand the key concepts in data science.
2. Demonstrate an understanding of statistics and machine learning concepts that are vital for data science.
3. Produce Python code to statistically analyse a dataset.
4. Critically evaluate data visualisations based on their design and use for communicating stories from data.
5. Plan and generate visualisations from data using Python.

Course Content

Unit 1
Python basics: Modules, Arithmetic, Control Flow, Functions, Strings, Exceptions, Lists, Tuples, Dictionaries, Sets

Unit 2
Data Visualization with matplotlib. Hypothesis and Inference: Statistical Hypothesis Testing, Confidence Intervals, P-hacking. Gradient Descent: The idea behind Gradient Descent, Estimation of Gradient, Applying Gradient Descent. Machine Learning: Modeling, types, Overfitting and Underfitting, Correctness, The Bias-Variance Trade-off.

Unit 3
k-Nearest Neighbors: The Model, Application, The Curse of Dimensionality. Implementing a spam filter with Naive Bayes. Correlation and Regression. Simple Linear Regression: The Model, Using Gradient Descent, Maximum Likelihood Estimation. Logistic Regression: Applying the Logistic Function model, Goodness of Fit.

Unit 4
Decision Trees: Decision Tree, Entropy. Neural Networks: Perceptrons, Feed-Forward Neural Network, Backpropagation. k-Means Clustering.

Reference Books:

1. Joel Grus. ―Data Science from Scratch: First Principles With Python‖, O‘Reilly
2. Jake vanderplas. ―Python data science handbook: essential tools for working with data‖, O‘Reilly
3. Davy Cielen, Arno D. B. Meysman, Mohamed Ali. "Introducing Data Science", Manning Publication
4. Cathy O'Neil and Rachel Schutt. "Doing Data Science, Straight Talk From The Frontline". O'Reilly.
5. Sinan Ozdemir, "Principles of Data Science", Packt

CA-402-D Compiler Design

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Pre-requisites:
Mathematics, Automata Theory, Operating Systems and Computer Programming.

Learning Objectives:
To teach the students about the phases of designing a Compiler and describe the various data structures those can be used to represent the phases of the Compiler.

Learning Outcomes:
On completion of the course, the student should be able to:
1. structure a compiler as a sequence of distinct translation steps
2. use regular languages to describe the lexical elements of a programming language
3. describe lexical analysis using a finite automaton
4. use context free languages to describe the syntactic structure of a programming language
5. use the parsing methods top-down (recursive descent) and bottom-up (LR)
6. use abstract syntax trees to represent the results of the syntactic analysis
7. break down statements and expressions to simpler designs, and translate syntax trees to intermediate code
8. describe how recursive procedure calls can be implemented by means of stacks, activation posts and machine registers
9. translate the simplified intermediate code of a program to machine-specific instructions

Course Content:

Unit – I:
Introduction to Compilers : Phases of a Compiler - Syntax Definition – Context Free Grammars – Top Down Parsing – Bottom up Parsing – Predictive Parsing – Operator Precedence Parsing – LR Parser – SLR Parsing

Unit – II:
Program representation in Abstract Syntax Trees (AST); Symbol tables and scope rules for C-like languages; Type-checking for C-like languages

Unit-III:
Intermediate Code Generation - Different forms of intermediate code (IR); Generation of intermediate code; Call stacks and activation posts

Unit-IV:
Code Optimization and Generation for RISC-like machines; Basic blocks, control-flow graphs, likeness analysis, register allocation

Reference Books:

1. John J. Donovan ―System Programming , Tata Mc Graw- Hill.
2. A.V. Aho Sethi, J.D. Ullman, ―Compiler Principles, Techniques and Tools, Addison Wesley Publishing Company, 1986.
3. Dhamdhare D., ―System Programming and Operating System , Tata Mc Graw-Hill.

CA-403-A Organizational Behavior

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisites:
Basic understanding of management principles and human behavior.

Learning Objectives:
1. Introduce key concepts of organizational behavior and its historical development.
2. Explore individual and group dynamics in the workplace.
3. Understand personnel management functions and strategies.
4. Examine leadership, conflict resolution, and organizational culture.

Learning Outcomes:
1. Explain the evolution and importance of organizational behavior.
2. Analyze personality, job attitudes, and motivation theories.
3. Apply personnel management techniques in recruitment, training, and performance appraisal.
4. Understand group behavior, leadership theories, and conflict resolution strategies.
5. Describe organizational structures, culture, and change management models.

Course Content:

Unit 1
Basics of OB—History of OB: Stages of development- Pre-Scientific, Classical, Behavioural, Modern Organization structure & process: Components and their interaction Approaches: Traditional & Modern: Contributing disciplines Emerging Issues: Globalization, Diversity, Demographics, Ethical behaviour

Unit 2
The Individual—Personality: Factors (Big Five), Attributes, Measurement (Myers-Briggs Type Indicator); Job Attitude: Components, Major attitudes, Job satisfaction & its measurement, Dissatisfaction; Motivation: Early theories (Need hierarchy & Two-factor) & Recent Theories (ERG & Expectancy)

Unit 3
Personnel Management—Personnel Function: Its evolution, objectives, principles, philosophies and policies, duties and responsibilities of the personnel manager, position of the personnel department on the organization, Line and staff relationship. Manpower planning: Its uses and benefits, problems. limitations, manpower skill analysis and practices. Recruitment: Job specification, selection process, psychological testing, interviewing techniques, transfers, promotion and its policies. Training and Developments: Its objectives and places, planning and organizing then training department, training manager and his job. On and off the job training, techniques, career planning, objectives of Performance appraisal and its methods.

Unit 4
The Group—Groups: Types, Roles, Norms, Size, Group vs. Team, Cohesiveness, Group Decision Making Leadership: Theories (Trait, Behavioural & Contingency), Finding and Creating Effective Leaders. Conflict: Approach, Sources & Negotiation: Stages in the process and remedial strategies; The Organization—Structure: Basics, Common Designs: Simple, Bureaucratic and Matrix structures Culture: Basics, Functions: Boundary, Identity, Commitment & Stability and Management Change: Agents, Resistance, Management: Lewin's 3-step and Kotter's 8-step models

Reference Books:

1. Robbins, Sanghi & Judge, Essentials of Organisational Behaviour, PHI
2. Kavita Singh, Organisational Behaviour :Text and cases, Pearson
3. Fiona Wilson, Organisational Behaviour and Work, Oxford.

CA-403-B Web and Text Mining/ Text Analysis

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Pre-requisites: Linear algebra, Discrete Mathematics, Statistics, Computer Science basics

Learning Objectives:

1. Learn both the theoretical and practical aspects of web mining and text mining
2. Learn fundamental of Information Retrieval and Natural Language Processing
3. Explore the domain of Social Network Analysis

Learning Outcomes: A student who takes this course should be able to

1. Design and implement a crawler application to collect and index documents from the web.
2. Analyze text to determine the reliability of the information including potential bias.
3. Evaluate and apply key analytics techniques used in natural language processing and text retrieval
4. Interpret the results, gain insights, and recommend possible actions from analytics performed on text data.
5. Explore career path as Data Scientist, Business Analyst, and Data Analyst.

Course Content:

Unit 1:

Basic Concepts of Information Retrieval, Information Retrieval Models: Boolean Model, Vector Space Model, Statistical Language Model. N-Gram Language Model: Simple (Unsmoothed) N-grams, Smoothing: Add-one Smoothing, Witten Bell Discounting, Good Turing Discounting.

Unit 2:

Text and Web Page Pre-Processing: Stopword Removal, Stemming, Web Page Pre-Processing, Duplicate Detection. Inverted Index and Its Compression: Inverted Index, Search Using an Inverted Index, Index Construction, Index Compression.

Unit 3:

Web Search: Searching and Ranking. Web Crawling, Basic Crawler Algorithm: Breadth-First Crawlers, Preferential Crawlers, Craweler Implementation Issues: Fetching, Parsing, Stopword Removal and Stemming, Link Extraction and Canonicalization, Spider Traps.

Unit 4:

Introduction to Social Network Analysis, Social Networks Preliminaries and Properties: Homophily, Triadic Closure and Clustering Coefficient, Dynamics of Network Formation, Power-Law Degree Distributions, Measures of Centrality and Prestige: Degree Centrality and Prestige, Closeness Centrality and Proximity Prestige, Betweenness Centrality, Rank Centrality and Prestige.

Reference Books

1. Liu, Bing. Web data mining: exploring hyperlinks, contents, and usage data. Vol. 1. Berlin: springer, 2011.
2. Jurafsky, Dan. Speech & language processing. Pearson Education India, 2000.
3. Aggarwal, Charu C. Data mining: the textbook. Vol. 1. New York: springer, 2015.
4. Christopher, D. Manning, Raghavan Prabhakar, and Schutze Hinrich. An Introduction to information retrieval. Cambridge University Press, 2008.
5. Chakrabarti, Soumen. Mining the Web: Discovering knowledge from hypertext data. Morgan Kaufmann, 2002.
6. Zong, Chengqing, Rui Xia, and Jiajun Zhang. Text Data Mining. Vol. 711. Springer, 2021.
7. Scime, Anthony, ed. Web Mining: applications and techniques. IGI Global, 2005.

CA-403-C Object Oriented Analysis & Design

Credits	**4**
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Prerequisite
Basic knowledge of Object-Oriented Programming (OOP)
Familiarity with software development life cycle (SDLC).

Learning Objectives
1. Understand principles of object-oriented modeling.
2. Learn UML for structural and behavioral modeling.
3. Create class, object, use case, and activity diagrams.
4. Apply advanced modeling techniques.
5. Model architectural components (e.g., component and deployment diagrams).
6. Analyze and design object-oriented systems using UML.

Learning Outcomes
1. Explain object-oriented modeling and UML.
2. Develop structural models (class, object diagrams).
3. Model system behavior (use case, interaction, activity diagrams).
4. Use advanced behavioral models (state machines, events).
5. Create architectural models (component, deployment diagrams).
6. Design object-oriented software solutions using UML.

Course Content
Unit 1
Introduction to UML: Importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture.
Basic Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams.

Unit 2
Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces types and roles, Packages.
Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.

Unit 3
Basic Behavioral Modeling-I: Interactions, Interaction diagrams.
Basic Behavioral Modeling-II: Use cases, Use case Diagrams, Activity Diagrams.

Unit 4
Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams. Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams.

Reference Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson: The HUnified Modeling LanguageH User Guide, Pearson Education.
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech IndiaPvt. Ltd.
3. Meilir Page Jones, Fundamentals of Object Oriented Design in UML, Pearson Education.

CA-403-D Bioinformatics

Credits	4
Assignment / Seminar Presentation	10
Quiz	10
Internal Assessment / Mid Semester	20
Theory Examination (End Term)	60
Total	**100**
Time Duration of Theory Examination	3 Hours

Pre-requisites:
Basic knowledge of Biology, Programming & Mathematics

Learning Objectives:

1. To provide an integrative approach to the understanding of both theory and practice of bioinformatics
2. To apply biological concepts at different levels to study gene / protein analysis, and the
3. proteins implicated in diseases
4. To understand the evolution of the life

Learning Outcomes: A student who takes this course should be able to

1. Get knowledge and awareness of the basic principles and concepts of biology, computer science and mathematics
2. Use existing software effectively to extract information from large databases and to use this information in computer modeling
3. Develop problem-solving skills, including the ability to develop new algorithms and analysis methods
4. Get an understanding of the intersection of life and information sciences, the core of shared concepts, language and skills the ability to speak the language of structure-function relationships, information theory, gene expression, and database queries

Course Content:

Unit 1 Molecular Biology and Biological Chemistry: The Genetic Material, Gene structure and Information Content, Protein Structure and Function, The nature of Chemical bonds, Molecular Biology Tools, Genomic Information Content, Data Searches and Pairwise Alignments: Dot Plot, Simple Alignments, Gaps, Scoring Matrices, Needleman and Wunsch Algorithm, Global and local Alignments, Database searches, Multiple sequence Alignments,

Unit 2 Substitution Patterns: Patterns of substitutions within Genes, Estimating Substitution numbers, Variations in evolutionary rates between Genes, Molecular clocks, evolution in Organelles. Distance based methods of Phylogenetics: History of Molecular Phylogenies, Phylogenetic trees, Distance matrix methods, Maximum likelihood approaches, Multiple sequence Alignments. Character Based methods of Phylogenetics: Parsimony, Inferred ancestral sequences, Strategies for Faster searches, Consensus trees, tree confidence, Comparison of Phylogenetic methods, Molecular Phylogenies.

Unit 3 Genomics and Gene Recognition: Prokaryotic genomes, Prokaryotic gene structure, GCcontent Prokaryotic genomes, Prokaryotic gene density, Eukaryotic genomes, Eukaryotic gene structure, Open reading frames, GC-content Eukaryotic genomes, Gene expression, Transposition, Repetitive elements, Eukaryotic gene density, Protein and RNA structure prediction: Amino acids, Polypeptide composition, Secondary structure, Tertiary and quaternary structure,

Unit 4 Algorithms for Modeling Protein Folding, Structure prediction, Predicting RNA secondary structures, Proteomics: from Genomes to Proteomes, Protein classification, Experimental techniques, Inhibitors and drug design, Ligand screening, X-ray crystal structures, NMR structures, Empirical methods and prediction techniques, Post-translational modification prediction. .

Reference Books:

1. Dan E. Krane, Michael L. Raymer, "Fundamental Concepts of Bioinformatics", First Edition, 2003, Pearson Education, Inc. New Delhi.
2. Teresa Attwood, David Parry-Smith, "Introduction to Bioinformatics", 1999, Pearson Education, Inc. New Delhi.
3. Shuba Gopal, A. Haake, R. P. Jones, P. Tymann, "Bioinformatics: A Computing Perspective", First Edition, 2009, McGraw-Hill Education (India), New Delhi.
4. Yi-Ping P. Chen, "Bioinformatics Technologies", 2006, Springer India Pvt. Ltd., New Delhi.
5. Arthur Lesk, "Introduction to Bioinformatics", 2009, Oxford University Press, ISBN-13: 978-0199208043. 4. Bryan Bergeron, "Bioinformatics Computing", 2003, PHI Learning. New Delhi.