UNIVERSITY OF DELHI

CNC-II/093/1(40)/EC-1270/2024-25/160

Dated: 07.08.2024

NOTIFICATION

Sub: Amendment to Ordinance V

Following addition be made to Appendix-II-A to the Ordinance V (2-A) of the Ordinances of the University;

Add the following:

In pursuant of EC Resolution No. 5-14/ dated 27.07.2024, the following amendment in the syllabi of Department of Computer under Faculty of Mathematical Sciences, is notified for the information of all concerned:

(i) Revision of MCA Curriculum – ***As per Annexure-1***

(ii) Revision of MSc. Curriculum – ***As per Annexure-2***

(iii) Change in pre-requisites for Computer Science Undergraduate Programmes under UGCF-NEP – ***As per Annexure-3***

(iv) Renaming of Courses having similar content & proposal for Additional Electives. ***As per Annexure-4***

(v) Changes in position of DSC/DSE/GE papers for the students admitted in the year 2023-24. ***As per Annexure-5***

REGISTRAR

Annexure-1

Summary of Minor Modifications in the MCA program syllabus

Some minor changes are proposed in the MCA program syllabus. These are listed below:

1. There are some intra-semester movements of some courses and minor updates in the names and contents of certain courses, listed below:

S.No.	Existing	Proposed	Change
1.	MCAC 102: Computer System Architecture (Semester I)	MCAC 102: Computer Organization and Architecture (Semester I)	Enhanced focus on organization aspect
2.	MCAC 203: Operating Systems (Semester II)	MCAC 105: Operating Systems (Semester I)	Moved from Semester II to Semester I
3.	MCAC 105: Data Mining (Semester I)	MCAC 203: Data Mining (Semester II)	Moved from Semester I to Semester II
4.	MCAC 204: Machine Learning (Semester II)	MCAC 204: Artificial Intelligence and Machine Learning (Semester II)	The change reflects an integrated approach to Artificial Intelligence and Machine Learning
5.	MCAE 309: Automata Theory (Semester III)	MCAE 203: Automata Theory (Semester II)	Moved from Semester III to Semester II

2. Core courses on (i) Statistical Methods and (ii) Artificial Neural Networks and Deep Learning have been introduced. Further, two credit courses on (i) Software Tools and Techniques and (ii) Reading Skills have been introduced.

3. Based on student feedback, elective courses on information retrieval, social networks, cloud computing, and data analysis and visualization have been introduced.

   Each theory course is a four-credit course and each practical course is a two-credit course. To pass a course, a student should obtain ‘D’ grade or a higher grade.

UNIVERSITY OF DELHI
MASTER OF COMPUTER APPLICATIONS (MCA)

I. MCA Programme Details:

Programme Objectives (POs):

Master of Computer Applications (MCA) is a full-time, four-semester course that includes one semester of project work in the fourth semester. The objective of the MCA programme is to impart quality education in computer science and its applications so that students are well prepared to face the challenges of the highly competitive IT industry. The course structure ensures the overall development of the student while concentrating on imparting the technical skills required for an IT profession. No wonder, today, after forty years of its existence, its alumni are holding important positions in the IT industry and academia in India and abroad.

Programme Specific Outcomes (PSOs):

The programme is designed to:

PSO1: enable the students to apply the computing and soft skills acquired in the MCA program to design and develop innovative applications for the betterment of society.
PSO2: provide exposure to techniques that would enable the students to design, implement, and evaluate IT solutions.
PSO3: To enable the students to meet the challenges of computer science and applications.

Programme Structure:

The MCA programme is a two-year course divided into four semesters. A student is required to complete 80 credits for the completion of the course and the award of a degree.

		Semester	***Semester***
Part – I	First Year	Semester I	Semester II
Part – II	Second Year	Semester III	Semester IV

<u>Course Credit Summary</u>

Semester	**Core Courses**			**Elective Course**			**Open Elective Course**			**Total Credits**
	No.of Courses	Credits (Th+T+P)	**Total Credits**	No. of Courses	Credits (Th+T+P)	**Total Credits**	No.of Courses	Credits (Th+T+P)	**Total Credits**	
I	6	16+0+6	**22**	0	0	**0**	0	0	**0**	**22**
II	5	12+0+6	**18**	1	3+0+1	**4**	0	0	**0**	**22**
III	3	9+0+3	**12**	1	3+0+1	**4**	1	3+0+1	**4**	**20**
IV	Project	0	**20**	0	0	**0**	0	0	**0**	**20**
Total Credits for the Courses			**72**			**8**			**4**	**84**

- English shall be the medium of instruction and examination.
- Examinations shall be conducted at the end of each semester as per the academic calendar notified by the University.
- The students will choose the elective courses out of the list of courses that are offered in a semester. An elective course offered by another department/ center/ institute may be taken subject to approval of the department.
- **Project:** Each student shall carry out a project in the fourth semester. The projects will be carried out under the supervision of the teacher(s) of the department. When the project is carried out in an external organization (academic institution/ industry), a supervisor may also be appointed from the external organization. The project work will be evaluated jointly by the internal supervisor and an examiner to be appointed by the department in consultation with the internal supervisor. The project evaluation shall be as follows:

  (a) Mid-semester evaluation: 30% weight
  (b) End-semester evaluation
    (i) Dissertation: 30% weight
    (ii) Viva-voce: 40% weight
- Examinations for courses specified in the odd (even) semesters shall be conducted only in the respective odd (even) semesters.
- **Promotion Criteria:** To be eligible for promotion to second year, a student must successfully complete at least 30 credits out of the courses prescribed for semester I and semester II, taken together.
- **Evaluation Criteria:** The scheme of evaluation shall be as follows: the performance of the students will be evaluated based on a comprehensive system of continuous and end-semester evaluation. For each course, there shall be one minor test, assignments/ laboratory work, quizzes, and an end-semester examination: (Mid-Term exam, assignments/practical & laboratory work - 30% weight; end-semester examination - 70% weight), except for practical courses where internal assessment and end-semester examination shall carry 50% weight each. The evaluation of the practical courses will be based on internal assessment and the end-semester evaluation by a board of examiners appointed by the Committee of Courses.
- In order to pass a course and earn credits prescribed for it, a student must obtain 'D' grade or a higher grade.
- **Conversion of Marks into Grades:**

Letter Grade	Numerical Grade	Formula	Computation of grade cut off
O (outstanding)	10	$m \geq \overline{X} + 2.5\sigma$	the value of $\overline{X} + 2.5\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 2.5\sigma$ or 90% whichever is lower
A+ (Excellent)	9	$\overline{X} + 2.0\sigma \leq m < \overline{X} + 2.5\sigma$	the value of $\overline{X} + 2.0\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 2.0\sigma$ or 80% whichever is lower
A (Very Good)	8	$\overline{X} + 1.5\sigma \leq m < \overline{X} + 2.0\sigma$	the value of $\overline{X} + 1.5\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 1.5\sigma$ or 70% whichever is lower
B+ (Good)	7	$\overline{X} + 1.0\sigma \leq m < \overline{X} + 1.5\sigma$	the value of $\overline{X} + 1.0\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 1.0\sigma$ or 60% whichever is lower
B (Above Average)	6	$\overline{X} \leq m < \overline{X} + 1.0\sigma$	the value of $\overline{X}$ to be taken into account for grade computation will be actual $\overline{X}$ or 50% whichever is lower

C (Average)	5	$\overline{X} - 0.5\sigma \leq m < \overline{X}$	the value of $\overline{X} - 0.5\sigma$ to be taken into account for grade computation will be actual $\overline{X} - 0.5\sigma$ or 45% whichever is lower
D (Pass)	4	$\overline{X} - 1.0\sigma \leq m < \overline{X} - 0.5\sigma$	the value of $\overline{X} - 1.0\sigma$ to be taken into account for grade computation will be actual $\overline{X} - 1.0\sigma$ or 40% whichever is lower
F (Fail)	0	$\overline{X} - 1.0\sigma > m$	

- **CGPA to Percentage Conversion:**
  The formula for calculating the final percentage of marks from Cumulative Grade Point Average (CGPA) will be as per the University rules.
- **Eligibility for Award of Degree and Division Criteria:**
  A student would be eligible for the award of an MCA degree, provided he/ she earns the required number of credits. Such a student shall be categorized (on the basis of the CGPA to percentage conversion as per university rules) on the basis of the CGPA acquired during Part-I and Part-II examinations taken together, as follows:

  a) I Division: 60% or more marks in the aggregate
  b) II Division: 50% or more marks but less than 60% marks in the aggregate.
  c) Pass: 40% or more marks but less than 50% marks in the aggregate.

- **Attendance Requirement**

  No candidate shall be considered to have pursued a regular course of study unless he/she has attended 66.67% of the total number of classroom/ tutorial/ lab sessions conducted in each semester during his/her course of study. A student not complying with this requirement shall not be allowed to appear in the semester examinations. However, considering the merit of the case, the Head of the Department may condone the required percentage of attendance by not more than 10 percent during a semester.

- **Span Period**
  The span period will be four years from the date of registration in the programme.

- **Eligibility, Mode of Admissions, and Number of Seats in the MCA programme:**
  To be decided by the University in every academic year.

Semester wise Details of MCA Course

Semester I					
	Number of core courses	5			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
MCAC101	Object Oriented Programming	3	0	1	**4**
MCAC102	Computer Organization and Architecture	3	0	1	**4**
MCAC103	Data Structures	3	0	1	**4**

MCAC104	Database Systems	3	0	1	**4**
MCAC105	Operating Systems	3	0	1	**4**
MCAC106	Software Tools and Techniques	1	0	1	**2**
	Total credits in core course	22			
	Number of elective courses	0			
	Total credits in elective course	0			
	Number of open electives	0			
	Total credits in elective course	0			
	Total credits in Semester I	**22**			

Semester II					
	Number of core courses	5			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
MCAC201	Design and Analysis of Algorithms	3	0	1	**4**
MCAC202	Data Communication and Computer Networks	3	0	1	**4**
MCAC203	Data Mining	3	0	1	**4**
MCAC204	Artificial Intelligence and Machine Learning	3	0	1	**4**
MCAC205	Reading Skills	0	0	2	**2**
	Total credits in core course	18			
	Number of elective courses	1			
	Credits in each elective	Theory	Tutorial	Practical	**Total**
	Elective course 1	3	0	1	**4**
	Total credits in elective courses	4			
	Number of open electives	0			
	Credits in each open elective	0			
	Total credits in open elective	0			
	Total credits in Semester II	22			

List of Elective Courses for Semester II		
Course Code	**Course Title**	**Th-T-P**
MCAE 201	Web Technologies	3-0-1
MCAE 202	Java Programming	3-0-1
MCAE 203	Automata Theory	3-0-1
MCAE 204	Statistical Methods	3-0-1

Semester III					
	Number of core courses	3			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
MCAC301	Information Security	3	0	1	**4**
MCAC302	Software Engineering	3	0	1	**4**
MCAC303	Artificial Neural Networks and Deep Learning	3	0	1	**4**
	Total credits in core course	**12**			
	Number of elective courses	1			
	Credits in each elective	Theory	Tutorial	Practical	**Total**
	Elective course I	3	0	1	**4**
	Total credits in elective courses	**4**			
	Number of open elective	1			
	Credits in each open elective	Theory	Tutorial	Practical	**Total**
	Open Elective I	3	0	1	**4**
	Total credits in open elective courses	**0**			
	Total credits in Semester III	**20**			

List of Elective Courses for Semester III		
Course Code	**Course Title**	**Th-T-P**
MCAE301	Compiler Design	3-0-1
MCAE302	Network Science	3-0-1
MCAE303	Natural Language Processing	3-0-1
MCAE304	Information Retrieval	3-0-1
MCAE305	Combinatorial Optimization	3-0-1
MCAE306	Digital Watermarking and Steganography	3-0-1
MCAE307	Quantum Computing	3-0-1
MCAE308	Computer Graphics	3-0-1
MCAE309	Digital Image Processing	3-0-1
MCAE310	Social Networks	3-0-1
MCAE311	Cloud Computing	3-0-1

List of Open Elective Courses for Semester III		
Course Code	**Course Title**	**Th-T-P**
MCAO301	Data Mining for Business Applications (for other departments)	3-0-1
MCAO302	Data Science using Python	3-0-1
MCAO303	Data Analysis and Visualization	3-0-1

Semester IV		
	Number of core courses	1
Course Code	**Course Title**	
MCAC401	Project work	20

	Number of elective courses	0
	Total credits in elective courses	0

	Number of open electives	0
	Total credits in open elective	0
	Total credits in Semester VI	**20**

Total Credits = 22 +22 + 20 + 20 = 84

Selection of Elective Courses:

The students may select the elective courses out of the list of courses that are offered in a semester.

II. Content Details for MCA Programme

SEMESTER – I

MCAC101: OBJECT ORIENTED PROGRAMMING [3-0-1]

Course Objectives: The course aims to develop the student's problem-solving skills. The course also focuses on debugging skills. The student learns to develop modular well documented code.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: select a suitable programming construct and in-built data structure for a given problem.

CO2: design, develop, document, and debug modular programs.

CO3: use recursion as a programming paradigm for problem-solving.

CO4: apply object-oriented paradigm for problem-solving.

Syllabus:

Unit-I Introduction: Notion of class, object, identifier, keyword, and literal; basic data types: int, float, string, Boolean; basic operators (arithmetic, relational, logical, assignment), standard libraries.

Unit-II Program Development: Modular program development, input and output statements, control statements: branching, looping, exit function, break, continue, and switch-break; arrays and pointers, testing and debugging a program.

Unit-III Recursion: Use of recursion as a programming paradigm for problem-solving.

Unit-IV Object Oriented Programming Concepts: Use of classes, inheritance, and Polymorphism

Unit-V Exception Handling and File Handling: Reading and writing text and structured files, errors and exceptions.

Readings:

1. R. G. Dromey, **How to Solve it by Computer,** Pearson, 2006.
2. Stanley B. Lippman, and Josée Lajoie, C++ PRIMER, Addison-Wesley, 2019.
3. Bjarne Stroustrup, **The C++ Programming Language (4th Edition)** Addison-Wesley, 2013

MCAC102: COMPUTER ORGANIZATION AND ARCHITECTURE [3-0-1]

Course Objectives: The course is designed to introduce basic building blocks of digital electronics, design and architecture of computer systems. This course aims to develop the skill to architect a digital computer system using a simulator.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe the basic organization of computer hardware.

CO2: represent and manipulate data – number systems, conversion between different number systems, perform binary arithmetic.

CO3: design simple combinational and sequential logic circuits - flip-flops, counters, shift registers, adders, subtractor, multiplexer, demultiplexer, and Arithmetic/Logic unit.

CO4: design a CPU simple computer / microprocessor: instruction format, instruction set, addressing modes, bus structure, input/output architecture, memory unit, Arithmetic/Logic and control unit, data, instruction and address flow.

Syllabus:

Unit-I Basic Building Blocks: Boolean logic and Boolean algebra, tri-state logic; flip-flops, counters, shift registers, adders, subtractors, encoders, decoders, multiplexers, demultiplexers.

Unit-II Processor Design: CPU organization, register organization, stack organization, microprogrammed control unit, RISC architecture; microprocessor architecture, modern computing architectures. Bus and memory transfers, arithmetic, logic shift micro-operations; basic computer organization: common bus system, instruction formats, instruction cycle, interrupt cycle, input/output configuration.

Unit-III Memory Unit: Primary memory, secondary memory, associative memory, sequential access, direct access storage devices.

Unit-IV Input-Output Architecture: Input/Output devices; data transfer schemes - programmed I/O and DMA transfer; data transfer schemes for microprocessors.

Readings:

1. M. Morris Mano, **Computer System Architecture,** Revised 3rd Edition, Pearson, 2018.
2. W. Stallings, **Computer Organization and Architecture: Designing for Performance**, 9th Edition, Pearson Education, 2012.
3. A.S. Tanenbaum**, Structured Computer Organization**, 6th Edition, Prentice-Hall of India, 2012.
4. J.P. Hayes, **Computer System Architecture and Organization**, 3rd Edition, McGraw-Hill Education, 2017.

MCAC103: DATA STRUCTURES [3-0-1]

Course Objectives: The course is designed to develop the student's ability to develop algorithms for creating and manipulating basic data structures. The students also learn to compare and contrast different data structures for a given problem.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: develop programs using basic data structures: sets, lists, stacks, queues, trees, graphs and advanced data structures like balanced trees.

CO2: identify the suitable data structure for the problem at hand.

CO3: evaluate computational complexity of an algorithm.

Syllabus:

Unit-I Computational Complexity: Growth of functions, Recurrence relations: Asymptotic notations, solving recurrences using recursion trees.

Unit-II Basic Data Structures: Primitive data types, abstract data types, linear vs non linear data structures, arrays - static and dynamic, linked lists - single, doubly-linked, circular; stacks and queues using arrays and linked lists; operations, their analysis and applications.

Unit-III Trees: Binary tree, tree traversals, binary search tree, height balanced trees: AVL, red-black trees, B and B+ Trees, heaps, priority queues, operations, their analysis and applications.

Unit-IV Sets: Sets, multisets, maps, hash tables, dictionaries.

Unit-V Graphs: Representation of graphs, searching in graphs: breadth first search and its applications, depth first search and its applications, shortest paths, spanning trees.

Readings:

1. M. Goodrich, R. Tamassia, D. Mount, **Data Structures and Algorithms in C++/Java**, 2nd Edition, 2016, Wiley.
2. Elliot B. Koffman, Paul A.T. Wolfgang, **Objects, Abstraction, Data Structures and Design Using C++/Java**, 1st Edition, 2005, Wiley Global Education.
3. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, **Introduction to Algorithms**, 3rd Edition, 2010, Prentice-Hall of India Learning Pvt. Ltd.

MCAC104: DATABASE SYSTEMS [3-0-1]

Course Objectives: This course aims to achieve competency in designing and developing databases for different applications, keeping in mind their constraints. The students learn to apply database design techniques to design a database and answer the user's queries.

Course Learning Outcomes:

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

CO1: apply logical database design principles, including data normalization and E-R/EE-R diagrams to design an application.

CO3: choose suitable integrity constraints for an application.

CO4: answer database queries using Structured Query Language (SQL).

CO5: enumerate the concurrency control issues and discuss their resolution strategies.

CO6: design and implement database projects.

Syllabus:

Unit-I Introduction: Data modelling for a database, abstraction and data integration, three-level DBMS architecture.

Unit-II Database Design: Entity-Relationship model, Extended Entity-Relationship model.

Unit-III Relational Model and Relational Data Manipulations: Relation, conversion of ER diagrams to relations, integrity constraints, relational algebra, relational domain and tuple calculus.

Unit-IV Structured Query Language: DDL, DML, Views, Embedded SQL.

Unit-V Relational Database Design Concepts: Functional dependencies, determining keys, normalization-, lossless join and dependency preserving decomposition.

Unit-VI Transaction Management: ACID properties, concurrency control in databases, transaction recovery.

Unit-VII Introduction to NoSQL and XML Databases.

Readings:

1. A. Silberschatz, H. Korth and S. Sudarshan**, Database System Concepts,** 6th Edition**,** McGraw Hill, 2014.
2. R. Ramakrishnan and J. Gehrke, **Database Management Systems,** 3rd Edition, McGraw Hill, 2014.
3. Philip Lewis, Arthur Berstein and Michael Kifer, **Databases and Transaction Processing - An application-oriented Approach**, Prentice Hall, 2003

MCAC105: OPERATING SYSTEMS [3-0-1]

Course Objectives: This course aims to provide a functional perspective of the operating systems and develop skills to experiment with different components, tasks and services, including job scheduling, memory management, device management, process management and file management.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe the basic functions of an Operating System.

CO2: distinguish between different types of operating systems to use each of them most efficiently in the respective application areas.

CO3: describe different techniques for managing computer resources like CPU, memory, file and devices.

CO4: implement algorithms for managing computer resources.

CO5: experiment with different components, tasks and services of the operating system.

Syllabus:

Unit 1- Introduction: operating system design goals, evolutionary history of operating systems; concept of user, job and resources; batch processing, multi-programming, time sharing; structure and functions of operating system.

Unit 2- Process Management: Process states, state transitions, process control structure, context switching, process scheduling, threads.

Unit 3- Memory Management: Address binding, dynamic loading and linking concepts, logical and physical addresses, contiguous allocation, fragmentation, paging, segmentation, combined systems, virtual memory, demand paging, page fault, page replacement algorithms, global vs local allocation, thrashing, working set model, pre-paging.

Unit 4- Process Synchronization: Process interaction, shared data and critical section, mutual exclusion, busy form of waiting, lock and unlock primitives, synchronization, classical problems of synchronization, semaphores, monitors, conditional critical regions, system deadlock, wait for graph, deadlock handling techniques: prevention, avoidance, detection and recovery.

Unit 5- File and Secondary Storage Management: File attributes, file types, file access methods, directory structure, file system organization and mounting, allocation methods, free space management; disk structure, logical and physical view, disk head scheduling, formatting, swap management.

Readings:

1. G. Gagne, A. Silberschatz, P. B. Galvin, and **Operating System Concepts** , 10th Edition,Wiley, 2018.
2. Gary Nutt, Nabendu Chaki, Sarmistha Neogy, **Operating Systems: A Modern Approach,** 3rd Edition, Addison Wesley, 2009.

3. D.M. Dhamdhere, **Operating Systems: A Concept Based Approach,** 2nd Edition, Tata McGraw-Hill, 2007.

MCAC106: SOFTWARE TOOLS AND TECHNIQUES [1-0-1]

Course Objective:

To develop proficiency in the use of software tools required for project development.

Course Learning Outcomes:

On completing this course, a student will be able to:

CO1: use the command line interface efficiently
CO2: use features of version control systems
CO3: debug and profile code
CO4: manage dependencies

Syllabus:

Shell tools and scripting, editors (Vim), data wrangling, command-line environment, version control (Git), debugging and profiling, metaprogramming: working with daemons, FUSE, backups, APIs, common command-line flags/patterns, window managers, VPNs, Markdown, Booting + Live USBs, Docker, Vagrant, VMs, cloud, OpenStack, notebook programming

Readings:

1. C. Newham, Learning the Bash Shell: **Unix shell programming**. O'Reilly Media, Inc.; 2005.
2. W. Shotts, **The Linux command line: a complete introduction**. No Starch Press; 2019.
3. https://git-scm.com/book/en/v2

SEMESTER – II

MCAC201: DESIGN AND ANALYSIS OF ALGORITHMS [3-0-1]

Course Objectives: The course introduces techniques for algorithm design and analysis of the asymptotic performance of these algorithms. This course aims to achieve competence in designing efficient algorithms using different data structures for real-world problems.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe various algorithm design techniques, including iteration, divide and conquer, dynamic programming, and greedy approach algorithms.

CO2: analyze the strengths and weaknesses of each technique.

CO3: identify and apply technique(s) suitable for simple applications.

CO4: demonstrate the correctness of algorithms and analyze their time complexity.

Syllabus:

Unit-I Computational Complexity: Review of growth of functions, asymptotic notation, Master's theorem.

Unit-II Iterative Algorithms: Searching and sorting techniques - linear search, binary search, insertion sort – time complexity and proof of correctness.

Unit-III Divide and Conquer: Binary search, merge sort and quick sort – time complexity

Unit-IV Lower bounding techniques: Decision Trees.

Unit-V Linear Sorting: Count sort, radix sort, bucket sort.

Unit-VI Greedy Algorithms: Interval scheduling, minimum spanning trees – Prim's algorithm, Kruskal algorithm, shortest path problem – Djikstra's algorithm.

Unit-VII Dynamic Programming: Weighted interval scheduling, matrix chain multiplication, knapsack problem, longest common subsequence.

Unit-VIII String Processing: Brute-force method, KMP algorithm.

Unit-IX Randomized algorithms: Introduction to random numbers, randomized Qsort, randomly built BST.

Readings:

1. J. Kleinberg and E.Tardos, **Algorithm Design**, 1st Edition, Pearson Education India, 2013.
2. T. H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, **Introduction to Algorithms**, 3rd Edition, Prentice-Hall of India, 2010.
3. Sara Baase, Allen Van Gelder, **Computer Algorithm – Introduction to Design and Analysis**, 3rd edition, Pearson Education, 2002.
4. Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, **Algorithms**, 1st Edition, Tata McGraw Hill, 2017.
5. Richard Johnsonbaugh and Marcus Schaefer, **Algorithms**, Pearson Education India, 2014.

MCAC202: DATA COMMUNICATION AND COMPUTER NETWORKS [3-0-1]

Course Objectives: The course introduces the basic concepts of data communication techniques, and computer network, various protocols and their applications. It also aims to develop skills for interworking between computer networks and switching components in telecommunication systems.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: apply data communication techniques in real-life experiments like telemetry and develop skills to modify the existing ones to better suit in different situations.

CO2: develop skills to apply services of various computer networks in various technical and professional fields.

CO3: reduce the overheads of different reference models and optimize their performances.

CO4: develop skills to apply, modify and develop new protocols in different layers of existing protocol stacks to suit customized requirements.

CO5: use various network applications to avail network services efficiently and develop skills to design new applications to open new services.

Syllabus:

Unit-I Data Communication Techniques: Theoretical basis of data communication, analog and digital signals, time domain and frequency domain analysis, frequency spectrum and bandwidth, asynchronous and synchronous transmission, data encoding and modulation techniques, baseband and broadband transmission, pulse code modulation, baud rate and bitrate of a channel, multiplexing- FDM and TDM, transmission medium, transmission errors – error detection techniques.

Unit-II Network Classification and Network Services: Local Area Networks, Metropolitan

Area Networks, Wide Area Network, wireless networks, internetworking and Internet, business and home applications, mobile user services.

Unit-III Network Architecture and Reference Models: Layered network architectures, protocol hierarchies, interface and services, ISO-OSI reference model, TCP/IP reference model, Internet protocol stack.

Unit-IV Data Link Layer Functions and Protocols: Framing, flow-control, error recovery protocols, Data link layer of internet-PPP protocol.

Unit-V Medium Access Sublayer: CSMA/CD protocol and Ethernet, hubs and switches, fast Ethernet, gigabit Ethernet, CSMA/CA protocol and WLAN.

Unit-VI Network and transport layers functions and protocols: Network switching mechanisms- circuit switching, packet switching, routing and congestion control, TCP/IP protocol architecture.

Unit-VII Network Applications: File transfer protocol, electronic mail, World Wide Web.

Readings:

1. A S Tanenbaum, **Computer Networks**, 5th Edition, Pearson Education India, 2013
2. Behrouz A Forouzan, **Data Communications and Networking**, 5th Edition, McGraw Hill Education, 2017.

MCAC203: DATA MINING [3-0-1]

Course Objectives: This course introduces the KDD process. It would enable students to translate real-world problems into predictive and descriptive tasks. The course also covers data cleaning and visualization, as well as supervised and unsupervised mining techniques.

Course Learning Outcomes:

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

CO1: experiment with the basic data exploration methods to develop an understanding of given data.

CO2: identify a suitable pre-processing method for a given problem.

CO3: implement and apply appropriate data mining algorithms for a given problem.

CO4: use GUI-based mining softwares.

Syllabus:

Unit-I Overview: The process of knowledge discovery in databases, predictive and descriptive data mining techniques, supervised and unsupervised learning techniques.

Unit-II Data preprocessing: Data cleaning, data transformation, data reduction, discretization.

Unit-III Classification: Curse of dimensionality, overfitting and underfitting, decision trees, decision rules, Naïve Bayes classifier, instance-based methods, model evaluation and validation methods.

Unit-IV Clustering: Basic issues in clustering, k-means clustering, hierarchical clustering, density-based methods, cluster validation methods and metrics.

Unit-V Association Rule Mining: Frequent item sets, closed and maximal item sets, Apriori algorithm for association rule mining.

Readings:

1. P. Tan, M. Steinbach and V. Kumar, **Introduction to Data Mining**, Addison Wesley, 2016.
2. Jiawei Han and Micheline Kamber, **Data Mining: Concepts and Techniques,** 3rd Edition, Morgan Kaufmann, 2011.
3. Charu C Agrawal, **Data Mining: The Textbook**, Springer, 2015.

MCAC204: ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING [3-0-1]

Course Objectives: Beginning with a comprehensive overview of the AI techniques, the course introduces the supervised and unsupervised machine learning (ML) techniques, alongwith their applications in solving real-world problems. The course also covers evaluation and validation methods for ML models.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: discuss Turing Test, and various methods of knowledge representation as applicable to a given context.

CO2: design and implement supervised and unsupervised machine learning algorithms for real-world applications while understanding the strengths and weaknesses.

CO3: analyse the computational complexity of various machine learning algorithms.

CO4: fine-tune machine learning algorithms and evaluate models generated from data.

Syllabus:

Unit-I Introduction to Artificial Intelligence: Evolution of artificial intelligence (AI) as a discipline, definitions and approaches, philosophical issues, AI for all, ethical issues and responsible AI.

Unit-II Introduction to Machine Learning: Hypothesis and target class, bias-variance tradeoff, Occam's razor, approximation and estimation errors, curse of dimensionality, dimensionality reduction, feature scaling, feature selection methods.

Unit-III Regression: Linear regression with one variable, linear regression with multiple variables, gradient descent, logistic regression, polynomial regression, over-fitting, regularization. performance evaluation metrics, validation methods.

Unit-IV Classification: Decision trees, Naive Bayes classifier, perceptron, multilayer perceptron, neural network, back-propagation algorithm, support vector machine, kernel functions.

Unit V Evaluation: Performance evaluation metrics, ROC Curves, validation methods, bias-variance decomposition, model complexity.

Unit-VI Unsupervised Learning: Clustering, distance metrics, mixture models, expectation maximization, cluster validation methods.

Readings:

1. E. Alpaydin, **Introduction to Machine Learning,** MIT press, 2014.
2. T. M. Mitchell, **Machine Learning,** McGraw Hill Education, 2017.
3. Christopher M. Bishop, **Pattern Recognition And Machine Learning**, Springer-Verlag, 2016.
4. Shai Shalev-Shwartz, Shai Ben-David, **Understanding Machine Learning: From Theory to Algorithms,** Cambridge Press, 2014.
5. Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, eds. **Machine learning: An artificial intelligence approach**, Springer Science & Business Media, 2013.

MCAC205: READING SKILLS [0-0-2]

Course Objectives: The course aims to develop an important skills of independent reading.

Course Learning Outcomes:

On completing this course, a student will be able to:

CO1: Develop a habit of independent reading.

CO2: Given a requirement, independently select sources of reading.

CO3: Read and assimilate independently.

This is a self-study course. The students will carry out extensive reading on a topic to be assigned by the department.

MCAE201: WEB TECHNOLOGIES [3-0-1]

Course Objectives: The course aims to develop skills for static and dynamic web development, including the use of front-end designing tools and client-side and server-side programming. It also sensitizes and prepares the student to apply security and privacy concerns in an application.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe Internet, World Wide Web, client-server architecture and communication protocols.

CO2: design multi-platform web applications.

CO3: apply various client-side web technologies for front-end development.

CO4: work with server-side web technologies.

CO5: apply web technologies for retrieval of information.

Syllabus:

Unit-I Introduction: Introduction to networking, TCP/IP, DNS, Internet and its evolution, World Wide Web, Web 2.0, Web 3.0, network communication protocols (HTTP/HTTPS, SMTP, IMAP, POP, FTP), client-server architecture, web applications architecture, application and web servers, web clients.

Unit-II Front-end Development: Introduction to HTML5, HTML elements, HTML tags, lists, tables, frames, forms, basics of XHTML, CSS style sheets, DOM, XML, XSLT

Unit-III Client-Side Programming: JavaScript syntax, variables and data types, literals, functions, objects, arrays, built-in objects, event handling, modifying element style, document trees.

Unit-IV Server-Side Programming: Creation of dynamic content, server-side programming using Java Servlets, web services, session management, introduction to server-side scripting, accessing database from the front-end.

Unit-V Web Security, Cookies and Authentication: Security threats, security risks of a website, web attacks and their prevention, web security model, setting, accessing and destroying cookies, anonymous access, authentication by IP address and domain, Integrated Windows Authentication, digital signatures, digital certificates, firewalls.

Readings:

1. Jeffery C. Jackson, **Web Technologies: A Computer Science Perspective**, Pearson Education India, 2007.
2. Achyut Godbole and Atul Kahate, **Web Technologies: TCP/IP, Web/Java Programming, and Cloud Computing,** 3rd Edition, McGraw-Hill Education, 2013.
3. Roger S Pressman and David Lowe, **Web Engineering: A Practitioner's Approach,** Tata McGraw-Hill, 2017.
4. Mark Pilgrim, **HTML5: Up and Running**, O'Reilly | Google Press, 2010 .
5. Jim Keogh, **J2EE: The Complete Reference**, McGraw Hill Education, 2017.

MCAE202: JAVA PROGRAMMING [3-0-1]

Course Objectives: This course introduces object-oriented concepts through Java programming. The course also include reading and writing files using exception handling.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: apply the object-oriented concepts: Classes, Objects, Inheritance, Polymorphism, for problem solving.

CO2: handle program exceptions.

CO3: design, implement, document, test, and debug a Java application consisting of multiple classes.

CO4: handle input/output through files.

CO5: create Java applications with graphical user interface (GUI).

Syllabus:

Unit-I Introductory Concepts: program, identifiers, variables, constants, primitive data types, expressions, control statements, structured data types, arrays, functions.

Unit-II Object Oriented Concepts: Abstraction, encapsulation, objects, classes, methods, constructors, inheritance, polymorphism, static and dynamic binding, overloading, abstract classes, interfaces and packages.

Unit-III File Handling: Byte stream, character stream, file I/O basics, file operations, serialization.

Unit-IV Exception Handling: Throw and catch exceptions, throw, try, and catch blocks, multiple catch blocks, finally clause, throwable class, types of exceptions, java.lang exceptions, built-in exceptions.

Unit-V GUI Design: GUI based I/O, input and message dialog boxes, swing components, displaying text and images in window.

Readings:

1. James Gosling, Bill Joy, Guy L. Steele Jr, Gilad Bracha, Alex Buckley, **The Java Language Specification, Java SE,** 7th Edition, Addison-Wesley, 2013.
2. Cay S. Horstmann, **Core Java - Vol. I – Fundamentals,** 10th Edition, Pearson, 2017.
3. Deitel and Deitel, **Java-How to Program,** 9th Edition, Pearson Education, 2012.
4. Richard Johnson, A**n Introduction to Java Programming and Object-Oriented Application Development,** Thomson Learning, 2006.
5. Herbert Schildt, **Java: The Complete Reference,** 10th Edition, McGraw-Hill Education, 2018.

MCAE203: AUTOMATA THEORY [3-0-1]

Course Objectives: The course introduces the theoretical models of computation and their limitations. The students will develop skills to analyze automata and their computational power to recognize languages. The students will learn to apply the knowledge of automata theory, grammar, and Turing machines for solving problems in language translation.

Course Learning Outcomes:

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

CO1: describe the mathematical model of machines.

CO2: examine whether a language is regular or context free.

CO3: identify ambiguity in grammar

CO4: construct a parse tree for the given grammar.

CO5: design finite automaton, push down automator, Turing machine for a language.

CO6: examine Turing decidability of a language.

Syllabus:

Unit-I Introduction: Alphabets, strings, and languages.

Unit-II Finite Automata and Regular Languages: Deterministic and non-deterministic finite automata, regular expressions, regular languages and their relationship with finite automata, pumping lemma and closure properties of regular languages.

Unit-III Context Free Grammars and Pushdown Automata: Context free grammars (CFG), parse trees, ambiguities in grammars and languages, pushdown automaton (PDA) and the language accepted by PDA, deterministic PDA, Non- deterministic PDA, properties of context free languages; normal forms, pumping lemma, closure properties, decision properties.

Unit-IV Turing Machines: Turing machine as a model of computation, programming with a Turing machine, variants of Turing machine and their equivalence.

Unit-V Undecidability: Recursively enumerable and recursive languages, undecidable problems about Turing machines: halting problem, Post Correspondence Problem, and undecidability problems about CFGs.

Readings:

1. J. E. Hopcroft, R. Motwani, and J. D. Ullman, **Introduction to Automata Theory, languages, and computation**, 2016.
2. H.R. Lewis, C.H. Papadimitriou, C. Papadimitriou, **Elements of the Theory of Computation**, 2nd Edition, Pearson Education, 2015
3. P. Linz, **Introduction to Automata Theory, Languages, and Computation**, Jones and Bartlett, 2016.

MCAE 204 Statistical Methods

Course Objectives: To equip students with the skills necessary to apply statistical methods for various applications.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: apply descriptive statistical techniques to summarize and interpret data
CO2: apply inferential statistical methods, including hypothesis testing and confidence interval estimation.
CO3: perform and interpret simple and multiple linear regression analysis
CO4: apply principles of experimental design in the context of a problem

Syllabus:

Unit-1 Introduction: Descriptive statistics: measures of central tendency and variability, representation of data: stem and leaf diagram, histogram, boxplot, and ogive; bar diagram and its variations, pie charts; probability distributions: discrete and continuous, joint and conditional probability; theory of attributes: coefficient of association and coefficient of colligation.

Unit-II: Statistical Inference: Parameter and statistic; sampling distributions, confidence intervals and margin of error, hypothesis testing; non-parametric inference: non-parametric tests: Mann-Whitney U test, Kruskal-Wallis test, Spearman's rank correlation coefficient.

Unit-III Regression and Classification: Correlation: measure and significance, simple linear regression, multiple linear regression, one-way classification, analysis of variance, two-way classification, analysis of covariance, curvilinear regression, factorial experiments.

Readings:

1. Robert S. Witte and John S. Witte, **Statistics**, John Wiley & Sons Inc; 11th edition, 2021
2. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, **An Introduction to Statistical Learning**, Springer, 2023.
3. G. W. Snedecor, W. G. Cochran, **Statistical Methods**, Iowa State University Press, 1973
4. John A. Rice, **Mathematical Statistics and Data Analysis**, Cengage, 2013

SEMESTER–III

MCAC301: INFORMATION SECURITY [3-0-1]

Course Objectives: The course aims to train the students to maintain the confidentiality, integrity and availability of data. The student learns various data encryption protocols for transmitting data over unsecured channels in a network.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe various security issues.

CO2: implement symmetric and asymmetric cryptographic methods.

CO3: describe the role and implementation of digital signatures.

CO4: make use of software tools for intrusion detection, auditing and logging.

Syllabus:

Unit-I Overview of Security: Protection versus security; aspects of security– confidentiality, data integrity, availability, privacy; user authentication, access controls, Orange Book Standard.

Unit-II Security Threats: Program threats, worms, viruses, Trojan horse, trap door, stack and buffer overflow; system threats- intruders; communication threats: tapping and piracy.

Unit-III Cryptography: Substitution, transposition ciphers, symmetric-key algorithms: Data Encryption Standard, Advanced Encryption Standard, IDEA, block cipher operation, stream ciphers: RC-4. Public key encryption: RSA, ElGamal. Diffie-Hellman key exchange. Elliptic Curve, EC cryptography, Message Authentication code (MAC), cryptographic hash function.

Unit-IV Digital signatures: ElGamal digital signature scheme, Elliptic Curve digital signature scheme, NISTdigital signature scheme.

Unit-V Key Management and Distribution: Symmetric key distribution, X.509 Certificate public key infrastructures.

Unit-VI Cyber Security: Cyberspace, cyber crimes, cybercriminals, cyber security, cyber security threats, DOS attack, vulnerability analysis, firewall, intrusion detection system.

Readings:

1. W. Stalling, **Cryptography and Network Security Principles and Practices,** 7th Edition, Pearson Education of India, 2018.
2. A. J. Elbirt, **Understanding and Applying Cryptography and Data Security**, CRC Press, Taylor Francis Group, New York, 2015.
3. C. Pfleeger and S. L. Pfleeger, Jonathan Margulies, **Security in Computing** (5th Edition,, Prentice-Hall of India, 2015.
4. Peter W. Singer and Allan Friedman, **Cybersecurity and Cyberwar**, Oxford University Press, 2014.
5. Jonathan Clough, **Principles of Cybercrime**, Cambridge University Press, 2015.

MCAC302: SOFTWARE ENGINEERING [3-0-1]

Course Objectives: The course targets developing skills that would enable the students to construct high-quality software by applying the established software design practices for analysis, design, implementation, testing and maintenance.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe the software engineering layered technology and compare software

process models.

CO2: apply agile development methods for developing software.

CO3: describe software/system requirements and the processes involved in the discovery and documentation of these requirements.

CO4: apply system modelling techniques and object-oriented design for software development.

CO5: test software using verification and validation, static analysis, reviews, inspections, and audits.

CO6: describe steps in software project management

CO7: work individually and in a team to develop and deliver quality software.

Syllabus:

Unit-I Software Engineering: The software crisis, software engineering principles, programming-in-the-small vs programming-in-the-large.

Unit-II Software process: The software lifecycle, the waterfall model and variations, risk-driven approaches, introduction to evolutionary and prototyping approaches, agile process models, and system classifications, Agile approach to software engineering.

Unit-III Project management: Relationship to lifecycle, project planning, project control, project organization, risk management, cost models, configuration management, version control, quality assurance, metrics.

Unit-IV Software requirements: Requirements analysis, functional and non-functional requirements elicitation, analysis tools, requirements definition, requirements specification, static and dynamic specifications, requirements review.

Unit-V Software design: Design for reuse, design for change, design notations, design evaluation and validation.

Unit-VI Implementation and Maintenance: Programming standards and procedures, modularity, data abstraction, static analysis, unit testing, integration testing, regression testing, verification and validation, tools for testing, fault tolerance, the maintenance problem, the nature of maintenance, planning for maintenance**.**

Readings:

1. R.S. Pressman, **Software Engineering: A Practitioner's Approach,** 7th Edition, McGraw-Hill, 2010.
2. I. Sommerville, **Software Engineering,** 10th Edition, Pearson Education, 2015.
3. R. Mall, **Fundamentals of Software Engineering,** 4th Edition, Prentice-Hall of India, 2014.
4. K.K. Aggarwal and Y. Singh, Software Engineering, 3rd Edition, New Age International Publishers, 2008.
5. P. Jalote, An Integrated Approach to Software Engineering, 3rd Edition, Narosa Publishing House, 2005.
6. N.S. Godbole, Software Quality Assurance: Principles and Practice for the new Paradigm, 2nd Edition, Alpha Science, 2007.

MCAC 303: ARTIFICIAL NEURAL NETWORKS AND DEEP LEARNING [3-0-1]

Course Objectives: The student learns various state-of-the-art deep learning algorithms and their applications to solve real-world problems. The student develops skills to design neural network architectures and training procedures using various deep learning platforms and software libraries.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe the feedforward and deep networks.

CO2:design single and multi-layer feed-forward deep networks and tune various hyper-parameters.
CO3: Use GANs to generate models that generate patterns.
CO4: Apply Large Language Models to an NLP task.
CO5: analyze the performance of deep networks.

Syllabus:

Unit-I Introduction: Historical context and motivation for deep learning; basic supervised classification task, optimizing logistic classifier using gradient descent, stochastic gradient descent, momentum, and adaptive subgradient method.

Unit-II Neural Networks: Feedforward neural networks, deep networks, regularizing a deep network, model exploration, and hyperparameter tuning.

Unit-III Convolution Neural Networks: Introduction to convolution neural networks: stacking, striding and pooling, applications like image, and text classification.

Unit-IV Sequence Modeling: Recurrent Nets: Unfolding computational graphs, recurrent neural networks (RNNs), bidirectional RNNs, encoder-decoder sequence to sequence architectures, deep recurrent networks.

Unit-V Autoencoders: Undercomplete autoencoders, regularized autoencoders, sparse autoencoders, denoising autoencoders, representational power, layer, size, and depth of autoencoders, stochastic encoders and decoders.

Unit VI: Generative Adversarial Networks (GANs): Introduction to Generative Adversarial Networks, GAN Architectures (DCGAN, CycleGAN), applications of GANs (Image Generation, Style Transfer)

Unit VII: Large Language Models: Introduction to Natural Language Processing (NLP), Traditional NLP Techniques, transformer architecture, pre-training and fine-tuning language models, ethical considerations and bias in language models, applications of large language models (text generation, sentiment analysis, question answering)

Unit-VIII Structuring Machine Learning Projects: Orthogonalization, evaluation metrics, train/dev/test distributions, size of the dev and test sets, cleaning up incorrectly labelled data, bias and variance with mismatched data distributions, transfer learning, multi-task learning.

Readings:

1. Ian Goodfellow, **Deep Learning**, MIT Press, 2016.
2. Jeff Heaton, **Deep Learning and Neural Networks**, Heaton Research Inc, 2015.
3. Mindy L Hall, Deep Learning, VDM Verlag, 2011.
4. Li Deng (Author), Dong Yu, **Deep Learning: Methods and Applications (Foundations and Trends in Signal Processing)**, Now Publishers Inc, 2009.

MCAE 301: COMPILER DESIGN [3-0-1]

Course Objectives: The course aims to develop the ability to design, develop, and test a functional compiler/ interpreter for a subset of a popular programming language.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: describe how different phases of a compiler work.

CO2: implement top-down and bottom-up parsing algorithms.

CO3: use tools like Lex and Yacc to implement syntax-directed translation.

Syllabus:

Unit- I Lexical and Syntactic Analysis: Review of regular languages, design of a lexical analyzer generator, context-free grammars, syntactic analysis: top-down parsing: recursive descent and predictive parsing, LL(k) parsing; bottom-up parsing: LR parsing, handling ambiguous in bottom-up parsers.

Unit-II Syntax directed translation: Top-down and bottom-up approaches, data types, mixed mode expression; subscripted variables, sequencing statement, subroutines and functions: parameters calling, subroutines with side effects.

Unit-III Code generation, machine dependent and machine-independent optimization techniques.

Readings:

1. A.V. Aho, M. S. Lam, R. Sethi and J. D. Ullman, **Compilers, Principles, Techniques and Tools,** Pearson, 2016.

2. Dick Grune, Kees van Reeuwijk, Henri E .Bal, Ceriel J.H. Jacobs, K Langendoen, **Modern Compiler Design,** Springer, 2012.

MCAE302: NETWORK SCIENCE [3-0-1]

Course Objectives: The course aims to acquaint the students with the graph theory concepts relevant for network science. The students learn dynamics of networks in the context of applications from disciplines like biology, sociology, and economics.

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1: discuss the ubiquity of graph data model.

CO2: identify the structural features of a network

CO3: describe the graph generation models

CO4: identify community structures in networks

CO5: write programs to solve complex network problems

Syllabus:

Unit-I Introduction: Introduction to complex systems and networks, modeling of complex systems, review of graph theory.

Unit-II Network properties: Local and global properties like clustering coefficient, eccentricity; centrality measures for directed and undirected networks.

Unit-III Graph models: Random graph model, Small world network model, Barabasi-Albert (preferential attachment) network model.

Unit-IV Community structure in networks: Communities and community detection in networks, hierarchical algorithms for community detection, modularity based community detection algorithms, label propagation algorithm.

Readings:

1. Mohammed J. Zaki, Wagner Meira Jr.; **Data Mining and Analysis: Fundamental Concepts and Algorithms**, Cambridge University Press, 2014.

2. Albert Barabasi, **Network Science** , Cambridge University Press, 2016.

3. David Easley and Jon Kleinberg, **Networks, Crowds, and Markets: Reasoning About a Highly Connected World**, Cambridge University Press, 2010.

MCAE 303: NATURAL LANGUAGE PROCESSING [3-0-1]

Course Objectives: The course provides a rigorous introduction to the essential components of a Natural Language Processing (NLP) system. The students will learn various statistical, machine learning, and deep learning techniques in NLP and apply them to solve machine translation and conversation problems.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: compare and contrast varius language models.

CO2: compare and contrast various machine translation approaches.

CO3: compare and contrast various text summarization techniques.

CO4: implement an NLP system.

Syllabus:

UNIT I Introduction: Natural Language Processing (NLP), history of NLP, neural networks for NLP, applications: sentiment analysis, spam detection, resume mining, conversation modeling, chat-bots, dialog agents, question processing.

UNIT II Language Modeling and Part of Speech Tagging: Unigram language model, bigram, trigram, n-gram, advanced smoothing for language modeling, empirical comparison of smoothing techniques, applications of language modeling, natural language generation, parts of speech tagging, morphology, named entity recognition.

UNIT III Words and Word Forms: Bag of words, skip-gram, continuous bag-of-words, embedding representations for words lexical semantics, word sense disambiguation, knowledge based and supervised word sense disambiguation.

UNIT IV Text Analysis, Summarization and Extraction: Sentiment mining, text classification, text summarization, information extraction, named entity recognition, relation extraction, question answering in multilingual setting; NLP in information retrieval, cross-lingual IR

UNIT V Machine Translation: Need of machine translation, Problems of machine translation, mt approaches, direct machine translations, rule-based machine translation, knowledge based MT System, Statistical Machine Translation (SMT), parameter learning in SMT (IBM models) using EM, Encoder-decoder architecture, neural machine translation.

Readings:

1. Dan Jurafsky and James H. Martin, **Speech and Language Processing,** Pearson, 2009.
2. Jacob Eisenstein, **Introduction to Natural Language Processing,** MIT Press, 201 .
3. Yoav Goldberg, **Neural Network Methods for Natural Language Processing**. Morgan and Claypool Publisher (2017).
4. Jason Brownlee, **Deep Learning for Natural Language Processing,** Machine Learning Mastery, 2019.
5. Steven Bird, Ewan Klein and Edward Loper, **Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit**, O'Reilly, 2009.

MCAE 304: INFORMATION RETRIEVAL [3-0-1]

Course Objectives: This course aims to equip the students with basic techniques for information retrieval that find use in text analytics. The student will also learn to apply the tools for information extraction.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: describe early developments in IR.

CO2: apply measures for evaluating retrieved information.

CO3: choose appropriate model for document processing.

CO4: apply available tools for information retrieval.

CO5: develop simple information retrieval tools to solve real world problems.

Syllabus:

Unit 1- Introduction: Information, Information need and relevance; The IR system; early developments in IR, user interfaces.

Unit 2- Retrieval and IR Models: Boolean retrieval; term vocabulary and postings list; index construction; ranked and other alternative retrieval models.

Unit 3- Retrieval Evaluation: Notion of precision and recall; precision-recall curve, standard performance measures such as MAP, reciprocal ranks, F-measure, NDCG, rank correlation.

Unit 4- Document Processing: Representation; Vector space model; feature selection; stop words; stemming; notion of document similarity; standard datasets..

Unit 5- Classification and Clustering: Notion of supervised and unsupervised algorithms; naive bayes, nearest neighbour and rochio's algorithms for text classification; clustering methods such as k-means.

Unit-6: Link Analysis: Page Rank, HITs, web crawling. applications.

Readings:

1. R. Baeza-Yaets, B. Ribeiro-Neto, **Modern Information Retrieval: The Concept and Technology behind Search**, Latest Edition, Addison-Wesley, 1999.

2. C. D. Manning, P. Raghvan, H. Schutze, **Introduction to Information Retrieval,** Cambridge University Press, 2008.

3. D. A. Grossman, O. Frieder, **Information Retrieval: Algorithms and Heuristics,** 2nd Ed., Springer, 2004.

4. S. Buettcher, Charles L.A. Clarke, G. V. Carmack, **Information Retrieval: Implementing and Evaluating Search Engines,** MIT Press.

5. B. Croft, D. Metzler, T. Strohman, **Search Engines: Information Retrieval in Practice,** Addison Wesley

MCAE 305: COMBINATORIAL OPTIMIZATION [3-0-1]

Course Objectives: This course aims to introduce the mathematical tools for solving and analyzing combinatorial optimization problems. The course develops skills for the formulation of problems in different computing and data science domains as combinatorial optimization problems.

Course Learning Outcomes:

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

CO1: differentiate between the computational complexities of linear programming (LP) and integer programming (IP).

CO2: construct and solve LP and IP problems for a given application.

CO3: apply polyhedral analysis to develop algorithms.

CO4: use the concept of duality to design exact and approximate algorithms.

Syllabus:

Unit-I Introduction: Optimization problems, neighborhoods, local and global optima, convex sets and functions, simplex method, degeneracy; duality and dual simplex algorithm, computational considerations for the simplex and dual simplex algorithms-Dantzig-Wolfe algorithms.

Unit-II Integer Linear Programming: Cutting plane algorithms, branch and bound technique and approximation algorithms for traveling salesman problem.

Unit-III Graph Algorithms: Primal-Dual algorithm and its application to shortest path (Dijkstra's algorithm, Floyd-Warshall algorithms), max-flow problem (Ford and Fulkerson labeling algorithms), matching problems (bipartite matching algorithm, non-bipartite matching algorithms, bipartite weighted matching-hungarian method for the assignment problem, non-bipartite weighted matching problem), efficient spanning tree algorithms.

Unit-IV Matroids: Independence systems and matroids, duality, matroid intersection.

Readings:

Bernhard Korte and Jens Vygen, **Combinatorial Optimization: Theory and Algorithms**, 6th edition, Springer, 2018.

J. Matousek and B. Gartner, **Understanding and Using Linear Programming**, Springer, 2017..

C.H. Papadimitriou and K.Steiglitz, **Combinatorial Optimization: Algorithms and complexity**, Dover Publications, 1998.

Mokhtar S.Bazaraa, John J. Jarvis and Hanif D. Sherali, **Linear Programming and Network Flows**, 4th Edition, Wiley-Blackwell, 2010.

1. H. A. Taha, **Operations Research: An Introduction**, 8th edition, Pearson Education India, 2014.

MCAE 306: DIGITAL WATERMARKING AND STEGANOGRAPHY [3-0-1]

Course Objectives: The students will learn about the need and various approaches of information hiding, steganography, watermarking models and message coding, watermark security and authentication. The students will develop skills to implement these techniques in real-world scenarios.

Course Learning Outcomes:

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

CO1: implement algorithms for steganography and watermarking operations.

CO2: evaluate steganography and watermarking algorithms for robustness.

CO3: describe various fingerprinting techniques.

Syllabus:

Unit-I Information hiding: Introduction, background, and applications of information hiding: data hiding, applications of data hiding.

Unit-II Steganography: Frameworks of secret communication, security of steganography systems, information hiding in noisy data, adaptive and non-adaptive algorithms, active and malicious attackers, information hiding in written text, invisible communication.

Unit-III Data hiding in still images: LSB encoding, BPCS steganography, lossless data hiding, data hiding by quantification, patchwork, transform domain methods, robust data hiding in JPEG images, frequency domain watermarking detecting malicious tempering, robust wavelet-based watermarking, Kundur-Hatzinakos watermarking, data hiding in binary images, Zhao-koch method, Wu-Lee method, CPT method, TP method, data hiding in fax images.

Unit-IV Watermarking: Introduction, watermarking principals, applications, requirements and algorithmic design issues, evaluation and standards of watermarking.

Unit-V Fingerprinting: introduction, terminology and requirements, classifications, fingerprinting schemes, statistical fingerprinting, and collusion-secure fingerprinting.

Readings:

1. I. J.Cox, M. L.Miller, J. A. Bloom, J. Fridrich, T. Kalker, Digital Watermarking and Stegonagraphy, Morgan Kaufman 2008.
2. F. Y. Shih, Digital Watermarking and Stegonagraphy Fundamentals and Techniques, CRC press 2008.

3. Stefon Katzeubeisser, F. A. Petitolos, Information Hiding Techniques for Stegonagraphy and digital watermarking, Aatech House, London, 2008.

MCAE 307: QUANTUM COMPUTING [3-0-1]

Course Objectives: This course provides a foundation for quantum computing, post-quantum cryptography, and quantum machine learning. It covers the fundamental concepts of quantum mechanics, quantum algorithms, and their applications in various areas, including cryptography, cybersecurity, machine learning, finance, and the energy sector.

Course Learning Outcomes:
Upon successful completion of this course, a student will be able to:

CO1: describe the relevance of quantum mechanics to quantum computing.
CO2: describe and analyze quantum algorithms.
CO3: apply quantum optimization techniques in problem-solving in various areas, including cryptography and machine learning.

Syllabus:

Unit-I Introduction: Mathematical foundations: vectors, vector space, inner product; qubits, introduction to quantum mechanics and its relevance to quantum gates, superposition principle, and entanglement quantum parallelism and interference, no cloning theorem, quantum teleportation.

Unit-II Post-Quantum Security: Deutsch-Jozsa algorithm, Simon's algorithm, Bernstein-Vazirani, RSA algorithm and factorization attack on RSA, Shor's algorithm for integer factorization, Grover's algorithm for unstructured search, hash preimage attack with Grover's algorithm, Quantum Fourier transform and its applications, Harrow–Hassidim–Lloyd (HHL) algorithm, Quantum attack resistant Digital Signatures.

Unit-III Quantum Machine Learning and Optimization: Quantum machine learning (QML) models – QSVM, QNN, QCNN, Quantum Linear Regression, Variational Quantum Classifier (VQC), Quantum k-means clustering; kernel methods, Quantum Boltzmann Machines; Quantum optimization techniques: QAOA, quantum annealing.

Unit-IV: Introduction to Quantum Simulation Tools and Platforms: Google CIRQ, Amazon Braket, IBM Qiskit, Pennylane, Q#, Tensorflow quantum, Tket/pyket, XACC, Project Q, Quantum Development Kit (QDK).

Readings:

1. Elias F. Combarro, Samuel González-Castillo, and Alberto Di Meglio. **A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms,** Packt Publishing Ltd, 2023.
2. Noson S. Yanofsky and Mirco A. Mannucci, **Quantum Computing for Computer Scientists**. Cambridge University Press, 2008.
3. Douglas R. Stinson and Maura B. Paterson. **Cryptography, Theory and Practice,** CRC Press, 2019.
4. Santanu Pattanayak. **Quantum Machine Learning with Python: Using Cirq from Google Research and IBM Qiskit**. Apress, 2021.
5. Santanu Ganguly, **Quantum Machine Learning: An Applied Approach,** Apress, 2021.
6. https://docs.quantum.ibm.com/
7. https://quantumai.google/cirq/experiments/textbook_algorithms

MCAE 308: COMPUTER GRAPHICS [3-0-1]

Course Objectives: The course is designed to introduce students to the basic concepts and theory of computer graphics. The aim is to introduce students regarding various object rendering algorithms, projections, and latest concepts related to virtual reality.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: acquire familiarity with the concepts and relevant mathematics of computer graphics.
CO2: ability to implement various algorithms to scan, convert the basic geometrical primitives, transformations, area filling, clipping.
CO3: describe the importance of viewing and projections.
CO4: ability to design basic graphics application programs.
CO5: familiarize with fundamentals of animation and Virtual reality technologies.
CO6: be able to design applications that display graphic images to given specifications.

Syllabus:

UNIT I. Application Areas of Computer Graphics: Overview of graphics systems and devices. points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. filled area primitives, polygon filling algorithms. curve generation: Bezier and B-Spline Curves.

UNIT II 2-D Geometrical Transforms: Translation, scaling, rotation, reflection and shear transformations composite transforms, transformations between coordinate systems. 2-D Viewing: viewing pipeline, viewing coordinate reference frame, window to viewport coordinate transformation, viewing functions.

UNIT III Line Clipping Algorithms: Cohen-Sutherland and Cyrus Beck Line Clipping Algorithms, Sutherland–Hodgeman polygon clipping algorithm. 3-D object representation: polygon surfaces, quadric surfaces, spline representation.

UNIT IV 3-D Geometric Transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations, 3-D viewing: viewing pipeline, viewing coordinates, view volume, general projection transforms and clipping.

UNIT V Visible Surface Detection Methods: Classification, back-face detection, depth-buffer, scanline, depth sorting, BSP-tree methods, area sub-division and octree methods illumination models and surface rendering methods: basic illumination models, polygon rendering methods computer animation: design of animation sequence, general computer animation functions key frame animation, animation sequence, motion control methods, morphing, mesh warping.

UNIT VI Virtual Reality: Basic concepts, classical components of VR system, types of VR systems, three-dimensional position trackers, navigation and manipulation interfaces, gesture interfaces. input devices, graphical rendering pipeline, haptic rendering pipeline, OpenGI rendering pipeline. applications of virtual reality.

Readings:

1. Donald Hearn, Pauline Baker, Warren Carithers, **Computer Graphics with Open GL**, Prentice Hall, 2010.
2. R. K Maurya, **Computer Graphics with Virtual Reality Systems**, Wiley, 2014

MCAE309: DIGITAL IMAGE PROCESSING [3-0-1]

Course Objectives: The course aims to cover core concepts in digital image processing. The course begins with the image enhancement techniques in the spatial and frequency domain, followed by the image morphological operations such as dilation, erosion, and hit-or-miss transformations. The course also covers image segmentation and image compression.

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1 compare different techniques of image acquisition, enhancement, compression and segmentation.
CO2 choose appropriate feature extraction technique for an application..
CO3 compare and contrast merits of different image compression techniques
CO4 implement various image processing techniques.

Syllabus:

Fundamental Steps in Image Processing: Element of visual perception, a simple image model, sampling and quantization, some basic relationships between pixel, image geometry in 2D, image enhancement in the spatial domain.

Introduction to spatial and frequency methods: Basic gray level transformations, histogram equalization, local enhancement, image subtraction, image averaging, basic spatial, filtering, smoothing spatial filters, sharpening spatial filters.

Introduction to the Fourier transformation: Discrete fourier transformation, fast Fourier transformation, filtering in the frequency domain, correspondence between filtering in the spatial and frequency domain smoothing frequency-domain filters, sharpening frequency-domain filters, homomorphic filtering,

Some basic morphological algorithms: Line detection, edge detection, gradient operator, edge linking and boundary detection, thresholding, region-oriented segmentation, representation schemes like chain codes, polygonal approximations, boundary segments, skeleton of a region.

Introduction to Image Compression: JPEG, MPEG, Wavelets

Readings:

1. Rafael C. Gonzalez and Richard E.Woods, **Digital Image Processing**, Prentice–Hall of India, 2002
2. William K. Pratt, **Digital Image Processing: PIKS Inside** (3rd ed.), John Wiley & Sons, Inc., 2001
3. Bernd Jahne, **Digital Image Processing**, (5th revised and extended edition), Springer, 2002
4. S. Annadurai and R. Shanmugalakshmi, **Fundamentals of Digital Image Processing**, Pearson Education, 2007
5. M.A. Joshi, **Digital Image Processing: An Algorithmic Approach**, Prentice-Hall of India, 2006
6. B. Chanda and D.D. Majumder, **Digital Image Processing and Analysis**, Prentice-Hall of India, 2007

MCAE310 SOCIAL NETWORKS [3-0-1]

Course Objectives: The course aims to equip students with various social network analysis approaches to data collection, cleaning, and pre-processing of network data.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: identify different types of social networks and their characteristics.

CO2: implement and apply various social network analysis techniques, such as, influence maximization, community detection, link prediction, and information diffusion.

CO3: apply network models to understand phenomena such as social influence, diffusion of innovations, and community formation.

Syllabus:

Unit-I: Introduction to Social Network Analysis: Introduction to social network analysis, types of networks, nodes edges, node centrality, betweenness, closeness, eigenvector centrality, network centralization, assortativity, transitivity, reciprocity, similarity, degeneracy and network measure, networks structures, network visualization, tie strength, trust, understanding structure through user attributes and behavior.

Unit-II: Link Analysis and Link Prediction: Applications of link analysis, signed networks, strong and weak ties, link analysis and algorithms, page rank, personalized pagerank, divrank, simrank, pathsim. temporal changes in a network, evaluation link prediction algorithms, heuristic models, probabilistic models, applications of link prediction.

Unit-III: Community Detection: Applications of community detection, types of communities, community detection algorithms, disjoint community detection, overlapping community detection, local community detection, evaluation of community detection algorithms.

Unit-IV: Influence Maximization: Applications of influence maximization, diffusion models, independent cascade model, linear threshold model, triggering model, time-aware diffusion model, non-progressive diffusion model. influence maximization algorithms, simulation-based algorithms, proxy-based algorithms, sketch-based algorithms, community-based influence maximization, and context-aware influence maximization.

Unit-V: Multilayer Social Network: Multilayer social networks, formation of multilayer social networks, heuristic-based approaches, greedy approaches, centrality-based approaches, meta-heuristic approaches, path-based approaches, measuring multilayer social networks.

Readings:

1. Tanmoy Chakraborty, **Social Network Analysis,** Wiley India, 2021.
2. David Knoke and Song Yang, **Social Network Analysis,** SAGE publications, 2019.
3. Mark E. Dickison, Matteo Magnani and Luca Rossi, **Multilayer Social Networks,** Cambridge University Press, 2016.
4. Jennifer Golbeck, **Analyzing the Social Web**, Morgan Kaufmann, 2013.
5. Stanley Wasserman, and Katherine Faust. **Social Network Analysis: Methods and applications,** Cambridge University Press, 2012.
6. M.E.J. Newman, **Networks: An introduction,** Oxford University Press, 2010.
7. Wei Chen, Carlos Castillo and Laks V.S. Lakshmanan, **Information and Influence Propagation in Social Networks**, Springer, 2014
8. Virinchi Srinivas and Pabitra Mitra, **Link Prediction in Social Networks: Role of Power-law Distribution**, Springer International Publishing, 2016

MCAE311: CLOUD COMPUTING [3-0-1]

Course Objectives: This course aims to equip the students with parallel and distributed computing and cloud computing concepts. Students will learn about cloud computing's characteristics, benefits, and historical developments. They will learn cloud computing architecture, service models (IaaS, PaaS, SaaS), deployment models, and emerging paradigms like Edge Computing and Mobile Cloud Computing.

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1: describe cloud computing's characteristics, benefits, and historical developments, including distributed systems and virtualization.
CO2: compare and contrast cloud computing architectures, service models, and deployment models.
CO3: analyze cloud economics, address open challenges.
CO4: discuss emerging paradigms like edge computing and mobile cloud computing
CO5: develop a cloud comping application.

Syllabus:

Unit-I. Introduction: Introduction to parallel and distributed computing, cloud computing: characteristics and benefits; historical developments and evolution of cloud computing: distributed systems, virtualization, Web 2.0, service-oriented computing, utility computing.

Unit-II. Virtualization: Cloud computing reference model, characteristics of virtualized environments, taxonomy of virtualization techniques, virtualization and cloud computing, pros and cons of virtualization, technology examples: Xen: paravirtualization, VMware: full virtualization, Microsoft Hyper-V.

Unit-III: Cloud Computing Architecture and Service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS); Deployment models: Public, Private, Hybrid, Community; IaaS: Introduction to IaaS, Resource Virtualization i.e. Server, Storage and Network virtualization; PaaS: Introduction to PaaS, Cloud platform & Management of Computation and Storage; SaaS: Introduction to SaaS, Cloud Services, Web services, Web 2.0, Web OS; Case studies related to IaaS, PaaS and SaaS, Economics of the cloud.

Unit-IV. Current Topics: Open Challenges in Cloud Computing; Introduction to emerging computing paradigms and research challenges: edge computing, mobile cloud computing, fog computing, etc.; Introduction to IoT cloud; study on simulators related to cloud computing and emerging computing paradigms.

Readings:

1. R. Buyya, C. Vecchiola, S. ThamaraiSelvi, **Mastering Cloud Computing**, McGraw Hill, 2013.

2. B. Sosinsky, **Cloud Computing Bible,** Wiley, 2010

3. K. Hwang, G. C. Fox, J. Dongarra, **Distributed and Cloud Computing: From Parallel Processing to the Internet of Things,** Morgan Kaufmann, 2011

MCAO 301: DATA MINING FOR BUSINESS APPLICATIONS (4+1)

Course Objectives: In this course, the objective is to explore the applications of data mining for business. The course should enable students to translate business problems into data hypotheses. Data wrangling and visualization followed by classification and clustering of data points are other intents of the course. The course also covers mining frequent itemsets from transactional data.

Course Learning Outcomes:

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

CO1: Visualize data from a business perspective
CO2: Carry out data wrangling tasks, like handling missing data and data transformation
CO3: Map a business problem to data mining tasks like classification, clustering, and rule mining, and apply a suitable technique to it.
CO4: Use programming tools for the application of data mining tasks.

Syllabus:

Unit-I Handling Business Data: Translate business problems into data hypotheses, exploring and describing datasets, predictive and descriptive data mining techniques, use of visualizations to generate hypotheses.

Unit-II Data Wrangling: Prepare and clean data for analysis, identify solutions for managing missing data, data transformation, data reduction, discretization.

Unit-III Data Visualization: Generate insight with graphs, design visualizations to express data clearly.

Unit-IV Linear Regression: Identify relationships between variables, explain the linear model.

Unit-V Classification: Logistic regression, decision trees, Naïve Bayes, voting, confusion matrix, interpreting results, selecting a model to guide decisions.

Unit-VI Clustering: Clustering methods: centroid-based clustering, expectation maximization, hierarchical clustering, density-based methods, cluster validation methods, and metrics.

Unit-VII Association Rule Mining: Frequent item sets, closed and maximal item sets, Apriori algorithm for association rule mining.

Readings:

1. P. Tan, M. Steinbach and V. Kumar, **Introduction to Data Mining**, Addison Wesley, 2016.
2. J Zaki Mohammed and Wagner Meira, **Data Mining and Analysis: Fundamental Concepts and Algorithms,** Cambridge University Press, 2014.
3. Jiawei Han and Micheline Kamber, **Data Mining: Concepts and Techniques** (3rd ed.), Morgan Kaufmann, 2011.
4. Robert Layton, **Learning Data Mining with Python**, Second Edition.

MCAO 302: DATA SCIENCE USING PYTHON[3-0-1]

Course Objectives: The objective of this course is to analyze the data statistically and discover valuable insights from it. The course gives hands-on practice on predictive and descriptive modeling of the preprocessed data. In addition, the student also learns to apply mining association rules from the transactional data and mining text from the document will also be covered during the course.

Course Learning Outcomes:

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

CO1: demonstrate proficiency with statistical analysis of data.
CO2: develop the ability to build and assess data-based models.
CO3: execute statistical analyses and interpret outcomes.
CO4: apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively.

Syllabus:

Unit-I Introduction: Introduction data acquisition, data preprocessing techniques including data cleaning, selection, integration, transformation, and reduction, data mining, interpretation.

Unit-II Statistical data modeling: Review of basic probability theory and distributions, correlation coefficient, linear regression, statistical inference, exploratory data analysis, and visualization.

Unit-III Predictive modeling: Introduction to predictive modeling, decision tree, nearest neighbor classifier, and naïve Bayes classifier, classification performance evaluation, and model selection.

Unit-IV Descriptive Modeling: Introduction to clustering, partitional, hierarchical, and density based clustering (k-means, agglomerative, and DBSCAN), outlier detection, clustering performance evaluation.

Unit-V Association Rule Mining: Introduction to frequent pattern mining and association rule mining, Apriori algorithm, measures for evaluating the association patterns.

Unit-VI Text Mining: Introduction of the vector space model for document representation, term frequency-inverse document frequency (tf-idf) approach for term weighting, proximity measures for document comparison, document clustering, and text classification.

Readings:

1. W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy and iPython, 2nd Ed., O'Reilly, 2017.
2. P. Tan, M. Steinbach, A Karpatne, and V. Kumar, Introduction to Data Mining, 2nd Edition, Pearson Education, 2018.
3. G. Grolemund, H. Wickham, R for Data Science, 1st Ed., O'Reilly, 2017.

MCAO 303: DATA ANALYSIS AND VISUALIZATION [3-0-1]

Course Objectives: The course develops student's competence in cleaning and analyzing data related to a chosen application. It also aims to develop skills in using various tools for data visualization and choosing the right tool for given data.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:
CO1: use data analysis tools with ease.
CO2: load, clean, transform, merge, and reshape data.
CO3: create informative visualisations and summarise data sets.
CO4: analyse and manipulate time series data.
CO5: solve real world data analysis problems.

Syllabus

Unit 1 Introduction: Introduction to data science, exploratory data analysis and data science process.

motivation for using Python for data analysis, introduction to Python shell, iPython, and Jupyter Notebook; essential Python libraries: NumPy, pandas, matplotlib, SciPy, scikit-learn, statsmodels.

Unit 2 Introduction to Pandas: Arrays and vectorized computation, introductio to Pandas data structures, essential functionality, summarizing and computing descriptive statistics. data loading, storage and file formats. reading and writing data in text format, web scraping, binary data formats, interacting with web APIs, interacting with databases, data cleaning and preparation, handling missing data, data transformation, string manipulation.

Unit 3 Data Wrangling: Hierarchical indexing, combining and merging data sets reshaping and pivoting. data visualization Matplotlib: basics of Matplotlib, plotting with Pandas and Seaborn, other Python visualization tools

Unit 4 Data Aggregation and Group operations: Data grouping, data aggregation, general split-apply-combine, pivot tables and cross tabulation

Unit 5 Time Series Data Analysis: Date and time data types and tools, time series basics, frequencies and shifting, time zone handling, periods and periods arithmetic, resampling and frequency conversion, moving window functions.

Readings:

1. W. McKinney, **Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython**, 2nd edition, O'Reilly Media.
2. C. O'Neil and R. Schutt (2013). **Doing Data Science: Straight Talk from the Frontline,** O'Reilly Media.

MCAE 304: STATISTICAL METHODS [3-0-1]

Course Objectives: To equip students with the skills necessary to apply statistical methods for various applications.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: apply descriptive statistical techniques to summarize and interpret data
CO2: apply inferential statistical methods, including hypothesis testing and confidence interval estimation.
CO3: perform and interpret simple and multiple linear regression analysis
CO4: apply principles of experimental design in the context of a problem

Syllabus:

Unit-1 Introduction: Descriptive statistics: measures of central tendency and variability, representation of data: stem and leaf diagram, histogram, boxplot, and ogive; bar diagram and its variations, Pie charts; probability distributions: discrete and continuous, joint and conditional probability; theory of attributes: coefficient of association and coefficient of colligation.

Unit-II: Statistical Inference: Parameter and statistic; sampling distributions, confidence intervals and margin of error, hypothesis testing; ANOVA, parametric tests: normal test, t-test, f-test; non-parametric tests: Chi-square test for goodness of fit, Mann-Whitney U test, Kruskal-Wallis test.

Unit-III Regression and Classification: Correlation: measure and significance, simple linear regression, multiple linear regression, one-way classification, analysis of variance, two-way classification, analysis of covariance, curvilinear regression, factorial experiments, Spearman's rank correlation coefficient.

Readings:

1. Robert S. Witte and John S. Witte, **Statistics**, John Wiley & Sons Inc; 11th edition, 2021
2. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, **An Introduction to Statistical Learning**, Springer, 2023.
3. G. W. Snedecor, W. G. Cochran, **Statistical Methods**, Iowa State University Press, 1973

4. John A. Rice, **Mathematical Statistics and Data Analysis**, Cengage, 2013

Annexure-2

Summary of Minor Modifications in the M.Sc. Computer Science Program Syllabus

Some minor changes are proposed in the M.Sc. Computer Science program syllabus. These are listed below:

1. There are some intra-semester movements of some courses and minor updates in the names and contents of certain courses, listed below-

S.No.	Existing	Proposed	Change
1.	MCSC 102: Artificial Intelligence (Semester I)	MCSC 102: Artificial Intelligence and Machine Learning (Semester I)	The change reflects an integrated approach to Artificial Intelligence and Machine Learning
2.	MCSE 304: Deep Learning (Semester III)	MCSC 202: Deep Learning (Semester II)	An elective course earlier is now a core course
3.	MCSC 203: Mobile and Satellite Communication Networks (Semester II)	MCSC 203: Internetworking with TCP/IP (Semester II)	A more industry-centric course has been introduced
4.	MCSE 301: Cyber Security (Semester III)	MCSE 301: Cyber-Physical Systems (Semester III)	The change reflects a focus on Physical Systems

2. Core courses on Artificial Neural Networks and Cloud Computing have been introduced. Further, two credit courses on Software Tools and Reading Skills have been introduced.

3. Based on student feedback, elective courses have been introduced on Natural Language Processing, Information Retrieval, Soft Computing, Quantum Computing, Social Networks, and Data Analysis and Visualization.

   Each theory course is a four-credit course and each practical course is a two-credit course. To pass a course, a student should obtain ‘D’ grade or a higher grade.

MASTER OF COMPUTER SCIENCE

2-YEAR FULL TIME PROGRAMME

RULES, REGULATIONS AND COURSE CONTENTS

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF MATHEMATICAL SCIENCES
UNIVERSITY OF DELHI
DELHI-110007
2024
MASTER OF COMPUTER SCIENCE
2-YEAR FULL TIME PROGRAMME

1. **MSc Computer Science Programme Details:**

Programme Objectives (POs):

Master of Computer Science is a full-time, four-semester course that includes one semester of project work in the fourth semester. The objective of the MSc Computer Science programme is to impart quality education in computer science, so that students are well prepared to face the challenges of academic research as well as software development in the IT industry.

Programme Specific Outcomes (PSOs):

PSO1: Prepares the students to take up a career in the highly competitive IT industry with research and development skills.

PSO2: Equips the students with comprehensive knowledge of the current trends in computer science.

PSO3: The choice of courses from a wide list of specialized courses enables the students to choose a career path in research or software development in the IT industry.

Programme Structure

The M.Sc. Computer Science programme is divided into two parts as under. Each part will consist of two semester.

Part-I	First Year	Semester-I	Semester-II
Part-II	Second Year	Semester-III	Semester-IV

Course Credit Summary

Semester	Core Courses			Elective Course			Open Elective Course			Total Credits
	No. of Courses	Credits (L+T+P)	**Total Credits**	No. of Courses	Credits (L+T+P)	**Total Credits**	No. of Courses	Credits (L+T+P)	**Total Credits**	
I	6	16+0+6	**22**	0	0+0	**0**	0	0+0+0	**0**	**22**
II	5	12+0+6	**18**	1	3+0+1	**4**	0	0+0+0	**0**	**22**
III	1	0+0+4	**4**	3	9+0+3	**12**	1	3+0+1	**4**	**20**
IV	Major project	20	**20**	0	0+0	**0**	0	0+0+0	**0**	**20**
Total Credits for the Course			**64**			**16**			**4**	**84**

Part-I Semester I

Semester I					
	Number of core courses	5			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
MCSC101	Design and Analysis of Algorithms	3	0	1	**4**
MCSC102	Artificial Intelligence and Machine Learning	3	0	1	**4**
MCSC103	Information Security	3	0	1	**4**
MCSC104	Mathematical Foundations of Computer Science	3	0	1	**4**
MCSC105	Data Mining	3	0	1	**4**
MCSC106	Software Tools	1	0	1	**2**
	Total credits in core course	22			
	Number of elective courses	0			
	Total credits in elective course	0			
	Number of open electives	0			
	Total credits in elective course	0			
	Total credits in Semester I	**22**			

Part-I Semester II

Semester II					
	Number of core courses	4			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
MCSC201	Artificial Neural Networks	3	0	1	**4**
MCSC202	Deep Learning	3	0	1	**4**
MCSC203	Internetworking with TCP/IP	3	0	1	**4**
MCSC204	Cloud Computing	3	0	1	**4**
MCSC205	Reading Skills	0	0	2	**2**

	Total credits in core course				18
	Number of elective courses				1
		Theory	Tutorial	Practical	**Total**
	Elective 1	3	0	1	**4**
	Total credits in elective courses				4
	Number of open electives				0
	Credits in each open elective	Theory	Tutorial	Practical	**Total**
	Open Elective 1	0	0	0	**0**
	Total credits in open elective				0
	Total credits in Semester II				**22**

List of Elective Courses

List of Electives for Semester II		
Course Code	**Course Title**	**L-T-P**
MCSE201	Digital Image Processing	3-0-1
MCSE202	Compiler Design	3-0-1
MCSE203	Natural Language Processing	3-0-1

Part-II Semester III

Semester III					
	Number of core courses				1
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
MCSC301	Minor Project	0	0	4	**4**
	Total credits in core course				4
	Number of elective courses				3
	Credits in each open elective	Theory	Tutorial	Practical	**Total**

	Elective course 1	3	0	1	**4**
	Elective course 2	3	0	1	**4**
	Elective course 3	3	0	1	**4**
	Total credits in elective courses				12
	Number of open electives				1
	Credits in each open elective	Theory	Tutorial	Practical	**Total**
	Open Elective 1	3	0	1	**4**
	Total credits in open elective				4
	Total credits in Semester III				20

List of Elective Courses

List of Elective Courses for Semester III		
Course Code	**Course Title**	**L-T-P**
MCSE301	Cyber Physical Systems	3-0-1
MCSE302	Graph Theory	3-0-1
MCSE303	Network Science	3-0-1
MCSE304	Information Retrieval	3-0-1
MCSE306	Soft Computing	3-0-1
MCSE307	Quantum Computing	3-0-1
MCSE308	Software Quality Assurance and Testing	3-0-1
MCSE309	Social Networks	3-0-1
List of Open Courses for Semester III		
Course Code	**Course Title**	**L-T-P**
MCSO301	Data Analysis and Visualization	3-0-1
MCSO302	Data Science	3-0-1
XXXXXXX*	Inter-Departmental Elective	X-X-X

L-T-P: Lectures -Tutorials- Practical
*As per the elective offered by the concerned Department.

Part-II Semester IV

Semester IV		
	Number of core courses	1
Course Code	**Course Title**	**Credits in each core course**
MCSC401	Major Project	20
	Total credits in core course	20
	Number of elective courses	0
	Total credits in elective courses	0
	Number of open electives	0
	Total credits in open elective	0
	Total credits in Semester IV	20

3. SCHEME OF EXAMINATION

- English shall be the medium of instruction and examination.
- Examinations shall be conducted at the end of each semester as per the academic calendar notified by the University.
- The scheme of evaluation shall be as follows: the performance of the students will be evaluated based on a comprehensive system of continuous and end-semester evaluation. For each course, there shall be one minor test, assignments/ laboratory work, quizzes, and an end-semester examination: (Mid-Term exam, assignments/practical & laboratory work - 30% weight; end-semester examination - 70% weight), except for practical courses where internal assessment and end-semester examination shall carry 50% weight each. The evaluation of the practical courses will be based on internal assessment and the end-semester evaluation by a board of examiners appointed by the Committee of Courses.
- The students will choose the elective courses out of the list of courses that are offered in a semester. An elective course offered by another department/ center/ institute may be taken subject to the approval of the department. The minor project will be carried out in the department. The major project may be carried out either in the department under the supervision of the teacher(s) to be approved by the Department or in the industry. In case the project is carried out in an organization, a supervisor may also be appointed by the organization. The projects will be evaluated by the internal supervisor, and an external examiner to be appointed by the department on the recommendation of the internal supervisor. The minor and major projects shall be evaluated as follows:

(a) Mid-semester evaluation: 30% weight
(b) End-semester evaluation
(i) Dissertation: 30% weight
(ii) Viva-voce: 40% weight

Examinations for courses shall be conducted only in the respective odd and even semesters, as per the scheme of examinations. Regular as well as Ex-Students shall be permitted to appear/re-appear/improve in courses of odd semesters only at the end of odd semesters and courses of even semesters only at the end of even semesters.

5. PASS PERCENTAGE

In order to pass a course and earn credits prescribed for it, a student must obtain a 'D' grade or a higher grade.

6. PROMOTION CRITERIA

Part I to Part II

To be eligible for promotion to the second year, a student must successfully complete at least 30 credits out of the courses prescribed for semester I and semester II, taken together. A student who fails to get promoted to Part II shall be required to seek fresh admission in part I as per the admission procedure/ University rules.

Eligibility for award of Degree

In order to be eligible for the award of the degree of M.Sc. Computer Science, a student must earn at least 80 credits out of the courses prescribed for part I and part II examinations, taken together.

7. Eligibility and Mode of Admissions and Number of seats in the M. Sc. programme:

To be decided by the University in every academic year.

8. Conversion of Marks into Grades:

Letter Grade	Numerical Grade	Formula	Computation of grade cut off
O (outstanding)	10	$m \geq \bar{X} + 2.5\sigma$	the value of $\bar{X} + 2.5\sigma$ to be taken into account for grade computation will be actual $\bar{X} + 2.5\sigma$ or 90% whichever is lower

A+ (Excellent)	9	$\overline{X} + 2.0\sigma \leq m < \overline{X} + 2.5\sigma$	the value of $\overline{X} + 2.0\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 2.0\sigma$ or 80% whichever is lower
A (Very Good)	8	$\overline{X} + 1.5\sigma \leq m < \overline{X} + 2.0\sigma$	the value of $\overline{X} + 1.5\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 1.5\sigma$ or 70% whichever is lower
B+ (Good)	7	$\overline{X} + 1.0\sigma \leq m < \overline{X} + 1.5\sigma$	the value of $\overline{X} + 1.0\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 1.0\sigma$ or 60% whichever is lower
B (Above Average)	6	$\overline{X} \leq m < \overline{X} + 1.0\sigma$	the value of $\overline{X}$ to be taken into account for grade computation will be actual $\overline{X}$ or 50% whichever is lower
C (Average)	5	$\overline{X} - 0.5\sigma \leq m < \overline{X}$	the value of $\overline{X} - 0.5\sigma$ to be taken into account for grade computation will be actual $\overline{X} - 0.5\sigma$ or 45% whichever is lower
D (Pass)	4	$\overline{X} - 1.0\sigma \leq m < \overline{X} - 0.5\sigma$	the value of $\overline{X} - 1.0\sigma$ to be taken into account for grade computation will be actual $\overline{X} - 1.0\sigma$ or 40% whichever is lower
F (Fail)	0	$\overline{X} - 1.0\sigma > m$	

9. CGPA to Percentage Conversion:

The formula for calculating the final percentage of marks from Cumulative Grade point average (CGPA) will be as per the University rules.

10. DIVISION CRITERIA

A student would be eligible for the award of an M.Sc. degree, provided he/ she earns the required number of credits. Such a student shall be categorized (on the basis of the CGPA to percentage conversion as per University rules) on the basis of the CGPA acquired during Part-I and Part-II examinations taken together, as follows:

a) I Division: 60% or more marks in the aggregate

b)	II Division:	50% or more marks but less than 60% marks in the aggregate.
c)	Pass:	40% or more marks but less than 50% marks in the aggregate.

11. SPAN PERIOD

The span period will be four years from the date of registration in the programme.

12. ATTENDANCE REQUIREMENTS

No candidate shall be considered to have pursued a regular course of study unless he/she has attended 66.67% of the total number of classroom/ tutorial/ lab sessions conducted in each semester during his/her course of study. A student not complying with this requirement shall not be allowed to appear in the semester examinations. However, considering the merit of the case, the Head of the Department may condone the required percentage of attendance by not more than 10 percent during a semester.

13. COURSE CONTENT FOR EACH COURSE

PART - I (SEMESTER - I)

MCSC101: DESIGN AND ANALYSIS OF ALGORITHMS [3-0-1]

Course Objectives

This course is designed to introduce advanced techniques of designing and analyzing algorithms. The course also familiarizes the students with some problems that are too hard to admit fast solutions. Some of the advanced algorithm design techniques provide good solutions to these problems.

Course Learning Outcomes

Upon successful completion of this course, the student will be able to:
CO1: describe advanced techniques to design algorithms like augmentation, randomization, parallelization and use of linear programming.
CO2: Analyse the strengths and weaknesses of different algorithm design techniques.
CO3: Analyze algorithms in a probabilistic framework.
CO4: Demonstrate correctness of algorithms and analyse their time complexity theoretically as well as practically.
CO5: Argue that certain problems are too hard to admit fast solutions and be able to prove their hardness.
CO6: describe approximation algorithms, their utility, and the notion of approximation ratio.

Syllabus:

Review: Review of basic sorting and searching algorithms, greedy algorithms

divide and conquer and dynamic programming.
Augmentation: Maximum flow and min cut problems, matching in bipartite graphs, minimum weight matching.
String Processing: Finite Automata method, KMP.
Randomized algorithms: Introduction to random numbers, randomized Qsort, randomized selection, randomly built BST, randomized min-cut.
Parallel Algorithms: Shared memory model, distributed memory model, speedup. searching, sorting, selection, matrix-vector multiplication, prefix-sum.
Linear Programming: Formulating an LP, feasible region and convex polyhedron, simplex algorithm, LP-rounding to obtain integral solutions, primal-dual algorithm.
Introduction to Complexity Classes: Classes P, NP - verifiability, NP-Hard - reducibility, NP Complete.
Introduction to Approximation Algorithms.

Readings

1. J. Kleinberg and E.Tardos, **Algorithm Design**, Pearson Education India, 1st Edition 2013.,
2. Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, **Algorithms**, Tata McGraw Hill, 1st Edition, 2017.
3. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, **Introduction to Algorithms**, Prentice-Hall of India Learning Pvt. Ltd, 3rd Edition, 2010.
4. Vijay V. Vazirani, **Approximation Algorithm**s, Springer, 2013,.
5. Bernhard Korte and Jens Vygen, **Combinatorial Optimization: Theory and Algorithms**, Springer, 6th edition, 2018..
6. Rajeev Motvani and Prabhat Raghavan, **Randomized Algorithms**, Cambridge University Press, 2004.

MCSC102: ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING [3-0-1]

Course Objectives: Beginning with a comprehensive overview of the AI techniques, the course introduces the supervised and unsupervised machine learning (ML) techniques, alongwith their applications in solving real-world problems. The course also covers evaluation and validation methods for ML models.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: discuss Turing Test, and various methods of knowledge representation as applicable to a given context.

CO2: design and implement supervised and unsupervised machine learning algorithms for real-world applications while understanding the strengths and weaknesses.

CO3: analyse the computational complexity of various machine learning algorithms.

CO4: fine tune machine learning algorithms and evaluate models generated from data.

Syllabus:

Unit-I Introduction to Artificial Intelligence: Evolution of artificial intelligence (AI) as a discipline, definitions and approaches, philosophical issues, AI for all, ethical issues and responsible AI.

Unit-II Introduction to Machine Learning: Hypothesis and target class, bias-variance tradeoff, Occam's razor, approximation and estimation errors, curse of dimensionality, dimensionality reduction, feature scaling, feature selection methods.

Unit-III Regression: Linear regression with one variable, linear regression with multiple variables, gradient descent, logistic regression, polynomial regression, over-fitting, regularization. performance evaluation metrics, validation methods.

Unit-IV Classification: Decision trees, Naive Bayes classifier, perceptron, multilayer perceptron, neural network, back-propagation algorithm, support vector machine, kernel functions.

Unit V Evaluation: Performance evaluation metrics, ROC Curves, validation methods, bias-variance decomposition, model complexity.

Unit-VI Unsupervised Learning: Clustering, distance metrics, mixture models, expectation maximization, cluster validation methods.

Readings:

1. E. Alpaydin, **Introduction to Machine Learning,** MIT press, 2014.
2. T. M. Mitchell, **Machine Learning**, McGraw Hill Education, 2017.
3. Christopher M. Bishop, **Pattern Recognition And Machine Learning**, Springer-Verlag, 2016.
4. Shai Shalev-Shwartz, Shai Ben-David, **Understanding Machine Learning: From Theory to Algorithms,** Cambridge Press, 2014.
5. Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, eds. **Machine learning: An artificial intelligence approach**, Springer Science & Business Media, 2013.

MCSC103: INFORMATION SECURITY [3-0-1]

Course Objectives: The course aims to train the students to maintain the confidentiality, integrity and availability of data. The student learns various data encryption protocols for transmitting data over unsecured channels in a network.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1 describe various security issues.
CO2 implement a symmetric and asymmetric cryptographic methods.
CO3 describe the role and implementation of digital signatures.
CO4 describe security mechanisms like intrusion detection, auditing and logging.

Syllabus:

Overview of Security: Protection versus security; aspects of security– confidentiality, data integrity, availability, privacy; user authentication, access controls, Orange Book Standard.

Security Threats: Program threats, worms, viruses, Trojan horse, trap door, stack and buffer overflow; system threats- intruders; communication threats- tapping and piracy.

Computer Security Models: BLP Model, BIBA Model, HRU Model.

Cryptography: Substitution, transposition ciphers, symmetric-key algorithms: Data Encryption Standard, Advanced Encryption Standard, IDEA, block cipher operation, stream ciphers: RC-4. Public key encryption: RSA, ElGamal. Diffie-Hellman key exchange. Elliptic curve cryptography, Message Authentication Code (MAC), cryptographic hash function.

Digital signatures: ElGamal digital signature scheme, Elliptic Curve digital signature scheme, NIST digital signature scheme.

Key Management and Distribution : Symmetric key distribution, X.509 Certificate public key infrastructures.

Intrusion detection and prevention.

Readings:

1. W. Stalling, **Cryptography and Network Security Principles and Practices** (7^{th} ed.), Pearson of India, 2018.
2. A.J. Elbirt, **Understanding and Applying Cryptography and Data Security**, CRC Press, Taylor Francis Group, New York, 2015.
3. C. Pfleeger and SL Pfleeger, Jonathan Margulies, **Security in Computing** (5th ed.), Prentice-Hall of India, 2015
4. M. Merkow and J. Breithaupt, **Information Security: Principles and Practices**, Pearson Education, 2006.

MCSC104: MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE [3-0-1]

Course Objective:
This course aims at developing the student skills in linear algebra, probability theory, and statistical methods.

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1: perform operations on vectors; represent vectors geometrically; apply vector algebra to solve problems in sub-disciplines of computer science.
CO2: perform operations on matrices and sparse matrices; compute the determinant, rank and eigenvalues of a matrix; apply matrix algebra to solve problems in sub-disciplines of computer science.
CO3: perform data analysis in probabilistic framework

CO4: visualise and model the given problem using mathematical concepts covered in the course

Syllabus:

Vectors: Definition of Vectors, Vector Addition, Dot and Cross Products, Span, Norm of vectors, Orthogonality, geometry of vectors, Application of vectors in document analysis

Matrix Algebra: Matrices as vectors; Matrix-vector, vector-matrix and matrix-matrix multiplications; inner and outer products, triangular matrix, diagonal matrix, systems of linear equations, linear independence, determinant, rank of matrix, eigen values and eigen vectors, matrix transformations, geometry of transformations, applications of matrix algebra in image representation and transformations.

Basic Probability Theory: Sample space and events, probability axioms, conditional probability, Bayes' law

Basic Statistics: Introduction to descriptive and inferential statistics, describing data sets as frequency tables, relative frequency tables and graphs, scatter diagram, grouped data, histograms, ogives; percentiles, box plot, coefficient of variation, skewness, kurtosis.

Distributions: Continuous and discrete random variables, probability density function, probability mass function, distribution function and their properties, mathematical expectation, conditional expectation, uniform (continuous and discrete), Binomial, Poisson, exponential, normal, $\chi 2$ distributions, weak law of large numbers, central limit theorem, Chebyshev's inequality.

Stochastic Processes: Introduction to stochastic process, Markov chain, transition probabilities, birth-death process

Readings:

1. Kishor S. Trivedi, **Probability and Statistics with Reliability, Queuing and Computer Science Applications**, John Wiely, 2016.
2. Sheldon M. Ross, **Probability Models for Computer Science**, Academic Press, 2001.
3. Ernest Davis, **Linear Algebra and Probability for Computer Science Applications**, CRC Press 2012. https://cs.nyu.edu/davise/MathTechniques/index.html
4. Norm Matloff, **From Algorithms to Z-Scores: Probabilistic and Statistical Modeling in Computer Science,** University of California, Davis (Creative Common Licence) http://heather.cs.ucdavis.edu/~matloff/132/PLN/probstatbook/ProbStatBook.pdf

MCSC105: DATA MINING [3-0-1]

Course Objectives: The objective is to introduce the KDD process. The course would enable students to translate real-world problems into predictive and descriptive tasks. The course also covers data cleaning and visualization, supervised and unsupervised mining techniques.

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1: describe data mining algorithms formally.
CO2: play with basic data exploration methods to develop understanding of given data
CO3: identify suitable pre-processing method for a given problem.
CO4: describe different data mining tasks and algorithms.
CO5: use programming tools (e.g. Weka/Python/R etc) for solving data mining tasks.

Syllabus:

Overview: The process of knowledge discovery in databases, predictive and descriptive data mining techniques, and unsupervised learning techniques.

Data preprocessing : Data cleaning, data transformation, data reduction, discretization

Classification: Supervised learning/mining tasks , decision trees, decision rules, Bayesian classification, instance-based methods (nearest neighbor), evaluation and validation methods.

Clustering : Basic issues in clustering, partitioning methods (k-means, expectation maximization), hierarchical methods for clustering, density-based methods, cluster validation methods and metrics

Association Rule Mining: Frequent item set, maximal and closed itemsets, apriori property, apriori algorithm.

Readings:

1. Mohammed J Zaki and Wagner Meira Jr, **Data Mining and Analysis: Fundamental Concepts and Algorithms,** Cambridge University Press, 2014.
2. P. Tan, M. Steinbach and V. Kumar, **Introduction to Data Mining**, Addison Wesley, 2006.
3. Jiawei Han and Micheline Kamber, **Data Mining: Concepts and Techniques** (3rd ed.), Morgan Kaufmann, 2011.
4. Charu C Agrawal, **Data Mining: The Textbook**, Springer, 2015

MCSC106: SOFTWARE TOOLS [1-0-1]

Course Objective:

To develop proficiency in the use of software tools required for project development.

Course Learning Outcomes:

On completing this course, a student will be able to:

CO1: use the command line interface efficiently
CO2: use features of version control systems
CO3: debug and profile code
CO4: manage dependencies

Syllabus:

Shell tools and scripting, editors (Vim), data wrangling, command-line environment, version control (Git), debugging and profiling, metaprogramming: working with daemons, FUSE, backups, APIs, common command-line flags/patterns, window managers, VPNs, Markdown, Booting + Live USBs, Docker, Vagrant, VMs, cloud, OpenStack, notebook programming

Readings:

1. C. Newham, Learning the Bash Shell: **Unix shell programming**. O'Reilly Media, Inc.; 2005.
2. W. Shotts, **The Linux command line: a complete introduction**. No Starch Press; 2019.
3. https://git-scm.com/book/en/v2

PART - I (SEMESTER - II)

MCSC201: ARTIFICIAL NEURAL NETWORKS [3-0-1]

Course Objectives: The course covers state-of-the-art techniques in neural network design, optimization, and specialized architectures. Students will gain hands-on experience in applying advanced neural network models to real-world problems.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: implement and analyze kernel methods, radial-basis function networks, and kernel regression.

CO2: implement and evaluate regularization networks and self-organizing maps.

CO3: develop information-theoretic models for the machine learning tasks.

Syllabus:

Unit I Kernel Methods and Radial-Basis Function Networks: Cover's theorem on the separability of pattern, the interpolation problem, radial-basis-function networks, recursive least-squares estimation of the weight vector, hybrid learning procedure for RBF Networks; interpretations of the Gaussian hidden units, kernel regression and its relation to RBF networks

Unit II Regularization Theory: Hadamard's Conditions for well-posedness, Tikhonov's regularization theory, regularization networks, generalized radial-basis-function networks, the regularized least-squares estimator, estimation of the regularization parameter, manifold regularization, differentiable manifolds, generalized regularization theory, Laplacian regularized least-squares algorithm.

Unit III Self-Organizing Maps: basic feature-mapping models, self-organizing map, properties of the feature map, contextual maps, hierarchical vector quantization, kernel self-organizing map, relationship between kernel SOM and Kullback–Leibler divergence.

Unit IV Information-Theoretic Learning Models: Entropy, maximum-entropy principle, mutual information, copulas, mutual information as an objective function to be optimized, maximum mutual information principle, infomax and redundancy reduction, spatially coherent features, spatially incoherent features, independent-components analysis, sparse coding of natural images and comparison with ICA Coding; Natural-Gradient learning for independent-components analysis, maximum-likelihood estimation for independent components analysis, maximum-entropy learning for blind source separation, maximization of negentropy for independent-components analysis, coherent independent-components analysis, rate distortion theory and information bottleneck, optimal manifold representation of data**.**

Unit V Stochastic Methods Rooted in Statistical Mechanics: Statistical mechanics, Markov chains, Metropolis algorithm, simulated annealing, Gibbs sampling, Boltzmann machine, logistic belief nets, deep belief nets, deterministic annealing, analogy of deterministic annealing with expectation-maximization algorithm

Readings:

1. Simon O. Haykin, **Neural Networks and Learning Machines,** Pearson Education, 3rd Edition, 2016
2. C. M. Bishop, **Pattern Recognition and Machine Learning**, Springer, 2010.

MCSC202: DEEP LEARNING [3-0-1]

Course Objectives: The student learns various state-of-the-art deep learning algorithms and their applications to solve real-world problems. The student develops skills to design neural network architectures and training procedures using various deep learning platforms and software libraries.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: describe the feedforward and deep networks.

CO2:design single and multi-layer feed-forward deep networks and tune various hyper-parameters.

CO3: Use GANs to generate models that generate patterns.

CO4: Apply Large Language Models to an NLP task.

Syllabus:

Unit-I Introduction: Historical context and motivation for deep learning; deep feedforward neural networks, regularizing a deep network, model exploration, and hyperparameter tuning.

Unit-II Convolution Neural Networks: Introduction to convolution neural networks: stacking, striding and pooling, applications like image, and text classification.

Unit-III Sequence Modeling: Recurrent Nets: Unfolding computational graphs, recurrent neural networks (RNNs), bidirectional RNNs, encoder-decoder sequence to sequence architectures, deep recurrent networks.

Unit-IV Autoencoders: Undercomplete autoencoders, regularized autoencoders, sparse autoencoders, denoising autoencoders, representational power, layer, size, and depth of autoencoders, stochastic encoders and decoders.

Unit V: Generative Adversarial Networks (GANs): Introduction to Generative Adversarial

Networks, GAN Architectures (DCGAN, CycleGAN), Applications of GANs (image generation, style transfer)

Unit VI: Large Language Models: Introduction to Natural Language Processing (NLP), traditional NLP Techniques, transformer architecture, pre-training and fine-tuning language models, ethical considerations and bias in language models, applications of Large Language Models (text generation, sentiment analysis, question answering)

Unit-VII Structuring Machine Learning Projects: Orthogonalization, evaluation metrics, train/dev/test distributions, size of the dev and test sets, cleaning up incorrectly labelled data, bias and variance with mismatched data distributions, transfer learning, multi-task learning.

Readings:

1. Ian Goodfellow, **Deep Learning**, MIT Press, 2016.
2. Jeff Heaton, **Deep Learning and Neural Networks**, Heaton Research Inc, 2015.
3. Mindy L Hall, Deep Learning, VDM Verlag, 2011.
4. Li Deng Dong Yu, **Deep Learning: Methods and Applications (Foundations and Trends in Signal Processing)**, Now Publishers Inc, 2009.

MCSC203: INTERNETWORKING WITH TCP/IP [3-0-1]

Course Objectives:

This course introduces architecture, design and behaviors of the Internet and of the TCP/IP suite of protocols. This course will enable students to test and troubleshoot IP-based communications systems. Furthermore, this course will discuss various flow control and congestion control mechanisms of TCP and the principles of IPv6 Addressing, IPv6 and ICMPv6 protocols.

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1: explain the TCP/IP architecture and utility of different layers

CO2: analyze IP addressing requirements, routing architecture and choose appropriate routing methods

CO3: describe the working of internetworking devices and their network configuration;

Syllabus

Unit-I: Introduction: TCP/IP Architecture and IP packet, IP Addressing, subnetting, and subnet routing, Classless Interdomain Routing (CIDR), ARP, fragmentation and reassembly, DHCP, NAT, IPv6.

Unit-II: Transmission Control Protocol: Transmission Control Protocol: UDP and TCP, TCP: three-way handshake, TCP flow control and data transfer, TCP congestion control, RTT-based congestion control for a datacenter.

Unit-III: Advanced Topics: Mobile IP, multicast routing, OpenFlow, SDN, and NFV, network security threats.

Readings:

1. Douglas E Comer, **Internetworking with TCP/IP Principles, Protocol, and Architecture** , Volume I, 6th Edition, Pearson Education, 2015.
2. D. E. Comer and D. L. Stevens, **Internetworking with TCP/IP Volume II: Design, Implementation, and Internals,** Pearson Education India; 3rd edition, 2015.
3. William Stallings, **Data and Computer Communications**, 9th Edition, Pearson Education, 2011

MCSC204: CLOUD COMPUTING [3-0-1]

Course Objectives: This course aims to equip the students with parallel and distributed computing and cloud computing concepts. Students will learn about cloud computing's characteristics, benefits, and historical developments. They will learn cloud computing architecture, service models (IaaS, PaaS, SaaS), deployment models, and emerging paradigms like Edge Computing and Mobile Cloud Computing.

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1: describe cloud computing's characteristics, benefits, and historical developments, including distributed systems and virtualization.
CO2: compare and contrast cloud computing architectures, service models, and deployment models.
CO3: analyze cloud economics, address open challenges.
CO4: discuss emerging paradigms like edge computing and mobile cloud computing
CO5: develop a cloud comping application.

Syllabus:

Unit-I. Introduction: Introduction to parallel and distributed computing, cloud computing: characteristics and benefits; historical developments and evolution of cloud computing: distributed systems, virtualization, Web 2.0, service-oriented computing, utility computing.

Unit-II. Virtualization: Cloud computing reference model, characteristics of virtualized environments, taxonomy of virtualization techniques, virtualization and cloud computing, pros and cons of virtualization, technology examples: Xen: paravirtualization, VMware: full virtualization, Microsoft Hyper-V.

Unit-III: Cloud Computing Architecture and Service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS); Deployment models: Public, Private, Hybrid, Community; IaaS: Introduction to IaaS, Resource Virtualization i.e. Server, Storage and Network virtualization; PaaS: Introduction to PaaS, Cloud platform & Management of Computation and Storage; SaaS: Introduction to SaaS, Cloud Services, Web services, Web 2.0, Web OS; Case studies related to IaaS, PaaS and SaaS, Economics of the cloud.

Unit-IV. Current Topics: Open Challenges in Cloud Computing; Introduction to emerging computing paradigms and research challenges: edge computing, mobile cloud computing, fog computing, etc.; Introduction to IoT cloud; study on simulators related to cloud computing and emerging computing paradigms.

Readings:

1. R. Buyya, C. Vecchiola, S. ThamaraiSelvi, **Mastering Cloud Computing**, McGraw Hill, 2013.

2. B. Sosinsky, **Cloud Computing Bible,** Wiley, 2010

3. K. Hwang, G. C. Fox, J. Dongarra, **Distributed and Cloud Computing: From Parallel Processing to the Internet of Things,** Morgan Kaufmann, 2011

MCSC205 READING SKILLS [0-0-2]

Course Objectives: The course aims to develop an important skills of independent reading.

Course Learning Outcomes:

On completing this course, a student will be able to:

CO1: Develop a habit of independent reading.

CO2: Given a requirement, independently select sources of reading.

CO3: Read and assimilate independently.

This is a self-study course. The students will carry out extensive reading on a topic to be assigned by the department.

MCSE201: DIGITAL IMAGE PROCESSING

Course Objectives: The course aims to cover core concepts in digital image processing. The course begins with the image enhancement techniques in the spatial and frequency domain, followed by the image morphological operations such as dilation, erosion, and hit-or-miss transformations. The course also covers image segmentation and image compression.

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1 compare different techniques of image acquisition, enhancement, compression and segmentation.
CO2 choose appropriate feature extraction technique for an application..

CO3 compare and contrast merits of different image compression techniques
CO4 implement various image processing techniques.

Syllabus:

Fundamental Steps in Image Processing: Element of visual perception, a simple image model, sampling and quantization, some basic relationships between pixel, image geometry in 2D, image enhancement in the spatial domain.

Introduction to spatial and frequency methods: Basic gray level transformations, histogram equalization, local enhancement, image subtraction, image averaging, basic spatial, filtering, smoothing spatial filters, sharpening spatial filters.

Introduction to the Fourier transformation: Discrete fourier transformation, fast Fourier transformation, filtering in the frequency domain, correspondence between filtering in the spatial and frequency domain smoothing frequency-domain filters, sharpening frequency-domain filters, homomorphic filtering,

Some basic morphological algorithms: Line detection, edge detection, gradient operator, edge linking and boundary detection, thresholding, region-oriented segmentation, representation schemes like chain codes, polygonal approximations, boundary segments, skeleton of a region.

Introduction to Image Compression: JPEG, MPEG, Wavelets

Readings:

1. Rafael C. Gonzalez and Richard E.Woods, **Digital Image Processing**, Prentice–Hall of India, 2002

2. William K. Pratt, **Digital Image Processing: PIKS Inside** (3rd ed.), John Wiley & Sons, Inc., 2001

3. Bernd Jahne, **Digital Image Processing**, (5th revised and extended edition), Springer, 2002

4. S. Annadurai and R. Shanmugalakshmi, **Fundamentals of Digital Image Processing**, Pearson Education, 2007

5. M.A. Joshi, **Digital Image Processing: An Algorithmic Approach**, Prentice-Hall of India, 2006

6. B. Chanda and D.D. Majumder, **Digital Image Processing and Analysis**, Prentice-Hall of India, 2007

MCSE202: COMPILER DESIGN

Course Objectives: The course aims to develop the ability to design, develop, and test a functional compiler/ interpreter for a subset of a popular programming language.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: describe how different phases of a compiler work.

CO2: implement top-down and bottom-up parsing algorithms.

CO3: use tools like Lex and Yacc to implement syntax-directed translation.

Syllabus:

Unit- I Lexical and Syntactic Analysis: Review of regular languages, design of a lexical analyzer generator, context-free grammars, syntactic analysis: top-down parsing: recursive descent and predictive parsing, LL(k) parsing; bottom-up parsing: LR parsing, handling ambiguous in bottom-up parsers.

Unit-II Syntax directed translation: Top-down and bottom-up approaches, data types, mixed mode expression; subscripted variables, sequencing statement, subroutines and functions: parameters calling, subroutines with side effects.

Unit-III Code generation, machine dependent and machine-independent optimization techniques.

Readings:

1. A.V. Aho, M. S. Lam, R. Sethi and J. D. Ullman, **Compilers, Principles, Techniques and Tools,** Pearson, 2016.
2. Dick Grune, Kees van Reeuwijk, Henri E .Bal, Ceriel J.H. Jacobs, K Langendoen, **Modern Compiler Design,** Springer, 2012.

MCSE 203: NATURAL LANGUAGE PROCESSING [3-0-1]

Course Objectives: The course provides a rigorous introduction to the essential components of a Natural Language Processing (NLP) system. The students will learn various statistical, machine learning, and deep learning techniques in NLP and apply them to solve machine translation and conversation problems.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: compare and contrast varius language models.

CO2: compare and contrast various machine translation approaches.

CO3: compare and contrast various text summarization techniques.

CO4: implement an NLP system.

Syllabus:

UNIT I Introduction: Natural Language Processing (NLP), history of NLP, neural networks for NLP, applications: sentiment analysis, spam detection, resume mining, conversation modeling, chat-bots, dialog agents, question processing.

UNIT II Language Modeling and Part of Speech Tagging: Unigram language model, bigram, trigram, n-gram, advanced smoothing for language modeling, empirical comparison of smoothing techniques, applications of language modeling, natural language generation, parts of speech tagging, morphology, named entity recognition.

UNIT III Words and Word Forms: Bag of words, skip-gram, continuous bag-of-words, embedding representations for words lexical semantics, word sense disambiguation, knowledge based and supervised word sense disambiguation.

UNIT IV Text Analysis, Summarization and Extraction: Sentiment mining, text classification, text summarization, information extraction, named entity recognition, relation extraction, question answering in multilingual setting; NLP in information retrieval, cross-lingual IR

UNIT V Machine Translation: Need of machine translation, Problems of machine translation, mt approaches, direct machine translations, rule-based machine translation, knowledge based MT System, Statistical Machine Translation (SMT), parameter learning in SMT (IBM models) using EM, Encoder-decoder architecture, neural machine translation.

Readings:

1. Dan Jurafsky and James H. Martin, **Speech and Language Processing,** Pearson, 2009.
2. Jacob Eisenstein, **Introduction to Natural Language Processing,** MIT Press, 201 .
3. Yoav Goldberg, **Neural Network Methods for Natural Language Processing**. Morgan and Claypool Publisher (2017).
4. Jason Brownlee, **Deep Learning for Natural Language Processing,** Machine Learning Mastery, 2019.
5. Steven Bird, Ewan Klein and Edward Loper, **Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit**, O'Reilly, 2009.

PART - II (SEMESTER – III)

MCSC301: MINOR PROJECT [0-0-4]

MCSE301: CYBER PHYSICAL SYSTEMS [3-0-1]

Course Objectives:

The objectives of this course are to introduce students to the modelling of cyber physical systems (CPS). The student will develop skills to plan, implement, and monitor cyber security mechanisms to protect information technology assets.

Course Learning Outcomes (CO):

Upon successful completion of this course, a student will be able to:

CO1: enumerate various network attacks, and describe their sources and mechanisms of prevention.

CO2: describe the need for cyber laws.

CO3: use software tools to simulate and analyze different CPS systems.

CO4: plan, implement, and monitor cyber security mechanisms to protect information technology assets.

Syllabus:

Unit-I: Introduction: Examples of cyber physical systems (CPS) in different domains, Important design aspects and quality attributes of CPS, Finite state machine, Characteristics of high confidence CPS,

Unit-II: Modeling of CPS: Discrete System Modelling, Continuous systems modelling, Extended state machines, Modelling of Hybrid systems, Various classes of Hybrid Systems, Analysis and Verification, Concepts of embedded systems, Input-outputs, Invariants and Temporal Logic, Linear Temporal Logic, Refinement and Equivalence, Model Development, Rechability Analysis and Model Checking, simulations.

Unit-II: Security of CPS: Cyberspace, Internet of things, Cyber Crimes, Cyber Security, Cyber Security Threats, Cyber laws and legislation, Law Enforcement Roles and Responses. Network Threat Vectors, MITM, OWAPS, ARP Spoofing, IP & MAC Spoofing, DNS Attacks, SYN Flooding attacks, UDP ping-pong and Fraggle attacks, TCP port scanning and reflection attacks, DoS, DDOS. Network Penetration Testing Threat assessment, Penetration testing tools, Penetration testing, Vulnerability Analysis, Threat matrices, Firewall and IDS/IPS, Wireless networks, Wireless Fidelity (Wi-Fi), Wireless network security protocols, Nmap, Network fingerprinting, BackTrack, Metasploit.

Readings:

1. R. Rajkumar, D. de. Niz and M. Klein, **Cyber Physical Systems,** Addision-Wesely, 2017

2. Rajiv Alur, **Principles of Cyber-Physical Systems**, MIT Press, 2015.

3. E.A.Lee and S A Shesia, **Embedded system Design: A Cyber-Physical Approach,** Second Edition, MIT Press, 2018

4. A. Platzer, **Logical Foundations of Cyber Physical Systems,** Springer, 2017.

5. Peter W. Singer and Allan Friedman, Cybersecurity and Cyberwar, Oxford University Press, 2014

6. Jonathan Clough, Principles of Cybercrime, Cambridge University Press, 2015

MCSE302: GRAPH THEORY

Course Objectives: This course will introduce the basic concepts of graphs theory, graph properties and formulations of typical graph problems. The student will learn to model real-life problems such as graph coloring and connectivity as graph problems.

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1: model real-life problems using different types of graphs like trees, bipartite graphs and planar graphs.
CO2: identify special graphs like Euler graphs and Hamiltonian graphs.
CO3: identify various forms of connectedness in a graph
CO4: examine different graph-coloring problems and their solutions.
CO5: model simple problems from real life as graph-coloring problems.

Syllabus:

Introduction: Examples of problems in graph theory, adjacency and incidence matrices, isomorphisms, paths, walks, cycles, components, cut-edges, cut-vertices, bipartite graphs, Eulerian graphs, vertex degrees, reconstruction conjecture, extremal problems, degree sequences, directed graphs, de Bruijn cycles, orientations and tournaments.

Trees: Trees and forests, characterizations of trees, spanning trees, radius and diameter, enumeration of trees, Cayley's formula, Prüfer code, counting spanning trees, deletion-contraction, the matrix tree theorem, graceful labelling, minimum spanning trees (Kruskal's algorithm), shortest paths (Dijkstra's algorithm).

Matching and Covers: Matchings, maximal and maximum matchings, M-augmenting paths, Hall's theorem and consequences, Min-max theorems, maximum matchings and vertex covers, independent sets and edge covers,Connectivity, vertex cuts, Edge-connectivity.

Connectivity and Paths: Blocks, k-connected graphs, Menger's theorem, line graphs, network flow problems, flows and source/sink cuts, Ford-Fulkerson algorithm, max-flow min-cut theorem.

Graph Coloring: Vertex colorings, bounds on chromatic numbers, chromatic numbers of graphs constructed from smaller graphs, chromatic polynomials, properties of the chromatic polynomial, the deletion-contraction recurrence.

Planar Graphs: Planar graphs, Euler's formula, Kuratowski's theorem, five and four color theorems.

Readings:

1. Douglas B West, **Introduction to Graph Theory**, Pearson, Second Edition, 2017.
2. Gary Chartrand and Ping Zhang, **Introduction to Graph Theory**, Tata McGraw Hill, 2017.
3. Jonathan L. Gross and Jay Yellen, **Graph Theory and Its Applications**, Chapman Hall (CRC), Second Edition, 2005.

MCSE303: NETWORK SCIENCE

Course Objectives: The course aims to acquaint the students with the graph theory concepts relevant for network science. The students learn dynamics of networks in the context of applications from disciplines like biology, sociology, and economics

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1: discuss the ubiquity of graph data model.
CO2: identify the structural features of a network
CO3: describe the graph generation models
CO4: identify community structures in networks
CO5: write programs to solve complex network problems

Syllabus:

Introduction: Introduction to complex systems and networks, modelling of complex systems, review of graph theory.

Network properties: Clustering coefficient, centrality measures for directed and undirected networks.

Graph models: Random graph model, Small world graph model, Network evolution using preferential attachment

Community structure in networks: Communities and community detection in networks, Hierarchical algorithms for community detection, Modularity based community detection algorithms, Label Propagation algorithm

Readings:

1. Mohammed J. Zaki, Wagner Meira Jr.; **Data Mining and Analysis: Fundamental Concepts and Algorithms,** Cambridge University Press, 2014
2. Albert Barabasi, **Network Science,** Cambridge University Press, 2016
3. M.E. J. Newman, **Networks: An Introduction,** Oxford University Press, 2010.
4. David Easley and Jon Kleinberg, **Networks, Crowds, and Markets: Reasoning About a Highly Connected World**, Cambridge University Press, 2010

MCSE 304: INFORMATION RETRIEVAL [3-0-1]

Course Objectives: This course aims to equip the students with basic techniques for information retrieval that find use in text analytics. The student will also learn to apply the tools for information extraction.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: describe early developments in IR.

CO2: apply measures for evaluating retrieved information.

CO3: choose appropriate model for document processing.

CO4: apply available tools for information retrieval.

CO5: develop simple information retrieval tools to solve real world problems.

Syllabus:

Unit 1- Introduction: Information, Information need and relevance; The IR system; early developments in IR, user interfaces.

Unit 2- Retrieval and IR Models: Boolean retrieval; term vocabulary and postings list; index construction; ranked and other alternative retrieval models.

Unit 3- Retrieval Evaluation: Notion of precision and recall; precision-recall curve, standard performance measures such as MAP, reciprocal ranks, F-measure, NDCG, rank correlation.

Unit 4- Document Processing: Representation; Vector space model; feature selection; stop words; stemming; notion of document similarity; standard datasets..

Unit 5- Classification and Clustering: Notion of supervised and unsupervised algorithms; naive bayes, nearest neighbour and rochio's algorithms for text classification; clustering methods such as k-means.

Unit-6: Link Analysis: Page Rank, HITs, web crawling. applications.

Readings:

1. R. Baeza-Yaets, B. Ribeiro-Neto, **Modern Information Retrieval: The Concept and Technology behind Search**, Latest Edition, Addison-Wesley, 1999.

2. C. D. Manning, P. Raghvan, H. Schutze, **Introduction to Information Retrieval,** Cambridge University Press, 2008.

3. D. A. Grossman, O. Frieder, **Information Retrieval: Algorithms and Heuristics,** 2nd Ed., Springer, 2004.

4. S. Buettcher, Charles L.A. Clarke, G. V. Carmack, **Information Retrieval: Implementing and Evaluating Search Engines,** MIT Press.

5. B. Croft, D. Metzler, T. Strohman, **Search Engines: Information Retrieval in Practice,** Addison Wesley

MCSE306: SOFT COMPUTING [3-0-1]

Course Objectives:

This course provides insights of soft computing frameworks applicable to a wide range of complex applications.

Course Leaning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: discuss the advantages of soft computing approach over conventional approach.

CO2: design hybrid soft techniques for the situation at hand.

CO3: identify suitable soft computing methods to solve complex problems where standard computing procedures are not available or intractable.

Syllabus:

UNIT-I Soft Computing: Introduction of soft computing, soft computing vs. hard computing, various types of soft computing techniques, applications of soft computing, predicate calculus, rules of interference, overview of neural networks, estimating regularization parameter, Kohnen's self-organizing networks, Hopfield network, applications of neural networks.

UNIT-II Fuzzy Logic Computing: Introduction of fuzzy sets and fuzzy reasoning, Basic functions on fuzzy sets, relations, rule based models and linguistic variables, fuzzy controls, fuzzy decision making, , inferencing, defuzzification, fuzzy clustering, fuzzy rule based classifier, applications of fuzzy logic.

UNIT-III Evolutionary Algorithms: introduction to evolutionary algorithms, basic principles of evolutionary algorithms, evolutionary strategies, genetic algorithm, fitness computations, cross over, mutation, evolutionary programming, classifier systems, genetic programming parse trees, variants of GAs, applications, ant colony optimization, particle swarm optimization, artificial bee colony optimization, multi-objective optimization problems (MOOPs), Multi-Objective Evolutionary Algorithm (MOEA), Non-Pareto and Pareto-based approaches to solve MOOPs, applications of MOEAs.

Readings:

1. Simon S. Haykin, **Neural Networks,** Prentice Hall, Third Edition, 2008.
2. B. Yegnanrayana, **Artificial Neural Networks**, Prentice hall of India, 2004.
3. H. J. Zimmermann, **Fuzzy Set Theory and its Application**, 3rd Edition, 2001.
4. J.S.R. Jang, Sun C.T. and Mizutani E, **Neuro-Fuzzy and Soft computing**, Prentice Hall, 1998.
5. Timothy J. Ross, **Fuzzy Logic with Engineering Applications**, McGraw Hill, 1997.
6. D.E. Goldberg, **Genetic Algorithms: Search, Optimization and Machine Learning**, Addison Wesley, 1989.

MCSE307: QUANTUM COMPUTING [3-0-1]

Course Objectives: This course provides a foundation for quantum computing, post-quantum cryptography, and quantum machine learning. It covers the fundamental concepts of quantum mechanics, quantum algorithms, and their applications in various areas, including cryptography, cybersecurity, machine learning, finance, and the energy sector.

Course Learning Outcomes:
Upon successful completion of this course, a student will be able to:

CO1: describe the relevance of quantum mechanics to quantum computing.
CO2: describe and analyze quantum algorithms.

CO3: apply quantum optimization techniques in problem-solving in various areas, including cryptography and machine learning.

Syllabus:

Unit-I Introduction: Mathematical foundations: vectors, vector space, inner product; qubits, introduction to quantum mechanics and its relevance to quantum gates, superposition principle, and entanglement quantum parallelism and interference, no cloning theorem, quantum teleportation.

Unit-II Post-Quantum Security: Deutsch-Jozsa algorithm, Simon's algorithm, Bernstein-Vazirani, RSA algorithm and factorization attack on RSA, Shor's algorithm for integer factorization, Grover's algorithm for unstructured search, hash preimage attack with Grover's algorithm, Quantum Fourier transform and its applications, Harrow–Hassidim–Lloyd (HHL) algorithm, Quantum attack resistant Digital Signatures.

Unit-III Quantum Machine Learning and Optimization: Quantum machine learning (QML) models – QSVM, QNN, QCNN, Quantum Linear Regression, Variational Quantum Classifier (VQC), Quantum k-means clustering; kernel methods, Quantum Boltzmann Machines; Quantum optimization techniques: QAOA, quantum annealing.

Unit-IV: Introduction to Quantum Simulation Tools and Platforms: Google CIRQ, Amazon Braket, IBM Qiskit, Pennylane, Q#, Tensorflow quantum, Tket/pyket, XACC, Project Q, Quantum Development Kit (QDK).

Readings:

1. Elias F. Combarro, Samuel González-Castillo, and Alberto Di Meglio. **A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms,** Packt Publishing Ltd, 2023.
2. Noson S. Yanofsky and Mirco A. Mannucci, **Quantum Computing for Computer Scientists**. Cambridge University Press, 2008.
3. Douglas R. Stinson and Maura B. Paterson. **Cryptography, Theory and Practice,** CRC Press, 2019.
4. Santanu Pattanayak. **Quantum Machine Learning with Python: Using Cirq from Google Research and IBM Qiskit**. Apress, 2021.
5. Santanu Ganguly, **Quantum Machine Learning: An Applied Approach,** Apress, 2021.
6. https://docs.quantum.ibm.com/
7. https://quantumai.google/cirq/experiments/textbook_algorithms

MCSE308: SOFTWARE QUALITY ASSURANCE AND TESTING [3-0-1]

Course Objectives:

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1: describe quality management processes.
CO2: describe the importance of standards in the quality management process and role of SQA function in an organization.
CO3: apply statistical methods and process for software quality assurance
CO4: Compare and contrast different software testing strategies
CO5: model the quantitative quality evaluation of the software products.

Syllabus:

Unit-I Introduction: Concept of software quality, product and process quality, software quality metrics, quality control and total quality management, quality tools and techniques, quality standards, defect management for quality and improvement.

Unit-II Designing software quality assurance system: Statistical methods in quality assurance, fundamentals of statistical process control, process capability, six-sigma quality.

Unit-III Testing: Test strategies, test planning, functional testing, stability testing and debugging techniques.

Unit-IV Reliability: Basic concepts, reliability measurements, predictions and management.

Readings:

1. N.S. Godbole, **Software Quality Assurance: Principles and Practice for the New Paradigm,** Narosa Publishing, 2^{nd} Edition, 2017.
2.G. Gordon Schulmeyer (4th eds.), **Handbook of Software Quality Assurance,** Artech House, 2008.
3. G. O'Regan, **A Practical Approach to Software Quality**, Springer Verlag, 2002.
4. D. Galin, **Quality Assurance: From Theory to Implementation,** Pearson Education, 2004.
5. S. H. Kan, **Metrics and Models in Software Quality Engineering,** 2^{nd} Edition, Pearson Education Inc., 2003.
6. J.D. McGregor and D.A. Sykes, **A Practical Guide to Testing,** Addison-Wesley, 2001.
7. Glenford J. Myers, **The Art of Software Testing,** 2^{nd} Edition, John Wiley, 2004.
8. D. Graham, E.V. Veenendaal, I. Evans and R. Black, **Foundations of Software Testing,** Thomson Learning, 2007.

MCAE310 SOCIAL NETWORKS

Course Objectives: The course aims to equip students with various social network analysis approaches to data collection, cleaning, and pre-processing of network data.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: identify different types of social networks and their characteristics.

CO2: implement and apply various social network analysis techniques, such as, influence maximization, community detection, link prediction, and information diffusion.

CO3: apply network models to understand phenomena such as social influence, diffusion of innovations, and community formation.

Syllabus:

Unit-I: Introduction to Social Network Analysis: Introduction to social network analysis, types of networks, nodes edges, node centrality, betweenness, closeness, eigenvector centrality, network centralization, assortativity, transitivity, reciprocity, similarity, degeneracy and network measure, networks structures, network visualization, tie strength, trust, understanding structure through user attributes and behavior.

Unit-II: Link Analysis and Link Prediction: Applications of link analysis, signed networks, strong and weak ties, link analysis and algorithms, page rank, personalized pagerank, divrank, simrank, pathsim. temporal changes in a network, evaluation link prediction algorithms, heuristic models, probabilistic models, applications of link prediction.

Unit-III: Community Detection: Applications of community detection, types of communities, community detection algorithms, disjoint community detection, overlapping community detection, local community detection, evaluation of community detection algorithms.

Unit-IV: Influence Maximization: Applications of influence maximization, diffusion models, independent cascade model, linear threshold model, triggering model, time-aware diffusion model, non-progressive diffusion model. influence maximization algorithms, simulation-based algorithms, proxy-based algorithms, sketch-based algorithms, community-based influence maximization, and context-aware influence maximization.

Unit-V: Multilayer Social Network: Multilayer social networks, formation of multilayer social networks, heuristic-based approaches, greedy approaches, centrality-based approaches, meta-heuristic approaches, path-based approaches, measuring multilayer social networks.

Readings:

1. Tanmoy Chakraborty, **Social Network Analysis,** Wiley India, 2021.
2. David Knoke and Song Yang, **Social Network Analysis,** SAGE publications, 2019.
3. Mark E. Dickison, Matteo Magnani and Luca Rossi, **Multilayer Social Networks,** Cambridge University Press, 2016.
4. Jennifer Golbeck, **Analyzing the Social Web**, Morgan Kaufmann, 2013.
5. Stanley Wasserman, and Katherine Faust. **Social Network Analysis: Methods and applications,** Cambridge University Press, 2012.
6. M.E.J. Newman, **Networks: An introduction,** Oxford University Press, 2010.
7. Wei Chen, Carlos Castillo and Laks V.S. Lakshmanan, **Information and Influence Propagation in Social Networks**, Springer, 2014
8. Virinchi Srinivas and Pabitra Mitra, **Link Prediction in Social Networks: Role of Power-law Distribution**, Springer International Publishing, 2016

MCSO301: DATA ANALYSIS AND VISUALIZATION [3-0-1]

Course Objectives: The course develops student's competence in cleaning and analyzing data related to a chosen application. It also aims to develop skills in using various tools for data visualization and choosing the right tool for given data.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:
CO1: use data analysis tools with ease.
CO2: load, clean, transform, merge, and reshape data.
CO3: create informative visualisations and summarise data sets.
CO4: analyse and manipulate time series data.
CO5: solve real world data analysis problems.

Syllabus

Unit 1 Introduction: Introduction to data science, exploratory data analysis and data science process. motivation for using Python for data analysis, introduction to Python shell, iPython, and Jupyter Notebook; essential Python libraries: NumPy, pandas, matplotlib, SciPy, scikit-learn, statsmodels.

Unit 2 Introduction to Pandas: Arrays and vectorized computation, introductio to Pandas data structures, essential functionality, summarizing and computing descriptive statistics. data loading, storage and file formats. reading and writing data in text format, web scraping, binary data formats, interacting with web APIs, interacting with databases, data cleaning and preparation, handling missing data, data transformation, string manipulation.

Unit 3 Data Wrangling: Hierarchical indexing, combining and merging data sets reshaping and pivoting. data visualization Matplotlib: basics of Matplotlib, plotting with Pandas and Seaborn, other Python visualization tools

Unit 4 Data Aggregation and Group operations: Data grouping, data aggregation, general split-apply-combine, pivot tables and cross tabulation

Unit 5 Time Series Data Analysis: Date and time data types and tools, time series basics, frequencies and shifting, time zone handling, periods and periods arithmetic, resampling and frequency conversion, moving window functions.

Readings:

1. W. McKinney, **Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython**, 2nd edition, O'Reilly Media.
2. C. O'Neil and R. Schutt (2013). **Doing Data Science: Straight Talk from the Frontline,** O'Reilly Media.

MCSO302: DATA SCIENCE [3-0-1]

Course Objectives: The objective of this course is to analyze the data statistically and discover valuable insights from it. The course gives hands-on practice on predictive and descriptive modeling of data. In addition, the student also learns to apply mining association rules from the transactional data and mining text data.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: demonstrate proficiency with statistical analysis of data.

CO2: develop the ability to build and assess data-based models.

CO3: execute statistical analysis tasks and interpret outcomes.

CO4: apply data science methods to solve problems in real-world contexts and communicate these solutions effectively.

<u>Syllabus:</u>

Unit-I Introduction: Introduction data acquisition, data preprocessing techniques including data cleaning, selection, integration, transformation, and reduction, data mining, interpretation.

Unit-II Statistical data modeling: Review of basic probability theory and distributions, correlation coefficient, linear regression, statistical inference, exploratory data analysis, and visualization.

Unit-III Predictive modeling: Introduction to predictive modeling, decision tree, nearest neighbor classifier, and naïve Bayes classifier, classification performance evaluation, and model selection.

Unit-IV Descriptive Modeling: Introduction to clustering, partitional, hierarchical, and density based clustering (k-means, agglomerative, and DBSCAN), outlier detection, clustering performance evaluation.

Unit-V Association Rule Mining: Introduction to frequent pattern mining and association rule mining, Apriori algorithm, measures for evaluating the association patterns.

Unit-VI Text Mining: Introduction of the vector space model for document representation, term frequency-inverse document frequency (TF-IDF) approach for term weighting, proximity measures for document comparison, document clustering, and text classification.

Readings:

1. W. McKinney, **Python for Data Analysis: Data Wrangling with Pandas, NumPy and iPython,** 2^{nd} Edition, O’Reilly, 2017.
2. P. Tan, M. Steinbach, A Karpatne, and V. Kumar, **Introduction to Data Mining,** 2nd Edition, Pearson Education, 2018.
3. G. Grolemund, H. Wickham, **R for Data Science,** 1^{st} Edition, O’Reilly, 2017.

Annexure-3

Summary of Changes in Programmes under UGCF

Recommended Prerequisites for Computer Science Undergraduate

Some courses laid down prerequisites that were desirable but could be optional. They have been revised keeping in view the background of the students. For example, some students already admitted to the BA program did not study Mathematics at +2 stage, but it is okay for them to take Programming and Data Structures courses. Also, the requisite background is covered as part of the course.

Annexure-3

	Recommended Prerequisites for Computer Science Undergraduate Programmes under UGCF-NEP						
S. No.	**Programme Name**	**Core/ DSE**	**Semester**	**Course Name Existing**	**Course Name Recommended**	**Pre-requisite**	
						Existing	**Recommended**
1	B.Sc.(H) Computer Science	DSC	IV	DSC 11 Database Management Systems	Database Management Systems	DSC01 Programming using Python / A course in Python at plus 2 level, DSC08	NIL
2	B.Sc.(H) Computer Science	DSC	IV	DSC12 Computer Networks	Computer Networks	DSC 04 Object Oriented Programming with C++/ GE1a Programming using C++ / GE1b Programming with Python / DSC 01 Programming using Python / GE3b: Java Programming	NIL
3	B.Sc.(H) Computer Science	DSE/ GE	IV	Network Security		DSC 04 Object Oriented Programming with C++/ GE1a Programming using C++ / GE1b Programming with Python / DSC 01 Programming using Python / GE3b: Java Programming	NIL
4	B.Sc.(H) Computer Science	DSE/ GE	V	Data Mining II	Data Mining II	DSC01 Programming using Python, / GE1b Programming with Python / A1 Programming Fundamentals using Python ,Data Mining-I	Data Mining-I

5	B.Sc.(H) Computer Science	DSC	VI	DSC-16 Artificial Intelligence	Artificial Intelligence	DSC 04 Object Oriented Programming with C++/ GE 1a Programming using C++ / GE1b Programming with Python/ DSC 01 Programming using Python/ GE 3b: Java Programming	Programming using C++/Programming using Python/Object Oriented Programming using Python
6	B.Sc.(H) Computer Science	DSC	VI	DSC-17 Machine Learning	Machine Learning	DSC01 Programming using Python / A course in Python at plus 2 level	Programming using Python/ object Oriented Programming using Python
7	B.Sc.(H) Computer Science	DSE/ GE	VI	Deep Learning	Deep Learning	DSC03 Mathematics for Computing - I, DSC17 Machine Learning	Programming using Python/Object Oriented Programming using Python/Mathematics for Computing
8	BSc. (Physical Sciences/ Mathematical Sciences)	DSC	II	DSC02: Data Structures using C++	Data Structures	Nil	Programming using C++ / Programming using Python
9	BSc. (Physical Sciences/ Mathematical Sciences)	DSC	IV	DSC04: Operating Systems	Operating Systems	DSC 01 Programming using Python/ A course in C/C++/Python at plus 2 level.	Programming using C++ / Python/Java
10	BSc. (Physical Sciences/ Mathematical Sciences)	DSC	VI	DSC06: Computer Networks	Computer Networks	DSC 04 Object Oriented Programming with C++/ GE 1a Programming using C++ / GE1b Programming with Python/ DSC 01 Programming using Python/ GE 3b: Java Programming	NIL
11	B.A. Programmes with Computer Science as Major discipline	DSC	IV	DSC04: Operating Systems	Operating Systems	DSC 01 Programming using Python/ A course in C/C++/Python at plus 2 level	Programming using C++ / Python/Java
12	B.A. Programmes with Computer Science as Non-major discipline	DSC		DSC04: Operating Systems	Operating Systems	DSC 01 Programming using Python/ A course in C/C++/Python at plus 2 level	Programming using C++ / Python/Java

13	B.A. Programmes with Computer Science as Major/Non-major discipline	DSC	VI	DSC06: Computer Networks	Computer Networks	DSC 04 Object Oriented Programming with C++/ GE 1a Programming using C++ / GE1b Programming with Python/ DSC 01 Programming using Python/ GE 3b: Java Programming	NIL
14	B.A. Programmes with Computer Science as Major discipline	DSC -A6	VI	A6: Deep Learning	Deep Learning	A5 Machine Learning	Programming using Python/Object Oriented Programming using Python/Mathematics for Computing
15	GENERIC ELECTIVES (GE-6a)	GE	VI	GE6a Computer Networks	Computer Networks	DSC 04 Object Oriented Programming with C++/ GE 1a Programming using C++ / GE1b Programming with Python/ DSC 01 Programming using Python/ GE 3b Java Programming	NIL
16	GENERIC ELECTIVES (GE-6c)	GE	VI	GE6c: Artificial Intelligence	Artificial Intelligence	DSC 04 Object Oriented Programming with C++/ GE 1a Programming using C++ / GE1b Programming with Python/ DSC 01 Programming using Python/ GE 3b: Java Programming	Programming using C++/Programming using Python/Object Oriented Programming using Python
	Note: Eligibility criteria for all courses specified in B.A./ B.Sc. Programme and B.Sc.(H) Computer Science will be **class 12th pass** in place of **class 12th pass with Mathematics,** wherever applicable.						

Annexure-4

Summary of Changes in Programmes under UGCF

Programmes under UGCF-NEP
Renaming of Courses having Similar
Content & Proposal for Additional Electives

As recommended by the OSD (Examinations) and discussed in the house, we have renamed Courses having Similar Content. New electives have been proposed based on suggestions from colleagues in the colleges.

Renaming of Courses having Similar Content & Proposal for Additional Electives

The following table reflects the changes wrt courses (Semester I to Semester VI). The table also includes some of the courses that are already approved for Semester I-Semester VI, and are also being proposed for semester VII and VIII and that will have identical names and syllabi and some of the newly proposed electives:

SNo	Programme	Semester	Existing	Proposed
1	B.Sc(H) Computer Science	I	DSC01: Programming using Python	DSC01: Object Oriented Programming using Python
2	B.A. Programme	I	DSC-A1: Programming Fundamentals using Python	DSC-A1: Programming using Python
3	Generic Elective	I	GE1a: Programming with Python	GE1a: Programming using Python
4	B.Sc(H) Computer Science	II	DSC04: Object Oriented Programming with C++	DSC04: Programming using C++
5	B.A. Programme	I	DSC-1: Introduction to Programming using C++	DSC01: Programming using C++
6	B.Sc. Programme	I	Programming Fundamentals using C++	DSC01: Programming using C++
7	Generic Elective	I	GE1b: Programming using C++	GE1b: Programming using C++
8	B.Sc(H) Computer Science	I	DSC02: Computer System Architecture	DSC02: Computer System Architecture
9	B.A. Programme	III	DSC03: Computer System Architecture	DSC03: Computer System Architecture
10	B.Sc. Programme	III	DSC03: Computer System Architecture	DSC03: Computer System Architecture
11	Generic Elective	II	N.A.	GE2c: Computer System Architecture

12	B.Sc(H) Computer Science	III	DSC07: Data Structures	DSC07: Data Structures
13	B.A. Programme	II	DSC02: Data Structures	DSC02: Data Structures
14	B.Sc Programme	II	DSC02: Data Structures using C++	DSC02: Data Structures
15	Generic Elective	IV	GE4a: Data Structures using C++	GE4a: Data Structures using Python
16	B.A. Programme	II	DSC-A2: Data Analysis and Visualization using Python	DSC-A2: Data Analysis and Visualization using Python
17	Generic Elective	II	GE2a: Data Analysis and Visualization using Python	GE2a: Data Analysis and Visualization using Python
18	Discipline Specific Elective	III	DSE: Data Analysis and Visualization (DAV)	DSE: Data Analysis and Visualization using Python
19	B.Sc(H) Computer Science	IV	DSC11: Database Management Systems	DSC11: Database Management Systems
20	B.A. Programme	V	DSC05: Database Management System	DSC05: Database Management Systems
21	B.Sc. Programme	V	DSC05: Database Management System	DSC05: Database Management Systems
22	Generic Elective	III	GE3a: Database Management Systems	GE3a: Database Management Systems
23	B.Sc(H) Computer Science	III	DSC08: Operating Systems	DSC08: Operating Systems
24	B.A. Programme	IV	DSC04: Operating Systems	DSC04: Operating Systems
25	B.Sc. Programme	IV	DSC04: Operating Systems	DSC04: Operating Systems
26	Generic Elective	V	GE5a: Operating Systems	GE5a: Operating Systems
27	B.Sc(H) Computer Science	IV	DSC12: Computer Networks	DSC12: Computer Networks
28	B.A. Programme	VI	DSC06: Computer Networks	DSC06: Computer Networks
29	B.Sc. Programme	VI	DSC06: Computer Networks	DSC06: Computer Networks

30	Generic Elective	VI	Sem VI: GE6a: Computer Networks	Sem VI: GE6a: Computer Networks
31	B.Sc(H) Computer Science	IV	DSC10: Design and Analysis of Algorithms	DSC10: Design and Analysis of Algorithms
32	B.A. Programme	VII	DSC07: Design and Analysis of Algorithms	DSC07: Design and Analysis of Algorithms
33	B.Sc. Programme	VII	DSC07: Design and Analysis of Algorithms	DSC07: Design and Analysis of Algorithms
34	Generic Elective	VII	GE7a: Design and Analysis of Algorithms	GE7a: Design and Analysis of Algorithms
35	B.A. Programme	III	DSC-A3: Data Mining-I	DSC-A3: Data Mining-I
36	Discipline Specific Elective	IV	DSE: Data Mining I	DSE: Data Mining-I
37	B.A. Programme	IV	DSC-A4: Data Mining-II	DSC-A4: Data Mining-II
38	Discipline Specific Elective	V	DSE: Data Mining-II	DSE: Data Mining-II
39	B.A. Programme	IV	GE4b: Introduction to Web Programming	GE4b: Introduction to Web Programming
40	Discipline Specific Elective	V	DSE: Introduction to Web Programming	DSE: Introduction to Web Programming
41	Generic Elective	VI	N.A.	GE6d: Data Privacy
42	Discipline Specific Elective	V	DSE: Data Privacy	DSE: Data Privacy
43	B.Sc(H) Computer Science	VI	DSC17: Machine Learning	DSC17: Machine Learning
44	B.A. Programme	V	DSC-A5: Machine Learning	DSC-A5: Machine Learning
45	Generic Elective	VII	GE7c: Machine Learning	GE7c: Machine Learning
46	B.Sc(H) Computer Science	VI	DSC16: Artificial Intelligence	DSC16: Artificial Intelligence

47	Generic Elective	VI	GE6c: Artificial Intelligence	GE6c: Artificial Intelligence
48	Discipline Specific Elective	V	N.A.	DSE: Artificial Intelligence
49	B.A. Programme	VII	DSC-A6: Deep Learning	DSC-A6: Deep Learning
50	Discipline Specific Elective	VI	DSE: Deep Learning	DSE: Deep Learning
51	Discipline Specific Elective	VI	N.A.	DSE: Numerical Optimization
52	Generic Elective	VII	N.A.	GE7e: Ethical Hacking
53	Discipline Specific Elective	VI	DSE: Ethical Hacking	DSE: Ethical Hacking
54	Generic Elective	VIII	N.A.	GE8d: Cyber Forensics
55	Discipline Specific Elective	VII	N.A.	DSE: Cyber Forensics
56	B.Sc(H) Computer Science	VIII	DSC20: Information Security	DSC20: Information Security
57	B.A. Programme	VIII	DSC08: Information Security	DSC08: Information Security
58	B.Sc. Programme	VIII	DSC08: Information Security	DSC08: Information Security
59	Generic Elective	VIII	GE8a: Information Security	GE8a: Information Security
60	Generic Elective	VIII	GE8c: Introduction to Parallel Programming	GE8c: Introduction to Parallel Programming
61	Discipline Specific Elective	VI	N.A.	DSE: Introduction to Parallel Programming
62	B.Sc(H) Computer Science	VI	DSC18: Cloud Computing	DSC18: Cloud Computing
63	Generic Elective	VII	N.A.	GE7d: Cloud Computing
64	Discipline Specific Elective	VIII	N.A.	DSE8e: Cloud Computing

Note: N.A. in the fourth column in the above table indicates a newly proposed course.

Syllabus

DSC01: OBJECT ORIENTED PROGRAMMING USING PYTHON

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Object Oriented Programming using Python	4	3	0	1	Class XII pass	Nil

Course Objectives:

This course introduces programming to a novice. Python is used for problem solving. The course also introduces the concept of object- oriented programming.

Learning Outcomes:

On successful completion of the course, students will be able to:

• Develop, document, and debug modular Python programs.

• Handle files.

• Apply suitable programming constructs and built-in data structures to solve a problem.

• Use classes and objects in application programs

Syllabus

Theory

Unit 1 Introduction to Programming: Problem solving strategies; Structure of a Python program; Syntax and semantics; Executing simple programs in Python.

Unit 2 Creating Python Programs: Identifiers and keywords; Literals, numbers, and strings; Operators; Expressions; Input/output statements; Defining functions; Control structures (conditional statements, loop control statements, break, continue and pass, exit function), default arguments.

Unit 3 Built-in data structures: Mutable and immutable objects; Strings, built-in functions for string, string traversal, string operators and operations; Lists creation, traversal, slicing and splitting operations, passing list to a function; Tuples, sets, dictionaries and their operations.

Unit 4 Object Oriented Programming: Introduction to classes, objects and methods; Standard libraries.

Unit 5 File and exception handling: File handling through libraries; Errors and exception handling.

Practical **(30 hours)**

List of Practicals

1. Write a program to find the roots of a quadratic equation
2. Write a program to accept a number 'n' and
   a. Check if 'n' is prime
   b. Generate all prime numbers till 'n'
   c. Generate first 'n' prime numbers This program may be done using functions
3. Write a program to create a pyramid of the character '*' and a reverse pyramid
4. Write a program that accepts a character and performs the following:
   a. print whether the character is a letter or numeric digit or a special character.
   b. if the character is a letter, print whether the letter is uppercase or lowercase
   c. if the character is a numeric digit, prints its name in text (e.g., if input is 9, output is NINE)
5. Write a program to perform the following operations on a string
   a. Find the frequency of a character in a string.
   b. Replace a character by another character in a string.
   c. Remove the first occurrence of a character from a string.
   d. Remove all occurrences of a character from a string.
6. Write a program to swap the first n characters of two strings.
7. Write a function that accepts two strings and returns the indices of all the occurrences of the second string in the first string as a list. If the second string is not present in the first string then it should return -1.
8. Write a program to create a list of the cubes of only the even integers appearing in the input list (may have elements of other types also) using the following:

a. 'for' loop
b. list comprehension

9. Write a program to read a file and
   a. Print the total number of characters, words and lines in the file.
   b. Calculate the frequency of each character in the file. Use a variable of dictionary type to maintain the count.
   c. Print the words in reverse order.
   d. Copy even lines of the file to a file named ‘File1’ and odd lines to another file named ‘File2’.
10. Write a program to define a class Point with coordinates x and y as attributes. Create relevant methods and print the objects. Also define a method distance to calculate the distance between any two point objects.
11. Write a function that prints a dictionary where the keys are numbers between 1 and 5 and the values are cubes of the keys.
12. Consider a tuple t1=(1, 2, 5, 7, 9, 2, 4, 6, 8, 10). Write a program to perform following operations:
    a. Print half the values of the tuple in one line and the other half in the next line.
    b. Print another tuple whose values are even numbers in the given tuple.
    c. Concatenate a tuple t2=(11,13,15) with t1.
    d. Return maximum and minimum value from this tuple
13. Write a program to accept a name from a user. Raise and handle appropriate exception(s) if the text entered by the user contains digits and/or special characters.

Essential Readings

- Taneja, S., Kumar, N. Python Programming- A modular Approach, 1st edition, Pearson Education India, 2018.
- Balaguruswamy E. Introduction to Computing and Problem Solving using Python, 2nd edition, McGraw Hill Education, 2018.

Suggestive Readings

- Brown, Martin C. Python: The Complete Reference, 2nd edition, McGraw Hill Education, 2018.

- Guttag, J.V. Introduction to computation and programming using Python, 2nd edition, MIT Press, 2016.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC-A1/GE1a: PROGRAMMING USING PYTHON

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Programming using Python	4	3	0	1	Class XII pass	Nil

Course Objectives

This course is designed to introduce programming concepts using Python to students. The course aims to develop structured as well as object-oriented programming skills using Python. The course also aims to achieve competence amongst its students to develop correct and efficient Python programs to solve problems spanning multiple disciplines.

Learning Outcomes

On successful completion of this course, a student will be able to:

- Write simple programs using built-in data types of Python.
- Implement arrays and user defined functions in Python.
- Solve problems spanning multiple disciplines using suitable programming constructs in Python.

Syllabus

Unit 1

Introduction to Python Programming: Problem solving strategies; Structure of a Python program; Syntax and semantics; Python interpreter/shell, indentation; Executing simple programs in Python.

Unit 2

Creating Python Programs: Identifiers and keywords; literals, numbers, and strings; Operators and expressions; Input and output statements; control structures (conditional statements, loop control statements, break, continue and pass), Errors and exception handling.

Unit 3

User Defined Functions: Defining functions, passing arguments and returning values, default arguments

Unit 4

Built-in data structures: Strings, Lists, Tuples, Sets, Dictionaries; their built-in functions, operators and operations.

References

1. Kamthane, A. N., & Kamthane, A.A. *Programming and Problem Solving with Python*, McGraw Hill Education. 2017.

2. Balaguruswamy E., "*Introduction to Computing and Problem Solving using Python*",2nd Edition, McGraw Hill Education, 2018.

3. Taneja, S., Kumar, N. *Python Programming- A modular Approach*. Pearson Education India, 2018.

Additional References

(i) Guttag, J. V. *Introduction to computation and programming using Python*. MIT Press. 2018

(ii) Downey, A. B. *Think Python–How to think like a Computer Scientist* 2nd Edition. O'Reilly 2015

Suggested Practical List

1. Write a program to calculate total marks, percentage and grade of a student. Marks obtained in each of three subjects are to be input by the user. Assign grades according to the following criteria:

   Grade A : if Percentage >=80
   Grade B : if Percentage >=60 and Percentage <80
   Grade C : if Percentage >=40 and Percentage <60
   Grade D : if Percentage <=40

2. Write a program to print factors of a given number.
3. Write a program to add N natural numbers and display their sum.
4. Write a program to print the following conversion table (use looping constructs):

Height(in Feet)	Height(in inches)
5.0ft	60 inches
5.1ft	61.2inches

.
.
.

5.8ft	69.6inches
5.9ft	70.8inches
6.0ft	72inches

5. Write a program that takes a positive integer n and the produce n lines of output as shown:

```
*
* *
* * *
* * * *
```

(for n =4)

6. Write a menu driven program using user defined functions to print the area of rectangle, square, circle and triangle by accepting suitable input from user.
7. Write a function that calculates factorial of a number n.
8. Write a program to print the series and its sum: (use functions)

$1/1! + 1/2! + 1/3! \ldots\ldots. 1/n!$

9. Write a program to perform the following operations on an input string

    a. Print length of the string

    b. Find frequency of a character in the string

    c. Print whether characters are in uppercase or lowercase

10. Write a program to create two lists: one of even numbers and another of odd numbers. The program should demonstrate the various operations and methods on lists.

11. Write a program to create a dictionary where keys are numbers between 1 and 5 and the values are the cubes of the keys.

12. Write a program to create a tuple t1 = (1,2,5,7,2,4). The program should perform the following:

    a. Print tuple in two lines, line 1 containing the first half of tuple and second line having the second half.

    b. Concatenate tuple t2 = (10,11) with t1.

DSC04/DSC01/GE1b: PROGRAMMING USING C++

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre- requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Programming using C++	4	3	0	1	Class XII pass	NIL

Course Objectives:

This course is designed to introduce programming concepts using C++ to students. The course aims to develop structured as well as object-oriented programming skills using C++

programming language. The course also aims to achieve competence amongst its students to develop correct and efficient C++ programs to solve problems spanning multiple domains.

Learning outcomes

On successful completion of the course, students will be able to:

● Write simple programs using built-in data types of C++.

● Implement arrays and user defined functions in C++.

● Write programs using dynamic memory allocation, handling external files, interrupts and exceptions.

● Solve problems spanning multiple domains using suitable programming constructs in C++.

● Solve problems spanning multiple domains using oriented programming concepts in C++.

Syllabus

Unit-1 **(3 hours)**

Introduction to C++: Overview of Procedural and Object-Oriented Programming, Using main() function, Header Files, Compiling and Executing Simple Programs in C++.

Unit-2 **(12 hours)**

Programming Fundamentals: Data types, Variables, Operators, Expressions, Arrays, Keywords, Decision-making constructs, Iteration, Type Casting, Input-output statements, Functions, Command Line Arguments/Parameters

Unit-3 **(15 hours)**

Object Oriented Programming: Concepts of Abstraction, Encapsulation. Creating Classes and objects, Modifiers and Access Control, Constructors, Destructors, Implementation of Inheritance and Polymorphism, Template functions and classes

Unit-4 **(9 hours)**

Pointers and References: Static and dynamic memory allocation, Pointer and Reference Variables, Implementing Runtime polymorphism using pointers and references

Unit-5 **(6 hours)**

Exception and File Handling: Using try, catch, throw, throws and finally; Nested try, creating user defined exceptions, File I/O Basics, File Operations

Practical **(30 hours)**

1. Write a program to compute the sum of the first n terms of the following series: The number of terms n is to be taken from the user through the command line. If the command line argument is not found then prompt the user to enter the value of n.
2. Write a program to remove the duplicates from an array.
3. Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
4. Write a menu driven program to perform string manipulation (without using inbuilt string functions):
   a. Show address of each character in string
   b. Concatenate two strings.
   c. Compare two strings
   d. Calculate length of the string (use pointers)
   e. Convert all lowercase characters to uppercase
   f. Reverse the string
   g. Insert a string in another string at a user specified position
5. Write a program to merge two ordered arrays to get a single ordered array.
6. Write a program to search a given element in a set of N numbers using Binary search
   a. with recursion
   b. without recursion.
7. Write a program to calculate GCD of two numbers
   a. with recursion
   b. without recursion.
8. Create a Matrix class. Write a menu-driven program to perform following Matrix operations (exceptions should be thrown by the functions if matrices passed to them are incompatible and handled by the main() function):
   a. Sum
   b. Product
   c. Transpose
9. Define a class Person having name as a data member. Inherit two classes Student and Employee from Person. Student has additional attributes as course, marks and year and Employee has department and salary. Write display() method in all the three classes to display the corresponding attributes. Provide the necessary methods to show runtime polymorphism.
10. Create a Triangle class. Add exception handling statements to ensure the following conditions: all sides are greater than 0 and sum of any two sides are greater than the

third side. The class should also have overloaded functions for calculating the area of a right angled triangle as well as using Heron's formula to calculate the area of any type of triangle.

11. Create a class Student containing fields for Roll No., Name, Class, Year and Total Marks. Write a program to store 5 objects of Student class in a file. Retrieve these records from the file and display them.
12. Copy the contents of one text file to another file, after removing all whitespaces.

Essential/recommended readings

1. Stephen Prata, C++ Primer Plus, 6th Edition, Pearson India, 2015.
2. E Balaguruswamy, Object Oriented Programming with C++, 8th edition, McGraw- Hill Education, 2020. 3. D.S. Malik, C++ Programming: From Problem Analysis to Program Design, 6th edition, Cengage Learning, 2013.

Suggestive Readings

1. Schildt, H. C++: The Complete Reference, 4th edition, McGraw Hill, 2003
2. Forouzan, A. B., Gilberg, R. F. Computer Science: A Structured Approach using C++, 2nd edition, Cengage Learning, 2010

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC02/DSC03/GE2c: COMPUTER SYSTEM ARCHITECTURE

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Computer System Architecture	4	3	0	1	Class XII pass	NIL

Course Objectives

The objectives of this course are as follows:

• Introduces the students to the fundamental concepts of digital computer organization, design and architecture.
• Develop a basic understanding of the building blocks of the computer system and highlight how these blocks are organized together to architect a digital computer system.

Learning Outcomes

On successful completion of the course, students will be able to:

• Design Combinational Circuits using basic building blocks. Simplify these circuits using Boolean algebra and Karnaugh maps. Differentiate between combinational circuits and sequential circuits.
• Represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.
• Determine various stages of instruction cycle, pipelining and describe interrupts and their handling.
• Describe how CPU communicates with memory and I/O devices.
• Distinguish between different types of processors.
• Simulate the design of a basic computer using a software tool.

Syllabus

Theory

Unit – 1 **(6 hours)**

Digital Logic Circuits

Logic Gates, Truth Tables, Boolean Algebra, Digital Circuits, Combinational Circuits, Introduction to Sequential Circuits, Circuit Simplification using Karnaugh Map, Don't Care Conditions, Flip-Flops, Characteristic Tables, Excitation Table.

Unit – 2 **(9 hours)**

Digital Components (Fundamental building blocks)

Designing of combinational circuits- Half Adder, Full Adder, Decoders, Encoders, Multiplexers, Registers and Memory (RAM, ROM and their types), Arithmetic Microoperations, Binary Adder, Binary Adder-Subtractor.

Unit – 3 **(6 hours)**

Data Representation and Basic Computer Arithmetic

Number System, r and (r-1)’s Complements, data representation and arithmetic operations.

Unit – 4 **(9 hours)**

Basic Computer Organization and Design

Bus organization, Microprogrammed vs Hardwired Control, Instruction Codes, Instruction Format, Instruction Cycle, Instruction pipelining, Memory Reference, Register Reference and Input Output Instructions, Program Interrupt and Interrupt Cycle..

Unit – 5 **(6 hours)**

Processors

General register organization, Stack Organization, Addressing Modes, Overview of Reduced Instruction Set Computer (RISC), Complex Instruction Set Computer (CISC), Multicore processor and Graphics Processing Unit (GPU).

Unit – 6 **(9 hours)**

Memory and Input-Output Organization

Memory hierarchy (main, cache and auxiliary memory), Input-Output Interface, Modes of Transfer: Programmed I/O, Interrupt initiated I/O, Direct memory access.

Essential Readings

• David A. Patterson and John L. Hennessy. “Computer Organization and Design: The Hardware/Software interface”, 5th edition, Elsevier, 2012.

• Mano, M. Computer System Architecture, 3rd edition, Pearson Education, 1993.

Suggestive Readings

• Mano, M. Digital Design, Pearson Education Asia, 1995.

• Null, L., & Lobur, J. The Essentials of Computer Organization and Architecture. 5th edition, (Reprint) Jones and Bartlett Learning, 2018.

• Stallings, W. Computer Organization and Architecture Designing for Performance 8th edition, Prentice Hall of India, 2010

Practicals (30 hours)

(Use Simulator – CPU Sim 3.6.9 or any higher version for the implementation)

1. Create a machine based on the following architecture
2. Create a Fetch routine of the instruction cycle.
3. Write an assembly program to simulate ADD operation on two user-entered numbers.
4. Write an assembly program to simulate SUBTRACT operation on two user-entered numbers.
5. Write an assembly program to simulate the following logical operations on two user-entered numbers.
   AND, OR, NOT, XOR, NOR, NAND
6. Write an assembly program for simulating following memory-reference instructions.
   ADD
   LDA
   STA
   BUN
   ISZ
7. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
   CLA
   CMA
   CME
   HLT
8. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution
   INC
   SPA
   SNA
   SZE

9. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
   CIR
   CIL \

10. Write an assembly program that reads in integers and adds them together; until a negative non-zero number is read in. Then it outputs the sum (not including the last number).
11. Write an assembly program that reads in integers and adds them together; until zero is read in. Then it outputs the sum.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC07/DSC02: DATA STRUCTURES

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical /Practice		
Data Structures	4	3	0	1	Passed 12th class with Mathematics	Programming using C++ / Programming using Python

Course Objectives

The course aims at developing the ability to use basic data structures like arrays, stacks, queues, lists, and trees to solve problems. C++ to be used to implement these data structures.

Learning Outcomes

On successful completion of the course, students will be able to:

● Compare two functions for their rates of growth.

● Discuss abstract specification of data-structures and their implementation.

● Compute time and space complexity of operations on a data-structure.

● Identify the appropriate data structure(s) for a given application and discuss the trade-offs involved in terms of time and space complexity.

● Apply recursive techniques to solve problems.

Syllabus

Unit-1 **(9 hours)**

Growth of Functions, Recurrence Relations: Functions used in analysis, asymptotic notations, asymptotic analysis, recurrence, Master Theorem.

Unit-2 **(16 hours)**

Arrays, Linked Lists, Stacks, Queues: Arrays: array operations, applications, two dimensional arrays, dynamic allocation of arrays; Linked Lists: singly linked lists, doubly linked lists, circularly linked lists, Stacks: implementing stacks using arrays, implementing stacks using linked lists, applications of stacks; Queues: implementing queues using arrays, implementing queues using linked lists,. Time complexity analysis.

Unit-3 **(5 hours)**

Recursion: Recursive functions, linear recursion, binary recursion.

Unit-4 **(6 hours)**

Trees, Binary Trees: Trees: Trees and their properties, tree traversal algorithms, and their time complexity analysis; binary trees: definition and properties, traversal of binary trees, and their time complexity analysis.

Unit-5 **(7 hours)**

Binary Search Trees: Binary Search Trees: insert, delete, search operations, time complexity analysis of these operations

Unit-6 **(2 hours)**

Binary Heap: Binary Heaps: heaps, heap operations.

Essential/recommended readings

1. Goodrich, M.T., Tamassia, R., & Mount, D., Data Structures and Algorithms Analysis in C++, 2nd edition, Wiley, 2011.

2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. Introduction to Algorithms, Prentice Hall of India, 2022.

Additional References

1. Sahni, S. Data Structures, Algorithms and applications in C++, 2nd edition, Universities Press, 2011.

2. Langsam Y., Augenstein, M. J., & Tanenbaum, A. M. Data Structures Using C and C++, Pearson, 2009.

Practical List **(30 hours)**

1. Write a program to implement singly linked list that supports the following operations:
   a. Insert an element x at the beginning of the singly linked list
   b. Insert an element x at ith position in the singly linked list
   c. Remove an element from the beginning of the singly linked list
   d. Remove an element from ith position in the singly link
   e. Search for an element x in the singly linked list
   f. Concatenate two singly linked lists
2. Write a program to implement doubly linked list that supports the following operations:
   a. Insert an element x at the beginning of the doubly linked list
   b. Insert an element x at ith position in the doubly linked list
   c. Insert an element x at the end of the doubly linked list
   d. Remove an element from the beginning of the doubly linked list
   e. Remove an element from ith position in the doubly linked list.
   f. Remove an element from the end of the doubly linked list
   g. Search for an element x in the doubly linked list
   h. Concatenate two doubly linked lists
3. Write a program to implement circular linked list which supports the following operations:
   a. Insert an element x at the front of the circularly linked list
   b. Insert an element x after an element y in the circularly linked list
   c. Insert an element x at the back of the circularly linked list
   d. Remove an element from the back of the circularly linked list
   e. Remove an element from the front of the circularly linked list
   f. Remove the element x from the circularly linked list
   g. Search for an element x in the circularly linked list and return its pointer
   h. Concatenate two circularly linked lists
4. Implement a stack using Arrays.
5. Implement a stack using the Linked List .

6. Write a program to evaluate a prefix/postfix expression using stacks.
7. Implement a circular Queue using arrays.
8. Implement Queue using the Circular Linked List .
9. Write a program to implement Binary Search Tree with the following operations:
   - **a.** Insert an element x
   - **b.** Delete an element x
   - **c.** Search for an element x in the BST and change its value to y and then place the node with value y at its appropriate position in the BST
   - **d.** Display the elements of the BST in preorder, inorder, and postorder traversal
   - **e.** Display the elements of the BST in level-by-level traversal
   - **f.** Display the height of the BST

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

GE4a: DATA STRUCTURES USING PYTHON

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical /Practice		
Data Structures Using Python	4	3	0	1	Passed 12th class with Mathematics	Programming using C++ / Programming using Python

Course Objectives

The course aims at developing the ability to use basic data structures like arrays, stacks, queues, lists, and trees to solve problems.

Learning Outcomes

On successful completion of the course, students will be able to:

● Compare two functions for their rates of growth.

● Discuss abstract specification of data-structures and their implementation.

● Compute time and space complexity of operations on a data-structure.

● Identify the appropriate data structure(s) for a given application and discuss the trade-offs involved in terms of time and space complexity.

● Apply recursive techniques to solve problems.

Syllabus

Unit-1 **(9 hours)**

Growth of Functions, Recurrence Relations: Functions used in analysis, asymptotic notations, asymptotic analysis, recurrence, Master Theorem.

Unit-2 **(16 hours)**

Arrays, Linked Lists, Stacks, Queues: Arrays: array operations, applications, two dimensional arrays, Linked Lists: singly linked lists, doubly linked lists, circularly linked lists, Stacks: sequential and linked implementations, applications of stacks; Queues: sequential and linked implementations, Time complexity analysis.

Unit-3 **(5 hours)**

Recursion: Recursive functions, linear recursion, binary recursion.

Unit-4 **(6 hours)**

Trees, Binary Trees: Trees: Trees and their properties, tree traversal algorithms, and their time complexity analysis; binary trees: definition and properties, traversal of binary trees, and their time complexity analysis.

Unit-5 **(7 hours)**

Binary Search Trees: Binary Search Trees: insert, delete, search operations, time complexity analysis of these operations

Unit-6 **(2 hours)**

Binary Heap: Binary Heaps: heaps, heap operations.

Essential/recommended readings

1. Goodrich M.T., Tamassia R., & Goldwasser M.H., Data Structures and Algorithms in Python, Wiley, 2021.
2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. Introduction to Algorithms, Prentice Hall of India, 2022.
3. Taneja, S. and Kumar, N., Python Programming: A modular approach, Pearson, 2017.

Additional References

1. Drozdek A., Data Structures and Algorithms in Python, 1st Edition, Cengage learning, 2024.

Practical List **(30 hours)**

1. Write a program to implement singly linked list that supports the following operations:
   a. Insert an element x at the beginning of the singly linked list
   b. Insert an element x at ith position in the singly linked list
   c. Remove an element from the beginning of the singly linked list
   d. Remove an element from ith position in the singly link
   e. Search for an element x in the singly linked list
   f. Concatenate two singly linked lists
2. Write a program to implement doubly linked list that supports the following operations:
   a. Insert an element x at the beginning of the doubly linked list
   b. Insert an element x at ith position in the doubly linked list
   c. Insert an element x at the end of the doubly linked list
   d. Remove an element from the beginning of the doubly linked list
   e. Remove an element from ith position in the doubly linked list.
   f. Remove an element from the end of the doubly linked list
   g. Search for an element x in the doubly linked list
   h. Concatenate two doubly linked lists
3. Write a program to implement circular linked list which supports the following operations:
   a. Insert an element x at the front of the circularly linked list
   b. Insert an element x after an element y in the circularly linked list
   c. Insert an element x at the back of the circularly linked list
   d. Remove an element from the back of the circularly linked list
   e. Remove an element from the front of the circularly linked list
   f. Remove the element x from the circularly linked list
   g. Search for an element x in the circularly linked list and return its pointer

    h. Concatenate two circularly linked lists
4. Implement a stack using standard lists in Python.
5. Implement a stack using the Linked List .
6. Write a program to evaluate a prefix/postfix expression using stacks.
7. Implement a circular Queue using standard lists in Python.
8. Implement Queue using the Circular Linked List .
9. Write a program to implement Binary Search Tree with the following operations:
    - **a.** Insert an element x
    - **b.** Delete an element x
    - **c.** Search for an element x in the BST and change its value to y and then place the node with value y at its appropriate position in the BST
    - **d.** Display the elements of the BST in preorder, inorder, and postorder traversal
    - **e.** Display the elements of the BST in level-by-level traversal
    - **f.** Display the height of the BST

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC-A2/GE2a/DSE: DATA ANALYSIS AND VISUALIZATION USING PYTHON

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre- requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Data Analysis and Visualization using Python	**4**	**3**	**0**	**1**	**Class XII pass with Mathematics**	Programming using Python

Course Objectives

This course is designed to introduce the students to real-world data analysis problems, the use of statistics to get a deterministic view of data, and interpreting results in the field of exploratory data science using Python. This course is the first in the “Data Science” pathway and builds the foundation for three subsequent courses in the pathway.

Learning outcomes

On successful completion of the course, students will be able to:

1. Apply descriptive statistics to obtain a deterministic view of data

2. Perform data handling using Numpy arrays

3. Load, clean, transform, merge, and reshape data using Pandas

4. Visualize data using Pandas and matplot libraries

5. Solve real world data analysis problems

SYLLABUS OF DSE

Unit 1 **(10 hours)**

Introduction to basic statistics and analysis:

Fundamentals of Data Analysis, Statistical foundations for Data Analysis, Types of data, Descriptive Statistics, Correlation and covariance, Linear Regression, Statistical Hypothesis Generation and Testing, Python Libraries: NumPy, Pandas, Matplotlib, Seaborn

Unit 2 **(8 hours)**

Array manipulation using Numpy:

Numpy array: Creating Numpy arrays; various data types of Numpy arrays, indexing and slicing, swapping axes, transposing arrays, data processing using Numpy arrays.

Unit 3 **(12 hours)**

Data Manipulation using Pandas:

Data Structures in Pandas: Series, DataFrame, Index objects, Loading data into Pandas data frame, Working with DataFrames: Arithmetics, Statistics, Binning, Indexing, Filtering, Handling missing data, Hierarchical indexing, Data wrangling: Data cleaning, transforming, merging and reshaping

Unit 4 **(8 hours)**

Plotting and Visualization:

Using Matplotlib to plot data: figures, subplots, markings, color and line styles, labels and legends, Plotting functions in Pandas: Line, bar, Scatter plots, histograms, stacked bars, Heatmap, 3D Plotting, interactive plotting using Bokeh and Plotly.

Unit 5 **(7 hours)**

Data Aggregation and Group operations:

Group by mechanics, Data aggregation, General split-apply-combine, Pivot tables and cross tabulation

Essential/recommended readings

1. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython, 2nd edition, O'Reilly Media, 2018.

2. Molin S. Hands-On Data Analysis with Pandas, Packt Publishing, 2019.

3. Gupta S.C., Kapoor V.K. Fundamentals of Mathematical Statistics,12 th edition, Sultan Chand & Sons, 2020.

Additional References

1. Chen D. Y. Pandas for Everyone: Python Data Analysis, First edition, Pearson Education, 2018.

2. Miller J.D. Statistics for Data Science, Packt Publishing Limited, 2017.

Suggested Practical List (If any): (30 Hours)

Practical exercises such as

Use a dataset of your choice from Open Data Portal (https:// data.gov.in/, UCI repository) or load from scikit, seaborn library for the following exercises to practice the concepts learnt.

1. Load a Pandas dataframe with a selected dataset. Identify and count the missing values in a dataframe. Clean the data after removing noise as follows

   a) Drop duplicate rows.
   b) Detect the outliers and remove the rows having outliers
   c) Identify the most correlated positively correlated attributes and negatively correlated attributes

2. Import iris data using sklearn library or (Download IRIS data from: https://archive.ics.uci.edu/ml/datasets/iris or import it from sklearn.datasets)

   i. Compute mean, mode, median, standard deviation, confidence interval and standard error for each feature
   ii. Compute correlation coefficients between each pair of features and plot heatmap
   iii. Find covariance between length of sepal and petal iv. Build contingency table for class feature
3. Load Titanic data from sklearn library , plot the following with proper legend and axis labels: a. Plot bar chart to show the frequency of survivors and non-survivors for male and female passengers separately

   b. Draw a scatter plot for any two selected features
   c. Compare density distribution for features age and passenger fare
   d. Use a pair plot to show pairwise bivariate distribution

4. Using Titanic dataset, do the following

   a. Find total number of passengers with age less than 30
   b. Find total fare paid by passengers of first class
   c. Compare number of survivors of each passenger class

5. Download any dataset and do the following

   a. Count number of categorical and numeric features
   b. Remove one correlated attribute (if any)

   c. Display five-number summary of each attribute and show it visually

Project: Students are encouraged to work on a good dataset in consultation with their faculty and apply the concepts learned in the course.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC11/DSC05/GE3a: DATABASE MANAGEMENT SYSTEMS

Credit distribution, Eligibility and Prerequisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Database Management Systems	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objectives

The course introduces the students to the fundamentals of database management system and its architecture. Emphasis is given on the popular relational database system including data models and data manipulation. Students will learn about the importance of database structure and its designing using conceptual approach using Entity Relationship Model and formal approach using Normalization. The importance of file indexing and controlled execution of transactions will be taught. The course would give students hands-on practice of structured query language in a relational database management system and glimpse of basic database administration commands.

Learning outcomes

On successful completion of the course, students will be able to:

● Use database management system software to create and manipulate the database.

● Create conceptual data models using entity relationship diagrams for modeling real-life situations and designing the database schema.

● Use the concept of functional dependencies to remove redundancy and update anomalies.

● Apply normalization theory to get a normalized database scheme.

● Write queries using relational algebra, a procedural language.

Syllabus

Unit 1 **(5 hours)**

Introduction to Database: Purpose of database system, Characteristics of database approach, data models, database management system, database system architecture, three-schema architecture, components of DBMS, data independence, and file system approach vs database system approach.

Unit 2 **(7 hours)**

Entity Relationship Modeling: Conceptual data modeling - motivation, entities, entity types, attributes, relationships, relationship types, constraints on relationship, Entity Relationship diagram notation.

Unit 3 **(7 hours)**

Relational Data Model: Update anomalies, Relational Data Model - Concept of relations, schema-instance distinction, keys, relational integrity constraints, referential integrity and foreign keys, relational algebra operators and queries.

Unit 4 **(12 hours)**

Structured Query Language (SQL): Querying in SQL, DDL to create database and tables, table constraints, update database-update behaviors, DML, aggregation functions group by and having clauses, retrieve data from the database, generate and query views. Access and manipulate databases using ODBC. Basic Database administration SQL commands.

Unit 5 **(10 hours)**

Database Design: Mapping an Entity Relationship model to relational database, functional dependencies and Normal forms, 1NF, 2NF, 3NF and BCNF decompositions and desirable properties of them.

Unit 6 **(4 hours)**

Data Storage and Indexes: Need of file indexes, file organizations, index structures, single- and multi-level indexing, concurrent execution of transactions, ACID properties,.

Essential/recommended readings

1. Elmasri, R., Navathe, B. S. Fundamentals of Database Systems, 7th Edition, Pearson Education, 2015.

2. Krogh, J. W. MySQL Connector/Python Revealed: SQL and NoSQL Data Storage Using MySQL for Python Programmers, Apress, 2018.

3. Murach J. Murach's MySQL, 3rd edition, Pearson, 2019.

Additional References

1. Ramakrishnan, R., Gehrke J. Database Management Systems, 3rd Edition, McGraw Hill, 2014.

2. Silberschatz, A., Korth, H. F., Sudarshan S. Database System Concepts, 7th Edition, McGraw Hill, 2019.

3. Connolly, T. M., Begg, C. E. Database Systems: A Practical Approach to Design, Implementation, and Management, 6th edition, Pearson, 2019.

Practicals (30 hours)

Create and use the following student-course database schema for a college to answer the given queries using the standalone SQL editor.

Here, Rollno (ADMISSION) and SID (ADMISSION) are foreign keys. Note that course type may have two values viz. Fulltime and Parttime and a student may enroll in any number of courses

1. Retrieve names of students enrolled in any course.
2. Retrieve names of students enrolled in at least one part time course.
3. Retrieve students' names starting with letter 'A'.
4. Retrieve students' details studying in courses 'computer science' or 'chemistry'.
5. Retrieve students' names whose roll no either starts with 'X' or 'Z' and ends with '9'
6. Find course details with more than N students enrolled where N is to be input by the user.
7. Update student table for modifying a student name.
8. Find course names in which more than five students have enrolled
9. Find the name of youngest student enrolled in course 'BSc(P)CS'
10. Find the name of most popular society (on the basis of enrolled students)
11. Find the name of two popular part time courses (on the basis of enrolled students)
12. Find the student names who are admitted to full time courses only.
13. Find course names in which more than 30 students took admission
14. Find names of all students who took admission to any course and course names in which at least one student has enrolled
15. Find course names such that its teacher-in-charge has a name with 'Gupta' in it and the course is full time.
16. Find the course names in which the number of enrolled students is only 10% of its total seats.
17. Display the vacant seats for each course

18. Increment Total Seats of each course by 10%
19. Add enrollment fees paid ('yes'/'No') field in the enrollment table.
20. Update the date of admission for all the courses by 1 year.
21. Create a view to keep track of course names with the total number of students enrolled in it.
22. Count the number of courses with more than 5 students enrolled for each type of course.
23. Add column Mobile number in student table with default value '9999999999'
24. Find the total number of students whose age is > 18 years.
25. Find names of students who are born in 2001 and are admitted to at least one part time course.

Create and use the following student-society database schema for a college to answer the given (sample) queries using the standalone SQL editor.

II. Do the following database administration commands:

Create user, create role, grant privileges to a role, revoke privileges from a role, create index

DSC08/DSC04/GE5a: OPERATING SYSTEMS

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Operating Systems	4	3	0	1	Passed 12th class with Mathematics	Programming using C++ / Python/Java

Course Objectives

The course provides concepts that underlie all operating systems and are not tied to any particular operating system. The emphasis is on explaining the need and structure of an operating system using its common services such as process management (creation, termination etc.), CPU Scheduling, Process Synchronization, Handling Deadlocks, main memory management, virtual memory, secondary memory management. The course also introduces various scheduling algorithms, structures, and techniques used by operating systems to provide these services.

Learning Outcomes

On successful completion of the course, students will be able to:

- Describe the need of an operating system and define multiprogramming and Multithreading concepts.
- Implement the process synchronization service (Critical Section, Semaphores), CPU scheduling service with various algorithms.
- Implement Main memory Management (Paging, Segmentation) algorithms, Handling of Deadlocks
- Identify and appreciate the File systems Services, Disk Scheduling service

Syllabus

Unit-1 **(6 hours)**

Introduction: Operating Systems (OS) definition and its purpose, OS Structure, OS Operations: Dual and Multi-mode, OS as resource manager.

Unit-2 **(9 hours)**

Operating System Structures: OS Services, System Calls: Process Control, File Management, Device Management, and Information Maintenance, Inter-process Communication, and Protection, System programs, OS structure- Simple, Layered, Microkernel, and Modular.

Unit-3 **(10 hours)**

Process Management: Process Concept, States, Process Control Block, Process Scheduling, Schedulers, Context Switch, Operation on processes, Threads, Multicore Programming, Multithreading Models, Process Scheduling Algorithms: First Come First Served, Shortest-Job-First, Priority & Round-Robin, Process Synchronization: The critical section problem, Deadlock characterization, Deadlock handling.

Unit-4 **(11 hours)**

Memory Management: Physical and Logical address space, Swapping, Contiguous memory allocation strategies - fixed and variable partitions, Segmentation, Paging. Virtual Memory Management: Demand Paging and Page Replacement algorithms: FIFO Page Replacement, Optimal Page replacement, LRU page replacement.

Unit-5 **(9 hours)**

File System: File Concepts, File Attributes, File Access Methods, Directory Structure: Single Level, Two-Level, Tree-Structured, and Acyclic-Graph Directories. Mass Storage Structure: Magnetic Disks, Solid-State Disks, Magnetic Tapes, Disk Scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN, LOOK, and C-LOOk Scheduling.

Essential/recommended readings

1. Silberschatz, A., Galvin, P. B., Gagne G. Operating System Concepts, 9 th edition, John Wiley Publications, 2016.

2. Tanenbaum, A. S. Modern Operating Systems, 3 rd edition, Pearson Education, 2007.
3. Stallings, W. Operating Systems: Internals and Design Principles, 9 th edition, Pearson Education, 2018.

Additional References

1. Dhamdhere, D. M., Operating Systems: A Concept-based Approach, 2nd edition, Tata McGraw-Hill Education, 2017.
2. Kernighan, B. W., Rob Pike, R. The Unix Programming Environment, Englewood Cliffs, NJ: Prentice-Hall, 1984.

Practicals

1. Execute various Linux commands for:
   a. Information Maintenance: wc, clear, cal, who, date, pwd
   b. File Management: cat, cp, rm, mv, cmp, comm, diff, find, grep, awk
   c. Directory Management : cd, mkdir, rmdir, ls
2. Execute various Linux commands for:
   a. Process Control: fork, getpid, ps, kill, sleep
   b. Communication: Input-output redirection, Pipe
   c. Protection Management: chmod, chown, chgrp
3. Write a programme (using fork() and/or exec() commands) where parent and child execute:
   a. same program, same code.

    b. same program, different code.
    c. Before terminating, the parent waits for the child to finish its task.
4. Write a program to to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information).
5. Write a program to report behaviour of Linux kernel including information on 19 configured memory, amount of free and used memory. (Memory information)
6. Write a program to copy files using system calls.
7. Use an operating system simulator to simulate operating system tasks.
8. Write a program to implement scheduling algorithms FCFS/ SJF/ SRTF/ non preemptive scheduling algorithms.
9. Write a program to calculate the sum of n numbers using Pthreads. A list of n numbers is divided into two smaller lists of equal size, and two separate threads are used to sum the sublists.
10. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC12/DSC06/GE6a: COMPUTER NETWORKS

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Computer Networks	4	3	0	1	Pass in Class XII	NIL

Course Objectives

The course objectives of this paper are to:

• Understand the concepts behind computer networks and data communication.

• Learn the different types of networks, network topologies and their characteristics.

• Learn the working of protocols used at various layers.

• Understand the utility of different networking devices.

Learning Outcomes

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

• differentiate between various types of computer networks and their topologies.

• understand the difference between the OSI and TCP/IP protocol suit.

• distinguish between different types of network devices and their functions.

• design/implement data link and network layer protocols in a simulated networking environment.

Syllabus

Unit 1 **(8 hours)**

Introduction:

Types of computer networks, Internet, Intranet, network topologies (bus, star, ring, mesh, tree, hybrid topologies), network classifications. layered architecture approach, OSI Reference Model, TCP/IP Reference Model. Transmission Modes: simplex, half duplex and full duplex, network devices and their role.

Unit 2 **(9 hours)**

Physical Layer:

Analog signal, digital signal, the maximum data rate of a channel, transmission media (guided transmission media, wireless transmission, satellite communication), multiplexing (frequency division multiplexing, time-division multiplexing, wavelength division multiplexing). Guided Media (Wired) (Twisted pair, Coaxial Cable, Fiber Optics. Unguided Media (Radio Waves, Infrared, Micro-wave, Satellite).

Unit 3 **(10 hours)**

Data Link and MAC Layer:

Data link layer services, error detection and correction techniques, error recovery protocols (stop and wait, go back n, selective repeat), multiple access protocols with collision detection, MAC addressing, Ethernet..

Unit 4 **(8 hours)**

Network layer:

Networks and Internetworks, virtual circuits and datagrams, addressing, subnetting, Dijkstra Routing algorithm, Distance vector routing, Overview of Network Layer protocols- (ARP, IPV4, ICMP, RARP, IPV6)

Unit 5 **(10 hours)**

Transport and Application Layer:

Process to process Delivery- (client-server paradigm, connectionless versus connection-oriented service); User Datagram Protocols, TCP/IP protocol, Flow Control. FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), Telnet (Remote login protocol), WWW (World Wide Web), HTTP (HyperText Transfer Protocol), URL (Uniform Resource Locator), DNS, DHCP, BOOTP.

Essential/recommended readings

1. Tanenbaum, A.S. & Wethrall, D.J.. Computer Networks, 5th edition, Pearson Education, 2012.

2. Forouzan, B. A.. Data Communication and Networking, 4th edition, McGraw-Hill Education, 2017.

Additional References

1. Comer, D. E.. Computer Networks and Internet, 6th edition, Pearson education, 2015.

2. Stallings, W., Data and Computer Communications, 10th edition, Pearson education India, 2017.

Practicals.

Introduce students to any network simulator tool and do the following:

1. To Study basic network command and Network configuration commands.

2. To study and perform PC to PC communication.

3. To create Star topology using Hub and Switch.

4. To create Bus, Ring, Tree, Hybrid, Mesh topologies.

5. Perform an initial Switch configuration.

6. Perform an initial Router configuration.

7. To implement Client Server Network.

8. To implement connection between devices using a router.

9. To perform remote desktop sharing within LAN connection.

DSC10/DSC07/GE7a: DESIGN AND ANALYSIS OF ALGORITHMS

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Design and Analysis of Algorithms	**4**	**3**	**0**	**1**	**Pass in Class XII**	Data Structures

Course Objectives

The course is designed to develop understanding of different algorithm design techniques and use them for problem solving. The course shall also enable the students to verify correctness of algorithms and analyze their time complexity.

Learning Outcomes

On successful completion of the course, students will be able to:

● Compute and compare the asymptotic time complexity of algorithms.
● Use appropriate algorithm design technique(s) for solving a given problem.

Syllabus

Unit 1 **(8 hours)**

Searching, Sorting, Selection: Linear Search, Binary Search, Insertion Sort, Selection Sort, Bubble Sort, Heapsort, Linear Time Sorting, running time analysis and correctness.

Unit 2 **(5 hours)**

Graphs: Review of graph traversals, graph connectivity, testing bipartiteness, Directed Acyclic Graphs and Topological Ordering, Minimum Spanning Trees.

Unit 3 **(8 hours)**

Divide and Conquer: Introduction to divide and conquer technique, Merge Sort, Quick Sort, Randomised quicksort, Maximum-subarray problem, Strassen's algorithm for matrix multiplication.

Unit 4 **(5 hours)**

Greedy algorithms: Introduction to the Greedy algorithm design approach, application to minimum spanning trees, fractional knapsack problem, and analysis of time complexity.

Unit 5 **(5 hours)**

Dynamic Programming: Introduction to the Dynamic Programming approach, application to subset sum, integer knapsack problems, and analysis of time complexity.

Unit 6 **(4 hours)**

Hash Tables Hash Functions, Collision resolution schemes.

Essential/recommended readings

1. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. Introduction to Algorithms, 4th edition, Prentice Hall of India, 2022.

2. Kleinberg, J., Tardos, E. Algorithm Design, 1st edition, Pearson, 2013.

Additional references

1. Basse, S., Gelder, A. V., Computer Algorithms: Introduction to Design and Analysis, 3rd edition, Pearson, 1999.

Practical List (If any): (30 Hours)

1. Write a program to sort the elements of an array using Insertion Sort (The program should report the number of comparisons).
2. Write a program to sort the elements of an array using Merge Sort (The program should report the number of comparisons).
3. Write a program to sort the elements of an array using Heap Sort (The program should report the number of comparisons).
4. Write a program to multiply two matrices using the Strassen's algorithm for matrix multiplication
5. Write a program to sort the elements of an array using Radix Sort.
6. Write a program to sort the elements of an array using Bucket Sort.
7. Display the data stored in a given graph using the Breadth-First Search algorithm.
8. Display the data stored in a given graph using the Depth-First Search algorithm.
9. Write a program to determine a minimum spanning tree of a graph using the Prim's algorithm.
10. Write a program to implement Dijkstra's algorithm to find the shortest paths from a given source node to all other nodes in a graph.
11. Write a program to solve the weighted interval scheduling problem.
12. Write a program to solve the 0-1 knapsack problem.

For the algorithms at S.No 1, 2 and 3 , test run the algorithm on 100 different input sizes varying from 30 to 1000. For each size find the number of comparisons averaged on 10 different input instances; plot a graph for the average number of comparisons against each input size. Compare it with a graph of nlogn.

DSC-A3/DSE: DATA MINING-I

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course	Eligibility criteria	Pre-requisite of the course (if any)

		Lecture	Tutorial	Practical/ Practice		
Data Mining - I	**4**	**3**	**0**	**1**	**Passed 12th class with Mathematics**	Programming using Python

Course Objectives

This course aims to introduce data mining techniques and their application on real-life datasets. The students will learn to pre-process the dataset and make it ready for application of data mining techniques. The course will focus on three main techniques of data mining i.e. Classification, Clustering and Association Rule Mining. Different algorithms for these techniques will be discussed along with appropriate evaluation metrics to judge the performance of the results delivered.

Learning outcomes

On successful completion of the course, students will be able to:

- Pre-process the data for subsequent data mining tasks
- Apply a suitable classification algorithm to train the classifier and evaluate its performance.
- Apply appropriate clustering algorithm to cluster the data and evaluate clustering quality
- Use association rule mining algorithms and generate frequent item-sets and association rules

Syllabus

Unit 1 **(8 hours)**

Introduction to Data Mining:

Motivation and Challenges for data mining, Types of data mining tasks, Applications of data mining, Data measurements, Data quality, Supervised vs. unsupervised techniques

Unit 2 **(9 hours)**

Data Pre-Processing:

Data aggregation, sampling, dimensionality reduction, feature subset selection, feature creation, variable transformation.

Unit 3 **(11 hours)**

Cluster Analysis:

Basic concepts of clustering, measure of similarity, types of clusters and clustering methods, K-means algorithm, measures for cluster validation, determine optimal number of clusters

Unit 4 **(8 hours)**

Association Rule Mining:

Transaction data-set, frequent itemset, support measure, rule generation, confidence of association rule, Apriori algorithm, Apriori principle

Unit 5 **(9 hours)**

Classification:

Naive Bayes classifier, Nearest Neighbour classifier, decision tree, overfitting, confusion matrix, evaluation metrics and model evaluation.

Essential/recommended readings

1. Tan P.N., Steinbach M, Karpatne A. and Kumar V. Introduction to Data Mining,2nd edition, Pearson, 2021.

2. Han J., Kamber M. and Pei J. Data Mining: Concepts and Techniques, 3 rd edition, 2011, Morgan Kaufmann Publishers.

3. Zaki M. J. and Meira J. Jr. Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd edition, Cambridge University Press, 2020.

Additional References

1. Aggarwal C. C. Data Mining: The Textbook, Springer, 2015.
2. Dunham M. Data Mining: Introductory and Advanced Topics, 1st edition, Pearson Education India, 2006.

Recommended Datasets for :

Classification: Abalone, Artificial Characters, Breast Cancer Wisconsin (Diagnostic)

Clustering: Grammatical Facial Expressions, HTRU2, Perfume data

Association Rule Mining: MovieLens, Titanics

Practicals

1. Apply data cleaning techniques on any dataset (e,g, wine dataset). Techniques may include handling missing values, outliers, inconsistent values. A set of validation rules can be prepared based on the dataset and validations can be performed.
2. Apply data pre-processing techniques such as standardization/normalization, transformation, aggregation, discretization/binarization, sampling etc. on any dataset
3. Run Apriori algorithm to find frequent itemsets and association rules on 2 real datasets and use appropriate evaluation measures to compute correctness of obtained patterns a) Use minimum support as 50% and minimum confidence as 75% b) Use minimum support as 60% and minimum confidence as 60 % I.
4. Use Naive bayes, K-nearest, and Decision tree classification algorithms and build classifiers on any two datasets. Divide the data set into training and test set. Compare the accuracy of the different classifiers under the following situations: a) Training set = 75% Test set = 25% b) Training set = 66.6% (2/3rd of total), Test set = 33.3% II. Training set is chosen by i) hold out method ii) Random subsampling iii) Cross-Validation. Compare the accuracy of the classifiers obtained. Data is scaled to standard format.
5. Use Simple K-means algorithm for clustering on any dataset. Compare the performance of clusters by changing the parameters involved in the algorithm. Plot MSE computed after each iteration using a line plot for any set of parameters.

Project: Students should be promoted to take up one project on any UCI/kaggle/data.gov.in or a dataset verified by the teacher. Preprocessing steps and at least one data mining technique should be shown on the selected dataset. This will allow the students to have a practical knowledge of how to apply the various skills learnt in the subject for a single problem/project.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSE-A4/DSE: DATA MINING-II

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Data Mining - II	**4**	**3**	**0**	**1**	Pass in Class XII	Data Mining-I

Course Objectives

The course introduces the students to the supervised and unsupervised learning techniques. Students will learn about the importance of ensemble methods, cluster analysis, anomaly detection and their applicability in mining patterns in real applications. At the end students will be exposed to two advanced topics: text mining and time-series mining. Students will use the learned topics in solving real applications using open-source software.

Learning outcomes

On successful completion of the course, students will be able to:

- Differentiate between partition-based, density-based and hierarchical clustering
- Build ensemble models to improve predictive performance of the classifier
- Identify anomalies and outliers using supervised and unsupervised techniques
- Analyze time-series data and extract patterns from the stamped data
- Mine textual data and do topic modelling

Syllabus

Unit 1 **(9 hours)**

Clustering:

Partitioning Methods, Hierarchical Methods, Density-Based Methods, Comparison of different methods

Unit 2 **(8 hours)**

Ensemble Methods:

Need of ensemble, Random Forests, Bagging and Boosting

Unit 3 **(10 hours)**

Anomaly Detection:

Outliers and Outlier Analysis, Outlier Detection Methods, Statistical Approaches, Proximity-based and density-based outlier detection, Clustering-based approaches

Unit 4 **(8 hours)**

Mining Text Data:

Document Preparation and Similarity, Clustering Methods for Text, Topic Modeling

Unit 5 **(10 hours)**

Stream Mining:

Time series basics, Date Ranges, Frequencies, and Shifting, Resampling and moving windows functions, Decay function, Clustering stamped data: STREAM and CluStream

Essential/recommended readings

1. Tan P.N., Steinbach M, Karpatne A. and Kumar V. Introduction to Data Mining, 2nd edition, Pearson, 2019.

2. Zaki M. J. and Meira J. Jr. Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd edition, Cambridge University Press, 2020.

3. Aggarwal C. C. Data Mining: The Textbook, Springer, 2015.

Additional References

1. Han J. Kamber M. and Pei J. Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2011.

2. Dunham M. Data Mining: Introductory and Advanced Topics, Pearson, 2006

Suggested Practicals List (If any): (30 Hours)

Practical exercise such as

1. Apply Partitioning Methods, Hierarchical Methods, Density-Based Methods for clustering on a data set and compare the performance of the obtained results using different metrics
2. Create an ensemble using Random Forest and show the impact of bagging and boosting on the performance
3. Apply different outlier-detection methods on a noisy dataset and compare their effectiveness in terms of outliers reported
4. Compute similarity between two documents after required document preparation
5. Considering a time-stamped data (sales data/weather data), compare the aggregate values visually using different moving windows function
6. Write a program to find the latent topics in a document using any topic modeling method and display top 5 terms that contribute to each topic along with their strength. Also, visualize the distribution of terms contributing to the topics.

Project: Students should be promoted to take up one project covering at least one unit of the syllabus on any UCI/kaggle/data.gov.in or a dataset verified by the teacher. This will allow the students to have a practical knowledge of how to apply the various skills learnt in the subject for a single problem/project.

GE4b/DSE: INTRODUCTION TO WEB PROGRAMMING

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course	Eligibility criteria	Pre-requisit e of the

		Lecture	Tutorial	Practical/ Practice		**course** (if any)
Introduction to web programming	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Course Objectives

The course aims at introducing the basic concepts and techniques of client side web programming. The student shall be able to develop simple websites using HTML, CSS and Javascript.

Learning outcomes

On successful completion of the course, students will be able to :

● Build websites using the elements of HTML.

● Build dynamic websites using the client side programming techniques with CSS, Javascript and jQuery.

● Learn to validate client-side data

Syllabus

Unit 1 **(5 hours)**

Introduction:

Introduction to Internet and web design. Basic concepts of web architecture.

Unit 2 **(12 hours)**

HTML:

Introduction to hypertext mark-up language (html), creating web pages, lists, hyperlinks, tables, web forms, inserting images, frames.

Unit 3 **(8 hours)**

Cascading style sheet (CSS):

Concept of CSS, creating style sheet, Importing style sheets, CSS properties, CSS styling (background, text format, controlling fonts), CSS rules, Style Types, CSS Selectors, CSS cascade, working with block elements and objects, working with lists and tables, CSS id and class, box model (introduction, border properties, padding properties, margin properties).

Unit 4 **(10 hours)**

Javascript:

Document object model, data types and variables, functions, methods and events, controlling program flow, JavaScript object model, built-in objects and operators, validations.

Unit 5 **(10 hours)**

jQuery and JSON:

Introduction to jQuery, syntax, selectors, events. JSON file format for storing and transporting data.

References:

1. Nixon, R. Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5, O'Rielly, 2018.

2. Powell, T.A. HTML & CSS: The Complete Reference, 5th edition, Tata McGrawHill, 2010.

3. Duckett, J. JavaScript and JQuery: Interactive Front-End Web Development, Wiley, 2014.

Additional References:

1. Minnick, J. Web Design with HTML5 and CSS3, 8th edition, Cengage Learning, 2015.

2. Boehm, A., & Ruvalcaba, Z. Munarch's HTML5 and CCS, 4th edition, Mike Murach & Associates, 2018.

3. J. A. Ramalho Learn Advanced HTML 4.0 with DHTML, BPB Publications, 2007.

4. Ivan Bayross Web Enabled Commercial Application Development Using Html, Dhtml, Javascript, Perl CGI, BPB Publications, 2009.

Suggested Practical List (If any): (30 Hours)

Practical exercises such as

HTML

1. Create an HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text.
2. Create an HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking
3. Create an HTML displaying this semester's time table.
4. Create a website with horizontal and vertical frames. Top horizontal frame showing your college's name and logo. Bottom horizontal frame split into two vertical frames. The left frame with hyperlinks to pages related to faculty, courses, student activities, etc. The right frame showing corresponding pages based on the link clicked on the left frame.
5. Create a student registration form using HTML which has the following controls:

   Text Box
   Dropdown box
   Option/radio buttons

   Check boxes Reset and Submit button

CSS

Create a webpage for your department with drop down navigation menu for faculty, courses, activities, etc.. Implement the webpage using styles, rules, selectors, ID, class.

Javascript

1. Create event driven programs for the following:

   a. Enter a number and on click of a button print its multiplication table.
   b. Print the largest of three numbers entered by the user.
   c. Find the factorial of a number entered by the user.
   d. Enter a list of positive numbers using the prompt terminated by a zero. Find the sum and average of these numbers.

2. Create a student registration form using text, radio button, check box, drop down box, text field and all other required HTML elements. Customise the CSS and javascript to input and validate all data. Create functions to perform validation of each element, example:

   a. Roll number is a 7-digit numeric value
   b. Name should be an alphabetical value(String)

   c. Non-empty and valid fields like DOB

jQuery and JSON

1. Change text color and contents using button click events using jQuery
2. Select elements using ID, class, elements name, attribute name
3. Run code on click events in jQuery
4. Handle HTML form, store the data in JSON object, pass them to another page and display it there using jQuery/Javascript

GE6d/DSE: DATA PRIVACY

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Data Privacy	4	3	0	1	Pass in Class XII	NIL

Objective:

This course aims to provides students with the ability to identify privacy related aspects of data uses, attacks on data privacy, evaluate proposed technical mechanisms for privacy protection and understand ethical issues related to data privacy

Course Learning Outcomes:

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

- Understand the basic principles of data privacy and the implications of data breaches.
- Identify and evaluate different methods of protecting sensitive data.
- Explain the role of privacy regulations in safeguarding personal information.
- Implement basic cryptographic techniques to secure data.
- Apply data anonymization techniques to protect personal information.
- Analyze the ethical considerations in data privacy.

Syllabus

Unit 1: Introduction to Data Privacy and Privacy Regulations

Definition of data privacy, Historical context of data privacy, Types of sensitive data, Privacy laws and regulations

Unit 2: Data Privacy Attacks, Cryptography and Data Protection

Type of Attacks/ Data Breaches on Data Privacy, Impact of Data Breaches / Attacks, Introduction to cryptography, Symmetric and asymmetric encryption, Hashing and digital signatures

Unit 3: Data Collection, Use and Reuse

Harms Associated with Data collections, use and reuse, Introduction to data anonymization, Data Anonymization Techniques for anonymizing data, Challenges in anonymizing data

Unit 4: Ethical Considerations in Data Privacy

Privacy and Surveillance, Ethics of Data Collection and Use, Bias and discrimination in data analysis

References:

1. Ronald Leenes, Rosamunde van Brakel, and Serge Gutwirth: Data Protection and Privacy: The Age of Intelligent Machines, Hart Publishing, 2017.

2. Stallings, William. Information privacy engineering and privacy by design: Understanding privacy threats, technology, and regulations based on standards and best practices. Addison-Wesley Professional, 2019.

3. Venkataramanan, Nataraj, and Ashwin Shriram. Data privacy: principles and practice. Chapman and Hall/CRC, 2016.

4. Jarmul, Katharine. Practical Data Privacy. " O'Reilly Media, Inc.", 2023.

Additional References:

1. THE DIGITAL PERSONAL DATA PROTECTION ACT, 2023 (https://www.meity.gov.in/writereaddata/files/Digital%20Personal%20Data%20Protection%20Act%202023.pdf)

2. Ravinder Kumar Gaurav Goyal, The Right to Privacy in India: Concept and Evolution, Publisher: Lightning Source, 2016.

3. https://onlinecourses.nptel.ac.in/noc22_cs37/preview

4. https://www.coursera.org/learn/northeastern-data-privacy/home/info

5. Naavi: Personal Data Protection Act of India (PDPA 2020): Be Aware, Be Ready and Be Compliant, Notion Press, 2020. **Additional References:**

Suggested Practicals:

Students may be asked to perform some of the following practical activities related to data privacy:

1. Using any high level language, students, may write a program for encryption/decryption using substitution (Julius Ceaser) Cipher.

2. Using any high level language, students may write a program for transpositional cipher.

3. Students may learn about the role/use of hashing, and digital signatures, and implement them in practice.

4. Data Privacy Audit: Students can conduct a data privacy audit of a company or organization to identify potential vulnerabilities and risks in their data privacy practices.

5. Privacy Impact Assessment: Students can conduct a privacy impact assessment (PIA) of a new technology or system to identify potential privacy risks and develop strategies to mitigate them.

6. Regulation Compliance: Students can explore the requirements of the Data Protection Regulations and develop a plan for ensuring compliance with the regulation.

7. Anonymization Techniques: Students can learn about data anonymization techniques, such as k-anonymity, differential privacy, and data masking, and apply them to a real-world dataset.

8. Privacy Policy Analysis: Students can analyze the privacy policies of different companies and identify gaps or areas for improvement.

9. Ethical Considerations: Students can explore ethical considerations in data privacy, such as the balance between privacy and security, the impact of data collection and analysis on marginalized communities, and the role of data ethics in technology development.

10. Case Studies: Students can analyze case studies of privacy breaches or successful privacy protection strategies, and identify key lessons and takeaways.

DSC17/DSC-A5/GE7c: MACHINE LEARNING

Credit distribution, Eligibility and Prerequisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Machine Learning	4	3	0	1	Pass in Class XII	Programming using Python/ Object Oriented Programming using Python

Course Objectives

The course aims at introducing the basic concepts and techniques of machine learning so that a student can apply machine learning techniques to a problem at hand.

Learning outcomes

On successful completion of the course, students will be able to:

- Differentiate between supervised and unsupervised learning tasks.
- State the need of preprocessing, feature scaling and feature selection.
- Formulate classification, regression and clustering problems as optimization problems
- Implement various machine learning algorithms learnt in the course.

SYLLABUS

Unit 1 **(5 Hours)**

Introduction:

Basic definitions and concepts, key elements, supervised and unsupervised learning, applications of ML.

Unit 2 **(8 Hours)**

Preprocessing:

Feature scaling, feature selection methods. dimensionality reduction (Principal Component Analysis), class balancing, outlier detection and removal.

Unit 3 **(12 Hours)**

Regression:

Linear regression with one variable, linear regression with multiple variables, gradient descent, over-fitting, regularization. Regression evaluation metrics.

Unit 4 **(12 Hours)**

Classification: Decision trees, Naive Bayes classifier, logistic regression, k-nearest neighbor classifier, perceptron, multilayer perceptron, neural networks, back-propagation algorithm, Support Vector Machine (SVM). Classification evaluation metrics

Unit 5 **(8 Hours)**

Clustering: Approaches for clustering, distance metrics, K-means clustering, hierarchical clustering.

Essential/recommended readings

1. Mitchell, T.M. Machine Learning, McGraw Hill Education, 2017.

2. James, G., Witten. D., Hastie. T., Tibshirani., R. An Introduction to Statistical Learning with Applications in R, Springer, 2014.

3. Alpaydin, E. Introduction to Machine Learning, MIT press, 2009.

Additional References

1. Flach, P., Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2015.

2. Christopher & Bishop, M., Pattern Recognition and Machine Learning, New York: Springer-Verlag, 2016.

3. Sebastian Raschka, Python Machine Learning, Packt Publishing Ltd, 2019

Suggested Practical List:

Practical exercises such as

Use Python for practical labs for Machine Learning. Utilize publicly available datasets from repositories like https://data.gov.in/ and https://archive.ics.uci.edu/ml/datasets.php

For evaluation of the regression/classification models, perform experiments as follows:

• Scale/Normalize the data • Reduce dimension of the data with different feature selection techniques • Split datasets into training and test sets and evaluate the decision models • Perform k-cross-validation on datasets for evaluation

Report the efficacy of the machine learning models as follows: • MSE and R2 score for regression models • Accuracy, TP, TN, FP, TN, error, Recall, Specificity, F1-score, AUC for classification models

For relevant datasets make prediction models for the following

1. Naïve Bayes Classifier

2. Simple Linear Regression multiple linear regression

3. Polynomial Regression

4. Lasso and Ridge Regression

5. Logistic regression

6. Artificial Neural Network

7. k-NN classifier

8. Decision tree classification

9. SVM classification

10. K-Means Clustering

11. Hierarchical Clustering

DSC16/GE6e/DSE: ARTIFICIAL INTELLIGENCE

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Artificial Intelligence	4	3	0	1	Pass in Class XII	Programming using C++/Programming using Python/Object Oriented

						Programming using Python

Course Objectives

The objectives of this course are to:

- To introduce basic concepts and techniques of Artificial Intelligence (AI).
- To apply informed search techniques for different applications.
- To learn various knowledge representation techniques and writing Prolog programs.
- To learn about the latest techniques for developing AI systems.

Learning outcomes

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

- identify problems that are amenable to solutions by specific AI methods.
- state the utility of different types of AI agents.
- apply different informed search techniques for solving problems.
- use knowledge representation techniques for AI systems.

SYLLABUS

Unit 1 **6 Hours**

Introduction: Introduction to artificial intelligence, background and applications, Turing test, Weak AI, Strong AI, Narrow AI, Artificial General Intelligence, Super AI, rational agent approaches to AI, introduction to intelligent agents, their structure, behavior and task environment.

Unit 2 **12 Hours**

Problem Solving and Searching Techniques: Problem characteristics, production systems, control strategies, breadth-first search, depth-first search, hill climbing and its variations, heuristics search techniques: best-first search, A* algorithm, constraint satisfaction problem, means-end analysis, introduction to game playing, min-max and alpha-beta pruning algorithms.

Unit 3 **16 Hours**

Knowledge Representation: Propositional logic, First-Order Predicate logic, resolution principle, unification, semantic nets, conceptual dependencies, frames, and scripts, production rules, Introduction to Programming in Logic (PROLOG).

Unit 4 **8 Hours**

Understanding Natural Languages: Components and steps of communication, the contrast between formal and natural languages in the context of grammar, Chomsky hierarchy of grammars, parsing, and semantics, Parsing Techniques, Context-Free and Transformational Grammars, Recursive transition nets.

Unit 5 **3 Hours**

AI The Present and the Future: Symbolic AI, Data-driven AI and Machine Learning, Introduction to Machine Learning and Deep Learning based AI, Interpretable and Explainable AI, Ethics of AI: benefits and risks of AI.

Essential/recommended readings

1. Russell, Stuart, J. and Norvig, Peter, *Artificial Intelligence - A Modern Approach*, Pearson, 4th edition, 2020..
2. Bratko, Ivan, *Prolog Programming for Artificial Intelligence*, Addison-Wesley, Pearson Education, 4th edition, 2012.
3. Patterson, DAN,W, *Introduction to A.I. and Expert Systems* – PHI, 2007.
4. Clocksin, W., F. and Mellish, *Programming in PROLOG*, 5th edition, Springer, 2003.

Additional references

1. Kaushik, Saroj, *Artificial Intelligence*, Cengage Learning India, 2011.
2. Rich, Elaine and Knight, Kelvin, *Artificial Intelligence*, 3rd edition, Tata McGraw Hill, 2010

Practical List :

Practical exercises such as

1. Write a program in Prolog to implement TowerOfHanoi(N) where N represents the number of disks.
2. Write a program to implement the Hill climbing search algorithm in Prolog.
3. Write a program to implement the Best first search algorithm in Prolog.
4. Write a program to implement A* search algorithm in Prolog.
5. Write a program to implement the min-max search algorithm in Prolog.

6. Write a program to solve the Water-Jug Problem in Prolog.
7. Implement sudoku problem (minimum 9×9 size) using constraint satisfaction in Prolog.
8. Write a Prolog program to implement the family tree and demonstrate the family relationship.
9. Write a Prolog program to implement knowledge representation using frames with appropriate examples.
10. Write a Prolog program to implement conc(L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3.
11. Write a Prolog program to implement reverse(L, R) where List L is original and List R is reversed list.
12. Write a Prolog program to generate a parse tree of a given sentence in English language assuming the grammar required for parsing.
13. Write a Prolog program to recognize context free grammar $a^n b^n$.

DSC-A6/DSE: DEEP LEARNING

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Deep Learning	**4**	**3**	**0**	**1**	Pass in Class XII	Programming using Python/Object Oriented Programming using Python/Mathematics for Computing

Course Objectives

The objective of this course is to introduce students to deep learning algorithms and their applications in order to solve real problems.

Learning outcomes

On successful completion of this course, the student will be able to:

- Describe the feed-forward and deep networks.
- Design single and multi-layer feed-forward deep networks and tune various hyper parameters.
- Implement deep neural networks to solve a problem
- Analyze performance of deep networks.
- Use pre-trained models to solve a problem.

SYLLABUS

Unit 1 **(8 Hours)**

Introduction to neural networks:

Artificial neurons, perceptron, computational models of neurons, Structure of neural networks, Multilayer feedforward neural networks (MLFFNN), Backpropagation learning, Empirical risk minimization, bias-variance tradeoff, Regularization, output units: linear, softmax , hidden units:tanh, RELU

Unit 2 **(8 Hours)**

Deep neural networks:

Difficulty of training DNNs, Greedy layerwise training, Optimization for training DNN's, Newer optimization methods for neural networks(AdaGrad, RMSProp, Adam), Regularization methods(dropout, drop connect, batch normalization).

Unit 3 **(8 Hours)**

Convolution neural networks(CNNs):

Introduction to CNN - convolution, pooling, Deep CNNs - LeNet, AlexNet. Training CNNs, weights initialization, batch normalization, hyperparameter optimization, Using a pre trained convnet

Unit 4 **(8 Hours)**

Recurrent neural networks (RNNs):

Sequence modeling using RNNs, Backpropagation through time, LongShort Term Memory (LSTM), Bidirectional RNN

Unit 5 **(8 Hours)**

Unsupervised deep learning:

Autoencoders, Generative Adversarial Networks.

Unit 6 **(5 Hours)**

Applications:

Computer vision, Speech recognition and NLP.

Essential/recommended readings

1. Ian Goodfellow, Yodhua Bengio and Aaron Courville, Deep Learning, MIT Press Book, 2016.

2. Francois Chollet, Deep Learning with python, 2nd edition, Meaning Publications Co, 2021.

Additional References

1. Bunduma, N., Fundamentals of Deep Learning, 1st edition, O'reilly Books, 2017.

2. Heaton, J., Deep Learning and Neural Networks, 1st edition, Heaton Research Inc., 2015.

Suggested Practical List :

Practical exercises such as

The following practicals are to be conducted using Python.

1. Implement a feed-forward neural networks for classifying movie reviews as positive or negative(using IMDB dataset)
2. Implement a deep-neural feed-forward network for estimating the price of house, given real-estate data(Boston Housing Price)
3. Implement a deep-neural network for classifying news wires by topic (Reuters dataset).
4. Implement CNN for classifying MNIST dataset

5. Create a model for time-series forecasting using RNN/LSTM 6. Implement an auto-encoder

DSE: NUMERICAL OPTIMIZATION

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Numerical Optimization	4	3	0	1	Passed 12th class with Mathematics	Programming using Python/ Programming using C++

Course Objectives

The course aims to provide students with the experience of mathematically formulating a large variety of optimization/decision problems emerging out of various fields like data science, machine learning, business, and finance. The course focuses on learning techniques to optimize problems in order to obtain the best possible solution.

Learning outcomes

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

- Mathematically formulate the optimization problems using the required number of independent variables.
- Define constraint functions on a problem.
- Check the feasibility and optimality of a solution.
- Apply conjugate gradient method to solve the problem.

SYLLABUS

Unit 1 **(6 hours)**

Introduction: Mathematical Formulation using example, Continuous versus Discrete Optimization, Constrained and Unconstrained Optimization, Global and Local Optimization, Stochastic and Deterministic Optimization, Convexity, Optimization Algorithms

Unit 2 **(14 hours)**

Fundamentals of Unconstrained Optimization: Concept of a Solution - Recognizing a Local Minimum, Nonsmooth Problems, Overview of Algorithms - Two Strategies: Line Search and Trust Region, Search Directions for Line Search Methods, Models for Trust-Region Methods, Scaling. Line Search - Convergence of Line Search Methods, Rate of Convergence - Convergence Rate of Steepest Descent; Newton's Method, Quasi-Newton Methods. Trust Region - The Cauchy Point Algorithm; Global Convergence - Reduction Obtained by the Cauchy Point; Convergence to Stationary Points.

Unit 3 **(7 hours)**

Conjugate Gradient Methods: Basic Properties of the Conjugate Gradient Method, A Practical Form of the Conjugate Gradient Method, and Rate of Convergence

Unit 4 **(8 hours)**

Calculating Derivatives: Finite-Difference Derivative Approximations, Approximating the Gradient, Approximating a Sparse Jacobian, Approximating the Hessian, Approximating a Sparse Hessian

Unit 5 **(10 hours)**

Theory of Constrained Optimization: Local and Global Solutions, Smoothness, Examples - A Single Equality Constraint, A Single Inequality Constraint, Two Inequality Constraints, Tangent Cone and Constraint Qualifications, First-Order Optimality Condition, Second-Order Conditions - Second-Order Conditions and Projected Hessians. Linear and non-linear constrained optimization, augmented Lagrangian Method

Essential/recommended readings

1. J. Nocedal and S.J. Wright, *Numerical Optimization*, 2nd edition, Springer Series in Operations Research, 2006.
2. A, Mehra, S Chandra, Jayadeva, *Numerical Optimization with Applications*, Narosa Publishing House, New Delhi, 2009,

Additional References

1. R. W. Hamming, *Numerical Methods for Scientists and Engineers,* 2nd edition, Dover Publications, 1986.
2. Q. Kong, T. Siauw, A. Bayen, *Python Programming and Numerical Methods: A Guide for Engineers and Scientists,* 1st edition, 2020.

Suggested Practical List (If any)

:(30 Hours)

Practical exercises such as

Write programs to implement the following methods:

Constrained and Unconstrained Optimization, Global and Local Optimization, Line Search and Trust Region, Convergence of Line Search Methods, Rate of Convergence - Convergence Rate of Steepest Descent, Newton's Method, Quasi-Newton Methods, The Cauchy Point algorithm, Finite-Difference Derivative Approximations, Convergence to Stationary Points, Conjugate Gradient Method, Rate of Convergence, Approximating a Sparse Jacobian, Approximating the Hessian, Approximating a Sparse Hessian, First-Order Optimality Condition, Second-Order Conditions - Second-Order Conditions, and Projected Hessians. Linear and non-linear constrained optimization Augmented Lagrangian Methods.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

GE7e/DSE: ETHICAL HACKING

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Ethical Hacking	4	3	0	1	Pass in Class XII	NIL

Course Objectives

The objective of this course is to enable students to be part of such a team that can conduct the security assessment of an organization through the process of ethical hacking. This course will introduce the students, the idea of security assessment of systems and networks under investigation and how to perform them under the legal and ethical framework. Further, this course will outline the importance of various stages of ethical hacking, including but not limited to tasks such as penetration testing, and usage of various tools at each stage.

Learning outcomes

On successful completion of the course, students will be able to:

1. Understand and acknowledge the relevance of legal, ethical, and professional challenges faced by an ethical hacker.
2. Apply fundamental principles of system, application, and network security to ethically attack / penetrate the system to uncover the security flaws.
3. Perform evaluation of security systems through a systematic ethical hacking process and recommend countermeasures to improve security.
4. Understand and use various tools and techniques used in various stages of the ethical hacking process.

Syllabus

Unit 1 **(4 Hours)**

Introduction: Overview of information security threats and attack vectors, vulnerability assessment and penetration testing concepts, information security controls, security laws and standards. OWASP top 10 vulnerabilities

Unit 2 **(6 hours)**

Footprinting and Reconnaissance: Introduction to network reconnaissance tools such ipconfig, ifconfig, domain tools, nmap, Wireshark, etc.

Unit 3 **(8 hours)**

Scanning and Enumeration: Network penetration testing, Password cracking techniques and countermeasures, NetBIOS tools

Unit 4 **(8 hours)**

Gaining and Maintaining Access: Network level attacks and countermeasures, Metasploit framework, Burp Suite

Unit 5 **(8 hours)**

Exploitation and Covering Tracks: Privilege escalation, social Engineering, identity theft, countermeasures, Covering tracks using attrib command and creating Alternate Data Stream (ADS) in Windows, Erasing evidence from Windows logs, Strategies for maintaining access.

Unit 6 **(8 hours)**

Advanced stages: Denial of service, Session hijacking, hacking web servers, hacking web applications, sql injection etc.

Unit 7 **(8 hours)**

NIST Cybersecurity framework and ISO standards: NIST cybersecurity framework, Cyber Kill chain, ISO/IEC 27001 and related standards.

Unit 8 **(4 Hours)**

Cyber Defense and Reporting: Preparing vulnerability assessment reports, presenting post testing findings, preparing recommendations

References

1. Patrick Engbretson, The Basics of Hacking and Penetration Testing, 2nd Edition, Syngress, 2013.
2. Georgia Weidman, Penetration TEsting: A Hands-On Introduction to Hacking, 1st Edition, No Starch Press, 2014.

Additional References

1. Peter Kim, The Hacker Playbook 3: Practical Guide to Penetration Testing, Zaccheus Entertainment, 2018.
2. Jon Erickson, Hacking: The Art of Exploitation, No Starch Press, 2008.
3. Online Resources:

   https://www.sans.org/cyberaces/

   https://skillsforall.com/

   https://www.hackingloops.com/ethical-hacking/

Suggested Practical List (If any): (30 Hours)

Perform the following activities, record and report in standard form.

(NOTE: Exercise extra caution while performing these exercises and codes)

1. Perform various Virtual Machine based exercises on https://vulnhub.com/
2. Perform Capture the Flag (CTF) exercises from https://www.hacker101.com/
3. Follow the lessons and activities from https://www.hackingloops.com/ethical-hacking/
4. Google site for hacking https://google-gruyere.appspot.com/
5. OWASP WebGoat https://github.com/WebGoat/WebGoat

GE8d/DSE: CYBER FORENSICS

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Cyber Forensics	4	3	0	1	Pass in Class XII	NIL

Course Objective:

This course is to equip students with the knowledge and skills necessary to identify, collect, analyze and present digital evidence in a manner that is admissible in legal proceedings. Students will be able to conduct a thorough investigation of cybercrime incidents, preserve digital evidence, and report findings to relevant stakeholders.

Course Learning Outcomes:

- Students will be able to demonstrate an understanding of the principles of digital forensics, including legal considerations, recognition, collection, and preservation of digital evidence.
- Students will develop skills in using digital forensics tools and techniques, such as creating disk images, conducting keyword and grep searches, and examining Windows registry.
- Students will learn evidence recovery methods, including deleted file recovery, formatted partition recovery, and data recovery procedures, and ethical considerations.
- Students will gain knowledge of cyber forensic investigation tools and techniques, including digital evidence collection, preservation, and password cracking.
- Students will understand cyber laws and crimes, including hacking, viruses, intellectual property, and e-commerce, and the legal system of information technology, including jurisdiction issues and security and evidence in e-commerce.

Unit 1 – Digital Forensics: Introduction to digital forensics, legal considerations, recognising and collecting digital evidence, preservation of evidence, hash values and file hashing, creating disk images, keyword and grep searches, network basics, reporting and peer review, digital forensics report.

Unit 2 – Windows OS Forensics: Bits, bytes, Endieness, Disk partition schema, File systems – FAT, NTFS, ex-FAT, windows registry forensics, examining windows registry, NTUser.Dat Hive File Analysis, SAM Hive file, Software Hive file, System Hive File, USRClass.dat Hive File, AmCache Hive File.

Unit 3 – Evidence Recovery: Introduction to Deleted File Recovery, Formatted Partition Recovery, Data Recovery Tools, Data Recovery Procedures and Ethics, Complete time line analysis of computer files based on file creation, File modification and file access, Recover Internet Usage Data, Recover Swap Files/Temporary Files/Cache Files, Introduction to Encase Forensic Edition, Forensic Tool Kit (FTK), Use computer forensics software tools to cross validate findings in computer evidence.

Unit 4 – Investigation: Introduction to Cyber Forensic Investigation, Investigation Tools, Digital Evidence Collection, Evidence Preservation, E-Mail Investigation, E-Mail Tracking, IP Tracking, E-Mail Recovery, Encryption and Decryption methods, Search and Seizure of Computers, Recovering deleted evidences, Password Cracking.

Unit 5 – Cyber Crimes and Cyber Laws: Introduction to IT laws & Cyber Crimes, Internet, Hacking, Cracking, Viruses, Software Piracy, Intellectual property, Legal System of Information Technology, Understanding Cyber Crimes in context of Internet, Indian Penal Law & Cyber Crimes Fraud Hacking Mischief, International law, E-Commerce-Salient Features On-Line contracts Mail Box rule Privities of, Contracts Jurisdiction issues in E-Commerce Electronic Data Interchange, Security and Evidence in E-Commerce Dual Key encryption Digital signatures security issues.

References:

1. Marjee T. Britz, Computer Forensics and Cyber Crime: An Introduction, Pearson Education, 2013.

2. C. Altheide& H. Carvey Digital Forensics with Open Source Tools, Syngress, 2011. ISBN: 9781597495868.

Additional References:

1. Computer Forensics: Investigating Network Intrusions and Cybercrime" by Cameron H. Malin, Eoghan Casey, and James M. Aquilina

2. Online Course management System: https://esu.desire2learn.com/

3. Computer Forensics, Computer Crime Investigation by John R,Vacca, Firewall Media, New Delhi.

4. Computer Forensics and Investigations by Nelson, Phillips Enfinger, Steuart, CENGAGE Learning

5. Real Digital Forensics by Keith j.Jones, Richard Bejitlich,Curtis W.Rose ,AddisonWesley Pearson Education

Suggested Practicals

It is suggested that the following tools/e-resources can be explored during the practical sessions • Wireshark • COFEE Tool • Magnet RAM Capture • RAM Capture • NFI Defragger • Toolsley • Volatility

1. Study of Network Related Commands (Windows)
2. Study of Network related Commands(Linux)
3. Analysis of windows registry
4. Capture and analyze network packets using Wireshark. Analyze the packets captured.
5. Creating a Forensic image using FTK Imager/ Encase Imager: creating forensic image, check integrity of data, analyze forensic image
6. Using System internal tools for network tracking and process monitoring do the following:
    a. Monitor live processes
    b. Capture RAM
    c. Capture TCP/UDP packets
    d. Monitor Hard disk
    e. Monitor Virtual Memory
    f. Monitor Cache Memory

DSC20/DSC08/GE8a: INFORMATION SECURITY

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Information Security	4	3	0	1	Pass in Class XII	NIL

Course Objective

The goal of this course is to make a student learn basic principles of information security. Over the due course of time, the student will be familiarized with cryptography, authentication and access control methods along with software security. Potential security threats and vulnerabilities of systems are also discussed along with their impacts and countermeasures. This course also touches upon the implications of security in cloud and Internet of Things (IoT).

Learning Outcomes

On successful completion of this course, a student will be able to

- Identify the major types of threats to information security.
- Describe the role of cryptography in security.
- Discover the strengths and weaknesses of private and public key cryptosystems.
- Identify and apply various access control and authentication mechanisms.
- Discuss data and software security and related issues.
- Explain network security threats and attacks.
- Articulate the need for security in cloud and IoT.

Syllabus

Unit 1 **(3 hours)**

Overview: Computer Security Concepts, Threats, Attacks, and Assets, Security Functional Requirements, Fundamental Security Design Principles.

Unit 2 **(6 hours)**

Cryptographic tools: Confidentiality with Symmetric Encryption, Message Authentication and Hash Functions, Public-Key Encryption, Digital Signatures and Key Management, Random and Pseudorandom Numbers, Practical Application: Encryption of Stored Data.

Unit 3 **(10 hours)**

User authentication and Access Control: Digital User Authentication Principle, Password-Based Authentication, Remote User Authentication, Security Issues for User Authentication Access Control Principles, Subjects, Objects, and Access Rights, Discretionary Access Control, Example: UNIX File Access Control, Role-Based Access Control, Attribute-Based Access Control, Identity, Credential, and Access Management, Trust Frameworks.

Unit 4 **(5 hours)**

Database and Data Center Security:

The Need for Database Security, SQL Injection Attacks, Database Access Control.

Unit 5 **(8 hours)**

Software Security: Types of Malicious Software, Advanced Persistent Threat, Propagation — Infected Content - Viruses, Propagation — Vulnerability Exploit - Worms, Propagation — Social Engineering — SPAM E-Mail, Trojans, Payload — System Corruption, Payload — Attack Agent — Zombie, Bots, Payload — Information Theft — Keyloggers, Phishing, Spyware, Payload — Stealthing — Backdoors, Rootkits, Countermeasures. **Overflow Attacks -** Stack Overflows, Defending Against Buffer Overflows, Other Forms of Overflow Attacks. **Software Security Issues -** Handling Program Input, Writing Safe Program Code, Handling Program Input.

Unit 6 **(6 hours)**

Network Security: Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of-Service Attacks, Overview of Intrusion Detection, Honeypots, The Need for Firewalls, Firewall Characteristics and Access Policy, Types of Firewalls, Public-Key Infrastructure.

Unit 7 **(7 hours)**

Wireless, Cloud and IoT Security: Cloud Computing, Cloud Security Concepts, Cloud Security Approaches, The Internet of Things, IoT Security. Wireless Security Overview, Mobile Device Security.

References

1. W. Stallings, L. Brown, *Computer Security: Principles and Practice*, 4th edition, Pearson Education, 2018.

Additional References

1. Pfleeger C.P., Pfleeger S.L., Margulies J. *Security in Computing*, 5th edition, Prentice Hall, 2015.
2. Lin S., Costello D.J., *Error Control Coding: Fundamentals and applications,* 2nd edition, Pearson Education, 2004.
3. Stallings W. *Cryptography and network security*, 7th edition, Pearson Education, 2018.
4. Berlekamp E. *Algebraic Coding Theory*, World Scientific Publishing Co., 2015.
5. Stallings W. *Network security essentials Applications and Standards*, 6th edition, Pearson Education, 2018.
6. Whitman M.E., Mattord H.J., *Principle of Information Security*, 6th edition, Cengage Learning, 2017.
7. Bishop M., *Computer Security: Art and Science*, 2nd Revised edition, Pearson Education, 2019.
8. Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2nd edition, John Wiley & Sons, 2008.

Suggested Practical List

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois.
2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Use nmap/zenmap to analyze a remote machine.
4. Use Burp proxy to capture and modify the message.
5. Implement caesar cipher substitution operation.
6. Implement monoalphabetic and polyalphabetic cipher substitution operation.
7. Implement playfair cipher substitution operation.
8. Implement hill cipher substitution operation.
9. Implement rail fence cipher transposition operation.
10. Implement row transposition cipher transposition operation.
11. Implement product cipher transposition operation.

GE8c/DSE: INTRODUCTION TO PARALLEL PROGRAMMING

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Introduction to Parallel Programming	4	3	0	1	Pass in Class XII	Computer System Architecture/A course in C++at class XII/Data Structures, Operating Systems

Course Objective

The course introduces the students to the basic concepts and techniques of parallel programming. It enables them to design and implement parallel algorithms. The course would give the students hands-on practice to write parallel programs using shared and distributed memory models using OpenMP and Message Passing Interface (MPI).

Course Learning Outcomes

On successful completion of this course, the student will be able to:

1. Appreciate the need of Parallel algorithms
2. Describe architectures for parallel and distributed systems.
3. Develop elementary parallel algorithms in shared memory models.
4. Develop elementary parallel algorithms in distributed memory models.

Syllabus

Unit 1
Introduction to Parallel Computing: Trends in microprocessor architectures, memory system performance, dichotomy of parallel computing platforms, physical organization of parallel platforms, communication costs in parallel machines, SIMD versus MIMD architectures, shared versus distributed memory, PRAM shared-memory model, distributed-memory model.

Unit 2
OpenMP programming for shared memory systems: Thread Basics, Controlling Thread and Synchronization Attributes, Multi-thread and multi-tasking, Context Switching, Basic OpenMP thread functions, Shared Memory Consistency Models and the Sequential Consistency Model,

Race Conditions, Scoping variables, work-sharing constructs, critical sections, atomic operations, locks, OpenMP tasks, Introduction to tasks, Task queues and task execution, Accessing variables in tasks, Completion of tasks and scoping variables in tasks.

Unit 3

MPI programming for distributed memory systems: MPI basic communication routines (Introduction to MPI and basic calls, MPI calls to send and receive data, MPI call for broadcasting data, MPI Non-blocking calls, Introduction to MPI Collectives, Types of interconnects (Characterization of interconnects, Linear arrays, 2D mesh and torus, cliques)

Unit 4

Applications: Matrix-matrix multiply, Odd-Even sorting, distributed histogram, Breadth First search, Dijkstra's algorithm.

References

1. Grama, A., Gupta, A., Karypis, G., Kumar, V., *Introduction to Parallel Computing*, 2nd edition, Addison-Wesley, 2003.
2. Quinn, M., *Parallel Programming in C with MPI and OpenMP*, 1st Edition, McGraw-Hill, 2017.
3. Revdikar, L., Mittal, A., Sharma, A., Gupta, S., *A Naïve Breadth First Search Approach Incorporating Parallel Processing Technique For Optimal Network Traversal*, International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 5, May 2016

Additional references

(i) B. Parhami, Introduction to Parallel Processing: Algorithms and Architectures, Plenum, 1999, Springer.

Suggested Practical List

1. Implement Matrix-Matrix Multiplication in parallel using OpenMP
2. Implement distributed histogram Sorting in parallel using OpenMP
3. Implement Breadth First Search in parallel using OpenMP
4. Implement Dijkstra's Algorithm in parallel using OpenMP

DSC17/GE7d/DSE8e: CLOUD COMPUTING

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Cloud Computing	4	3	0	1	Pass in Class XII	NIL

Course Objective:

The objective of an undergraduate cloud computing course is to provide students with a comprehensive understanding of cloud computing technologies, services, and applications.

Course Learning Outcomes:

Learning outcomes for an undergraduate course on cloud computing may include:

1. Knowledge of the fundamental concepts and principles of cloud computing, including virtualization, scalability, reliability, and security.
2. Ability to design, develop, and deploy cloud-based applications using popular cloud platforms and services.
3. Familiarity with cloud computing architectures, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
4. Visualize the economic, legal, and ethical implications of cloud computing, including issues related to data privacy, ownership, and security.
5. Ability to evaluate and select cloud-based solutions based on their technical, economic, and business requirements.
6. Understanding of the broader societal and environmental impacts of cloud-based services and applications.

Syllabus:

Unit 1: Overview of Computing Paradigm
Recent trends in Computing : Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing,

Unit 2: Introduction to Cloud Computing
Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers, Benefits and limitations of Cloud Computing,

Unit 3: Cloud Computing Architecture

Comparison with traditional computing architecture (client/server), Services provided at various levels, Service Models- Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS), How Cloud Computing Works, Deployment Models- Public cloud, Private cloud, Hybrid cloud, Community cloud, Case study of NIST architecture.

Unit 4: Case Studies

Case study of Service model using Google Cloud Platform (GCP), Amazon Web Services (AWS), Microsoft Azure, Eucalyptus.

Unit 5: Cloud Computing Management

Service Level Agreements(SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling.

Unit 6: Cloud Computing Security

Infrastructure Security- Network level security, Host level security, Application level security, Data security and Storage- Data privacy and security Issues, Jurisdictional issues raised by Data location, Authentication in cloud computing.

References:

1. Thomas Erl, Ricardo Puttini and Zaigham Mahmood, Cloud Computing: Concepts, Technology and Architecture, Publisher: PHI, 2013.
2. Rajkumar Buyya, James Broberg, and Andrzej Goscinski, Cloud Computing: Principles and Paradigms, Wiley, 2013.
3. Boris Scholl, Trent Swanson, and Peter Jausovec, Cloud Native: Using Containers, Functions, and Data to Build Next-Generation Applications, Publisher : Shroff/O'Reilly, 2019.

Additional References:

1. *Cloud Computing Bible,* Barrie Sosinsky*, Wiley-India,* 2010
2. *Cloud Computing: Principles and Paradigms,* Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski*, Wile,* 2011

3. *Cloud Computing: Principles, Systems and Applications, Editors:* Nikos Antonopoulos, Lee Gillam*, Springer,* 2012
4. *Cloud Security: A Comprehensive Guide to Secure Cloud Computing,* Ronald L. Krutz, Russell Dean Vines*, Wiley-India,* 2010

Suggested Practical List:

1. Create virtual machines that access different programs on same platform.
2. Create virtual machines that access different programs on different platforms .
3. Working on tools used in cloud computing online-
   a) Storage
   b) Sharing of data
   c) manage your calendar, to-do lists,
   d) a document editing tool
4. Exploring Google cloud
5. Exploring microsoft cloud
6. Exploring amazon cloud

Annexure-5

Summary of Changes in Programmes under UGCF

Changes in the position of DSC/DSE/GE papers for the students admitted in the year 2023-2024, and onwards:

1. The courses on Numerical Optimization and Introduction to Parallel Programming being too theoretical have been made elective. In their places, the courses on Computer Graphics and Cloud Computing have been made core courses.

2. The position of the courses **Internet Technologies: Web App Design and Development** and **Advanced Web Programming** has been swapped for better sequencing.

NEP-UGCF

Changes in position of DSC/DSE/GE papers for the students admitted in the year 2023-2024, and onwards:

SNo	Course Name	Existing Position	Recommended Minutes
1	Numerical Optimization	DSC09 in B.Sc(H) Computer Sc. in IIIrd Semester	DSE in VI Semester
2	Computer Graphics	DSE in VI Semester	DSC09 in B.Sc(H) Computer Sc. in IIIrd Semester
3	Introduction to Parallel Programming	DSC18 in B.Sc(H) Computer Sc. in VIth Semester	DSE in VI Semester
4	Cloud Computing	DSE in VI Semester	DSC18 in B.Sc(H) Computer Sc. in VIth Semester
5.	Internet Technologies: Web App Design and Development	GE6b: Internet Technologies: Web App Design and Development in VIth semester	GE5b: Internet Technologies: Web App Design and Development in Vth semester
6.	Advanced Web Programming	GE5b: Advanced Web Programming in Vth semester	GE6b: Advanced Web Programming in VIth semester

Addition to GE in Semester VI:

Data Privacy course has been added as G.E. in VI semester.