UNIVERSITY OF DELHI

MASTER OF COMPUTER APPLICATIONS (MCA)

CONTENTS

I. About the Department

Department of Computer Science was established at the University of Delhi, in 1981, with the objective of imparting quality education in the field of Computer Science. With rapidly evolving technology and continuous need for innovation, the department has been producing quality professionals, who are currently holding important positions in the Information Technology industry both in India and abroad.

The Department started Master of Computer Applications (MCA) programme in the year 1982, which was among the first such programmes in India. The MCA programme focuses on providing a sound theoretical background as well as good practical exposure to students in the relevant areas. It is intended to provide a modern, industry-oriented education in applied computer science. It aims at producing trained professionals who can successfully meet the demands of the IT industry. They obtain skills and experience in up-to-date approaches to analysis, design, implementation, validation, and documentation of computer software and hardware.

The Department started M.Sc. Computer Science course in the year 2004 with the aim to develop core competence in Computer Science and to prepare the students to take up challenges of research and development. The students develop the ability to apply a high level of theoretical expertise and innovation to complex problems and application of new technologies. M.Sc. programme has been designed to teach the mathematical principles of specification, design and efficient implementation of both software and hardware.

The Department also offers Doctor of Philosophy (Ph.D.) programme, aimed at producing quality researchers in several diverse branches of Computer Science. The Department also coordinates B.Sc. (H) Computer Science, B.Sc. Physical Science (Computer Science) and other courses taught at constituent colleges of University of Delhi.

Introduction to CBCS (Choice Based Credit System) Choice Based Credit System:

The CBCS provides an opportunity for the students to choose courses from the prescribed courses comprising core, elective/minor or skill-based courses. The courses will be evaluated following the grading system, which is considered to be better than the conventional marks system. Grading system provides uniformity in the evaluation and computation of the Cumulative Grade Point Average (CGPA) based on a student's performance in examinations which enables the student to move across institutions of higher learning. The uniformity in evaluation system also enables the potential employers in assessing the performance of the candidates.

Definitions:

(i) 'Academic Programme' means an entire course of study comprising its programme structure, course details, evaluation schemes etc. designed to be taught and evaluated in a teaching Department/Centre or jointly under more than one such Department/ Centre

(ii) 'Course' means a segment of a subject that is part of an Academic Programme

(iii) 'Programme Structure' means a list of courses (Core, Elective, Open Elective) that makes up an Academic Programme, specifying the syllabus, Credits, hours of teaching, evaluation and examination schemes, minimum number of credits required for successful completion of the programme etc. prepared in conformity to University Rules, eligibility criteria for admission

(iv) 'Core Course' means a course that a student admitted to a particular programme must successfully complete to receive the degree and which cannot be substituted by any other course

(v) 'Elective Course' means an optional course to be selected by a student out of such courses offered in the same or any other Department/Centre

(vi) 'Open Elective' means an elective course which is available for students of all programmes, including students of the same department. Students of other departments will opt these courses subject to fulfilling of eligibility of criteria as laid down by the Department offering the course.

(vii) 'Credit' means the value assigned to a course which indicates the level of instruction; One-hour lecture per week equals 1 Credit, 2 hours practical class per week equals 1 credit. Credit for a practical could be proposed as part of a course or as a separate practical course

(viii) 'SGPA' means Semester Grade Point Average calculated for an individual semester.

(ix) 'CGPA' is Cumulative Grade Points Average calculated for all courses completed by the students at any point of time. CGPA is calculated each year for both the semesters clubbed together.

(x) 'Grand CGPA' is calculated in the last year of the course by clubbing together a CGPA of two years, i.e., four semesters. Grand CGPA is being given in Transcript form. To benefit the student a formula for conversion of Grand CGPA into %age marks is given in the Transcript.

MCA Programme Details:

Programme Objectives (POs):

Master of Computer Applications (MCA) is a full-time four-semester course, which includes one semester of project work in the 4th semester. The objective of 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 overall development of the student, while concentrating on imparting 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 for designing and developing innovative applications for the betterment of the 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 100 credits for the completion of course and the award of degree.

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

<u>Course Credit Scheme</u>

Semester	**Core Courses**			**Elective Course**			**Open Elective Course**			**Total Credits**
	No. of papers	Credits (Th+P+T)	**Total Credits**	No. of papers	Credits (Th+P+T)	**Total Credits**	No. of papers	Credits (Th+P+T)	**Total Credits**	
I	5	20+5+0	**25**	0	0	**0**	0	0	**0**	**25**
II	5	20+5+0	**25**	0	0	**0**	0	0	**0**	**25**
III	2	8+2+0	**10**	2	8+2+0	10	1	4+1+0	**5**	**25**
IV	Project	0	**25**	0	0	**0**	0	0	**0**	**25**
Total Credits for the Course			**85**			**10**			**5**	**100**

* For each Core and Elective Course there will be 4 lecture hours of teaching per week.
* Open Electives to the maximum total of 8 credits.
* Duration of examination for each course shall be 3 hours.
* Each course will be of 100 marks out of which 70 marks shall be allocated for semester examination and 30 marks for internal assessment.
* Each student shall carry out a project in the fourth semester. The projects will be carried out under the supervision of teachers of the department. When the project is carried out in an external organization (academic institution/ industry), a supervisor will 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 shall carry 500 marks distributed as follows:
  Dissertation: 50% weightage
  Viva-voce: 50% weightage
* To be eligible to pass a course and earn credits for it, a student must satisfy the criteria laid down by the University.
* Examination for courses specified in the odd (even) semesters shall be conducted only in the respective odd (even) semesters.
* Promotion Criteria: As laid down by the University.

* Award of degree: In order to be eligible for the award of the Master of Computer Applications (MCA) degree, a student must earn all the credits (100) as per the structure of the course, specified in the above table.

Semester wise Details of MCA Course

Semester I

Semester I					
	Number of core courses	5			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Practical	Tutorial	**Total**
MCAC101	Object Oriented Programming	4	1	0	**5**
MCAC102	Computer Systems Architecture	4	1	0	**5**
MCAC103	Data Structures	4	1	0	**5**
MCAC104	Database Systems	4	1	0	**5**
MCAC105	Mathematical Foundations of Computer Applications	4	1	0	**5**
	Total credits in core course	25			
	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	**25**			

Semester II

Semester II					
	Number of core courses	5			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Practical	Tutorial	**Total**
MCAC201	Design and Analysis of Algorithms	4	1	0	**5**
MCAC202	Data Communication and Computer Networks	4	1	0	**5**
MCAC203	Operating Systems	4	1	0	**5**
MCAC204	Data Analysis and Visualization	4	1	0	**5**
MCAC205	Web Technologies	4	1	0	**5**
	Total credits in core course	25			
	Number of elective courses	0			
	Total credits in elective courses	0			
	Number of open electives	0			
	Credits in each open elective	0			
	Total credits in open elective	0			
	Total credits in Semester II	25			

Semester III

Semester III					
	Number of core courses	2			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Practical	Tutorial	**Total**
MCAC301	Cyber Security	4	1	0	**5**
MCAC302	Software Engineering	4	1	0	**5**
	Total credits in core course				**10**
	Number of elective courses	2			
	Credits in each elective	Theory	Practical	Tutorial	**Total**
	Elective course 1	4	1	0	**5**
	Elective course 2	4	1	0	**5**
	Total credits in elective courses				**10**
	Number of open elective	1			
	Credits in each open elective	Theory	Practical	Tutorial	**Total**
	Open Elective I	4	1	0	**5**
	Total credits in open elective courses				**5**
	Total credits in Semester III				**25**

List of Elective Courses for Semester III		
Course Code	**Course Title**	**Th-P-T**
MCAE301	Compiler Design	4-1-0
MCAE302	Artificial Intelligence	4-1-0
MCAE303	Information Security	4-1-0
MCAE304	Network Science	4-1-0
MCAE305	Deep Learning	4-1-0
MCAE306	Interpretable Learning	4-1-0

MCAE307	Natural Language Processing	4-1-0
MCAE308	Graph Theory	4-1-0
MCAE309	Text Analytics	4-1-0
MCAE310	Automata Theory	4-1-0
MCAE311	Combinatorial Optimization	4-1-0
MCAE312	Digital Watermarking and Steganography	4-1-0
MCAE313	Quantum Computing	4-1-0
MCAE314	Computer Graphics	4-1-0
MCAE315	Parallel and Distributed Computing	4-1-0
MCAE316	Data Mining	4-1-0
MCAE317	GPU Programming	4-1-0
MCAE318	Neural Networks	4-1-0
MCAE319	Machine Learning Applications	4-1-0
MCAE320	Digital Image Processing	4-1-0
MCAE321	Internet of Things	4-1-0

List of Open Elective Courses for Semester III		
Course Code	**Course Title**	**Th-P-T**
MCAO301	Java Programming	4-1-0
MCAO302	Data Mining for Business Applications (for other departments)	4-1-0
MCAO303	Data Science	4-1-0

Semester IV

Semester IV		
	Number of core courses	1
Course Code	**Course Title**	
MCAC401	Project work	25

	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	**25**

Total Credits = 25 +25 + 25 + 25 =100

Selection of Elective Courses:

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

Eligibility and Mode of Admissions for the MCA Programme:

To be decided by the University in every academic year.

Number of seats in the MCA programme: 58

The candidates who are appearing in the final year examinations of the Bachelor's degree on the basis of which admission is sought are eligible to apply in all categories mentioned above.

1. ***Relaxation will be given to the candidates belonging to SC, ST and OBC category as per the University rules***
2. ***For preparing the final merit list, 85% weightage will be given to the score in the Entrance Exam and 15% weightage will be given to the score in the Interview.***

Assessment of Students' Performance and Scheme of Examinations:

A. English shall be the medium of instruction and examination.
B. Examinations shall be conducted at the end of each semester as per the academic calendar notified by the University of Delhi.
C. Assessment of students' performance and scheme of evaluation shall be as per University rules.

Pass Percentage and Promotion Criteria

As per University rules.

Conversion of Marks into Grades

A. Grade Points: Grade point table as per University Examination rules.
B. CGPA Calculation

   As per University Examination rules.
C. SGPA Calculation

   As per University Examination rules.

D. Grand CGPA Calculation

   As per University Examination rules.

E. Conversion of Grand CGPA into Marks

   As per University Examination rules.

F. As notified by competent authority the formula for conversion of Grand CGPA into marks is:

   Final %age of marks = CGPA based on all four semesters × 9.5

Division of Degree into Classes

Post Graduate degree to be classified based on CGPA obtained into various classes as notified into Examination policy.

Attendance Requirement

As per University rule**.**

Span Period

As per University rule.

Guidelines for the Award of Internal Assessment Marks for the MCA Programme

Performance of the students will be evaluated based on a comprehensive system of continuous evaluation. For each course, there shall be a minor test, assignments/ laboratory work. There shall be a monitoring committee to be constituted at the beginning of each semester to monitor the internal assessment.

IV: Content Details for MCA Programme

SEMESTER – I

MCAC101: OBJECT ORIENTED PROGRAMMING [4-1-0]

Course Outcomes:

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

CO1: apply object-oriented paradigm for problem solving.

CO2: select a suitable programming construct and in-built data structure for the given problem.

CO3: design, develop, document, and debug modular programs.

CO4: use recursion as a programming 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; use of mutable and immutable structures. strings, lists, sets, tuples and dictionary, and associated operations testing, and debugging a program.

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

Unit-IV Object Oriented Programming: Use of classes, inheritance, and operator overloading in problem solving.

Unit-V Visualization using 2D and 3D graphics: Visualization using graphical objects like point, line, histogram, 3D objects, animation.

Unit-VI 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. J.V. Guttag, **Introduction to Computation and Programming Using Python**: **With Application to Understanding Data**, MIT Press, 2016.
3. S. Taneja, N. Kumar, **Python Programming: A Modular Approach,** With Graphics Database, Mobile, and Web Applications, Pearson Education, 2017.
4. C. S. Horstmann, **Core Java- Volume I, Fundamentals**. Pearson Education, 2016.
5. K. Arnold and J. Gosling, **The Java™ Programming Language**, Addisson Wesley, 2005.
6. Bjarne Stroustrup, **The C++ Programming Language (4th Edition)** Addison-Wesley, 2013

MCAC102: COMPUTER SYSTEMS ARCHITECTURE [4-1-0]

Course Outcomes:

On completion of 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, subtractor, encoders, decoders, multiplexers, demultiplexers.

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

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 & Organization**, 3rd Edition, McGraw-Hill Education, 2017.

MCAC103: DATA STRUCTURES [4-1-0]

Course Outcomes:

On completion of 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 and skip lists.

CO2: apply data structures like Tries, Prefix- and Suffix-trees to different applications.

CO3: identify best suited data structure for the problem at hand.

CO4: identify the programming constructs to optimize the performance of the data structure in different scenarios.

Syllabus:

Unit-I Basic data Structures: Primitive Data Types, Abstract Data Types, Arrays - Static and Dynamic, 2D Arrays, Linked Lists - Single, Doubly-linked, Circular; Stacks and Queues using arrays and linked lists; operations, their analysis; Applications to searching & sorting.

Unit-II Trees: Binary Tree, Binary Search Tree, Height Balanced Trees: AVL/RB Tree, 2-3Trees, B and B+ Trees, Splay Trees, Heaps, Priority Queues, Mergeable heaps, Tries, Prefix and Suffix Trees, Skip Lists; operations, their analysis, applications to searching.

Unit-III Sets: Sets, Multisets, Maps, Hash Tables, Dictionaries.

Unit-IV Graphs: Representation of Graphs, Searching in Graphs – BFS and its applications, DFS and its applications.

Readings:

1. Goodrich, M., Tamassia, R. and Mount D, **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 [4-1-0]

Course Outcomes:

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

CO1: understand basic database concepts, including the structure and operation of the relational data model.

CO2: apply logical database design principles, including E-R/EE-R diagrams, conversion of ER diagrams to relations.

CO3: understand the concepts of integrity constraints, relational algebra, relational domain & tuple calculus, data normalization.

CO4: construct simple and moderately advanced database queries using Structured Query Language (SQL).

CO5: understand the concept of a database transaction including concurrency control, backup and recovery, and data object locking.

CO6: design and implement database projects.

Syllabus:

Unit-I Basic Concepts: Data modeling for a database, abstraction and data integration, three level architecture of a DBMS.

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

Unit-III Relational Model & Relational Data Manipulations: Relation, conversion of ER diagrams to relations, integrity constraints, relational algebra, relational domain & 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 databases, XML databases.

<u>Readings:</u>

1. A. Silberschatz, H. Korth and S. Sudarshan**, Database System Concepts,** 6th Edition**,** McGraw Hill, 2014.

2. R. Elmasri and S. B. Navathe, **Fundamentals of Database Systems**, 7th Edition, Pearson, 2016.

3. R. Ramakrishnan and J. Gehrke, **Database Management Systems,** 3rd Edition, McGraw Hill, 2014.

4. Philip Lewis, Arthur Berstein and Michael Kifer, **Databases and Transaction Processing-An application oriented approach**, Prentice Hall, 2003

<u>MCAC105: MATHEMATICAL FOUNDATIONS OF COMPUTER APPLICATIONS [4-1-0]</u>

<u>Course Outcomes:</u>

On successful completion of this course, the 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 eigen values of a matrix; apply matrix algebra to solve problems in sub-disciplines of computer science.
CO3: perform data analysis in a probabilistic framework.
CO4: visualize and model the given problem using mathematical concepts covered in the course.

Syllabus:

Unit-I Linear Algebra:
Vector Spaces: Introduction, Subspace, Linear Independence and Dependence, Basis and Dimensions, Convex set, Rank of a matrix, System of linear equations, Application in text analytics.
Inner product Space and Orthogonality: Orthogonal bases, Projection, Gram-Schmidt orthogonality process.
Linear Mapping: Introduction, Kernel and Image space of a linear map, Matrix associated with linear map, Eigenvalues and Eigenvectors, PCA, SVD, Application to Image Processing.

Unit-II Probability and Statistics:
Introduction, Conditional Probability, Independent events, Bayes' theorem
Introduction to Descriptive and Inferential Statistics, Describing Data as Frequency tables, Relative frequency

tables and graphs, Scatter diagram, Grouped data, Histograms, Ogives, Percentiles, Box Plot, Coefficient of variation, Skewness, Kurtosis

Unit-III Random variable:
Introduction, Discrete (Bernoulli, Binomial, Multinomial, Poisson, Geometric, Negative Binomial), Continuous (Uniform, Exponential, Normal, Gamma), Expectation, variance, Conditional probability and conditional expectation, Central Limit Theorem, Markov and Chebyshev's inequality.

Readings:

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

SEMESTER – II

MCAC201: DESIGN AND ANALYSIS OF ALGORITHMS [4-1-0]

Course Outcomes:

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

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

CO2: analyze the strengths and weaknesses of each technique.

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

CO4: demonstrate correctness of algorithms and analyze their time complexity theoretically as well as practically.

CO5: model simple problems as graphs and solve them using Graph Algorithms.

CO6: analyze algorithms in the probabilistic framework.

CO7: appreciate that certain problems are too hard to admit fast solutions

Syllabus:

Unit-I Review of Growth of Functions

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: Recurrence Relation, Master's Theorem, Recursion Trees; Binary Search, Merge sort and Quick sort – time complexity and proof of correctness.

Unit-IV Lower bounding techniques: Decision Trees.

Unit-V Linear Sorting: Count Sort, Radix Sort, Bucket Sort.

Unit-VI More on Divide and Conquer: Integer Multiplication, Convolution and Fast-Fourier Transform.

Unit-VII Greedy Algorithms: Interval Scheduling, Minimum Spanning Trees – Prim's algorithm, Kruskal Algorithm, Shortest Path Problem – Djikstra's algorithm.

Unit-VIII Dynamic Programming: Weighted Interval Scheduling, Segmented Least Square problem, Knapsack problem, Shortest Paths.

Unit-IX String Processing: Finite Automata method, KMP.

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

Unit-XI Introduction to Complexity Classes: P, NP, NP-Hard, NP-Complete.

Readings:

1. J. Kleinberg and E.Tardos, **Algorithm Design**, 1st Edition 2013., Pearson Education India,

2. 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.

3. Sara Baase, Allen Van Gelder, **Computer Algorithm – Introduction to Design and Analysis**, 3rd edition , 2002, Pearson Education.

4. Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, **Algorithms**, 1st Edition, 2017, Tata McGraw Hill.

5. Richard Johnsonbaugh and Marcus Schaefer, **Algorithms**, 2014, Pearson Education India.

MCAC202: DATA COMMUNICATION AND COMPUTER NETWORKS [4-1-0]

Course Outcomes:

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

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

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

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

CO4: develop some basic 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 also develop basic 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 & 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 Datalink 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: OPERATING SYSTEMS [4-1-0]

Course Outcomes:

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

CO1: describe basic functions of an Operating System.

CO2: distinguish between different types of Operating Systems so as 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 simple algorithms for managing computer resources.

Syllabus:

Unit-I Introduction: Operating System as a resource manager, operating systems services, system calls, operating system classifications, operating systems architectures.

Unit-II Processor Management: Process overview, process states and state transition, multi-programming, multi-tasking, levels of schedulers and scheduling algorithms. Process Synchronization-Critical section and mutual exclusion problem, classical process synchronization problems, deadlock prevention. Multithreading.

Unit-III Memory Management: absolute and relative code, address translation, memory management techniques- partition, paging, segmentation, virtual memory. Static and dynamic memory management.

Unit-IV Device Management: Goals of I/O software, Design of device drivers- interrupt service routines, upper half of kernel software, lower half of kernel software.

Unit-V File Management: Overview of file management system, disk space management, directory structures, file sharing and protection, access control lists, protection models.

Readings:

1. Silberschatz, Galvin,and Gagne, **Operating Systems concepts** , Wiley ,2009.

2. Gary Nutt, Nabendu Chaki, Sarmistha Neogy, **Operating Systems: A Modern Approach (3rd ed.)** , Addison Wesley, 2009.

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

MCAC204: DATA ANALYSIS AND VISUALIZATION [4-1-0]

Course Outcomes:
On successful completion of the course, the students will be able to :
CO1:Use data analysis tools in the pandas library.
CO2:Load, clean, transform, merge and reshape data.
CO3:Create informative visualization and summarize data sets.
CO4:Analyze 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 of Python shell iPython and Jupyter Notebook. Essential Python Libraries: NumPy, pandas, matplotlib, SciPy, scikit-learn, statsmodels
Unit 2 Getting Started with Pandas: Arrays and vectorized computation, Introduction 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
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.
Unit 5 Advanced Pandas: Categorical Data, Advanced GroupBy Use, Techniques for Method Chaining

Readings
1. McKinney, W.(2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython. 2nd edition. O’Reilly Media.
2. O’Neil, C., & Schutt, R. (2013). Doing Data Science: Straight Talk from the Frontline, O’Reilly

Media.

MCAC205: WEB TECHNOLOGIES [4-1-0]

Course Outcomes:

On completion of 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: work with client-side web technologies - HTML, CSS, Javascript, DOM, XML, XSLT for front-end development.

CO4: work with server-side web technologies.

CO5: use 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 basic syntax, variables & 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 JSP and server-side scripting, accessing MySQL / Oracle database from 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 ed.), McGraw-Hill Education, 2013.

3. Roger S Pressman and David Lowe, **Web Engineering: A Practitioner's Approach,** TMH, 2017.

4. Mark Pilgrim, **HTML5: Up and Running**, O'Reilly | Google Press, 2010 .

5. Jim Keogh, **J2EE: The Complete Reference**, McGraw Hill Education, 2017.

SEMESTER – III

MCAC301: CYBER SECURITY [4-1-0]

Course Outcomes:

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

CO1: state the need and scope for cyber laws.

CO2: enumerate various network attacks, and describe their sources, and mechanisms of prevention

CO3: describe the genesis of SCADA policies and their implementation framework

CO4: Carryout malware analysis using simulations.

Syllabus:

Unit-I Introduction: Cyberspace, Internet, Internet of things, Cyber Crimes, cyber criminals, Cyber security, Cyber Security Threats, Cyber laws and legislation, Law Enforcement Roles and Responses.

Unit-II Network Attacks: 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.

Unit-III Introduction to SCADA (supervisory control and data acquisition) Understanding SCADA security policies, SCADA Physical and Logical Security, Understanding differences between physical and logical security, Define perimeter controls and terms, Define various security zones, Understand communication cyber threats, Understand firewall, architectures.

Unit-IV Introduction to Malware, Malware Analysis: Static Analysis, Code Review, Dynamic Analysis, Behavioral analysis of malicious executable, Sandbox Technologies, Reverse-engineering malware, Defeat anti-reverse engineering technique, automated analysis, intercepting network connections, Network flow analysis, Malicious Code Analysis, Network analysis, Anti-disassembling techniques, Identifying assembly logic structures with a disassembler.Malware Handling: Malicious Documents and Memory Forensics - Reverse engineering of malicious executable using memory forensic techniques, Analyze malicious Microsoft Office (Word, Excel, PowerPoint) and Adobe PDF documents, Analyzing memory to assess malware characteristics and reconstruct infection artifacts. Using memory forensics to analyze rootkit infections, Legal & Ethical Issues - Reinforce understanding and the application of discipline specific legal and ethical issues, Reverse Engineering Malware (REM) Methodology.

Readings:

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

2. Jonathan Clough, **Principles of Cybercrime**, Cambridge University Press, 24-Sep-2015

3. Jie Wang, Zachary A. Kissel**, Introduction to Network Security: Theory and Practice**, Wiely 2016.

4. Michael Bazzell , **Open Source Intelligence Techniques: Resources for Searching and Analyzing Online Information**, 2nd edition , CreateSpace Independent Publishing Platform,2014.

5. Robert Radvanovsky, Jacob Brodsky, **Handbook of SCADA/Control Systems Security**, CRC Press, 2013.

6. Ed Skoudis , Lenny Zeltser ,**Malware: Fighting Malicious Code**, Prentice Hall Series in Computer Networking and Distributed, 2003

7. Michael Sikorski, Andrew Honig ,**Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software** 2012, No Starch Press, San Fransisco.

MCAC302: SOFTWARE ENGINEERING [4-1-0]

Course Outcomes:

On completion of the course, the student is expected to:

CO1: demonstrate an understanding of software engineering layered technology and software process models that provide a basis for the software development lifecycle.

CO2: apply agile development methods for developing software.

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

CO4: practice system modeling techniques and object-oriented design for software development.

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

CO6: appreciate software project management that includes project planning, project estimation techniques, risk management, quality management, and configuration management.

CO7: work as an individual and/or in team to develop and deliver quality software.

Syllabus:

Unit-I Software Engineering: The software crisis, principles of software engineering, 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, system classifications.

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 ed.), McGraw-Hill, 2010.

2. I. Sommerville, **Software Engineering** (10th ed.), Pearson Education, 2015.

3. R. Mall, **Fundamentals of Software Engineering** (4th ed.), Prentice-Hall of India, 2014.

4. K.K. Aggarwal and Y. Singh, **Software Engineering** (3rd ed.), New Age International Publishers, 2008.

5. P. Jalote, **An Integrated Approach to Software Engineering** (3rd ed.), Narosa Publishing House, 2005.

6. N.S. Godbole, **Software Quality Assurance: Principles and Practice for Students**, Alpha Science International Limited, 2004.

MCAE301: COMPILER DESIGN [4-1-0]

Course Outcomes:

On completion of this course, the 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 compiler tools like lex and yacc for implementing syntax directed translator.

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. Alfred V. Aho, Ravi Sethi, D. Jeffrey Ulman, Monica S. Lam, **Principles, Techniques and Tools**, Pearson Education India, 2nd edition,, 2013.

2. A.V. Aho, M. S. Lam, R. Sethi and J. D. Ullman, **Compilers,** Pearson, 2016.

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

MCAE302: ARTIFICIAL INTELLIGENCE [4-1-0]

Course Outcomes:

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

CO1: describe various approaches to Artificial Intelligence.

CO2: design intelligent agents.

CO3: distinguish between Utility based agents and Goal based agents.

CO4: describe and apply concepts, methods, and theories of search, heuristics, games, knowledge representation, planning.

CO5: acquire basics knowledge of Natural language processing.

CO6: understand the limitations of Artificial Intelligence techniques.

Syllabus:

Unit-I Introduction: Introduction to Artificial Intelligence, various definitions of AI, AI Applications and Techniques, Turing Test and Reasoning - forward & backward chaining.

Unit-III Intelligent Agents: Introduction to Intelligent Agents, Rational Agent, their structure, , reflex, model-based, goal-based, and utility-based agents, behavior and environment in which a particular agent operates.

Unit-IV Problem Solving and Search Techniques: Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, iterative deepening, uniform cost search, Hill climbing and its Variations, simulated annealing, genetic algorithm search.

Heuristics Search Techniques: Best First Search, A* algorithm, AO* algorithm, Minmax & game trees, refining minmax, Alpha – Beta pruning, Constraint Satisfaction Problem, Means-End Analysis.

Unit-V Knowledge Representation: Introduction to First Order Predicate Calculus, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, semantic networks, Frames system, Production Rules, Conceptual Graphs, Ontologies.

Unit-VI Planning: Basic representation for planning, symbolic-centralized vs reactive-distributed, partial order planning algorithm.

Unit-VII Reasoning with Uncertain Knowledge: Different types of uncertainty - degree of belief and degree of truth, various probability constructs - prior probability, conditional probability, probability axioms, probability distributions, and joint probability distributions, Bayes' rule, other approaches to modeling uncertainty such as Dempster-Shafer theory and fuzzy sets/logic.

Unit-VIII Understanding Natural Languages: Components and steps of communication, contrast between formal and natural languages in the context of grammar, parsing, and semantics, Parsing Techniques, Context-Free and Transformational Grammars.

Readings:

1. S. Russell and P. Norvig, **Artificial Intelligence: A Modern Approach, 3rd edition**, Pearson Education, 2015.

2. Elaine Rich and Kelvin Knight, Artificial Intelligence, 3rd edition, Tata McGraw Hill , 2017.

3. DAN.W. Patterson, **Introduction to A.I. and Expert Systems** – PHI, 2007.

4. Michael Wooldridge, **An Introduction to MultiAgent Systems**, 2nd edition, John Wiley & Sons, 2009.

5. Fabio Luigi Bellifemine, Giovanni Caire, Dominic Greenwood, **Developing Multi-Agent Systems with JADE**, Wiley Series in Agent Technology, John Wiley & Sons, 2007.

6. W.F. Clocksin and C.S. Mellish, **Programming in PROLOG**, , 5th edition, Springer, 2003.

7. Saroj Kaushik, **Logic and Prolog Programming**, New Age International Publisher, 2012.

8. Ivan Bratko, **Prolog Programming for Artificial Intelligence**, Addison-Wesley, Pearson Education, 4th edition, 2011.

MCAE303: INFORMATION SECURITY [4-1-0]

Course Outcomes:

On completion of this course, the 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:

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 Intrusion detection and prevention.

Readings:

1. W. Stalling, **Cryptography and Network Security Principles and Practices** (7th ed.), 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 SL Pfleeger, Jonathan Margulies, **Security in Computing** (5th ed.), Prentice-Hall of India, 2015.

MCAE304: NETWORK SCIENCE [4-1-0]

Course Outcomes:

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

CO1: appreciate the ubiquity of graph data models.

CO2: understand the structural features of a network and their impact on network functions.

CO3: identify community structures in networks

CO4: solve real world problems modeled as complex networks

Syllabus:

Unit-I Introduction: Introduction to complex systems and networks, modelling 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.

MCAE305: DEEP LEARNING[4-1-0]

Course Outcomes:

On completion of 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: analyze 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 sub-gradient 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 Structuring Machine Learning Projects: Orthogonalization, evaluation metrics, train/dev/test distributions, size of the dev and test sets, cleaning up incorrectly labeled 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 306: Interpretable Learning [4-1-0]

Course Outcomes:

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

CO1: distinguish between local and global explanations.
CO2: choose a type of explanation method for a given situation.
CO3: provide mathematical reasoning behind different explanation methods.
CO4: to carry out what if-analysis of a machine learning model by generating counterfactuals.
CO5: use programming tools for generating explanations of machine learning models.

Unit-I Introduction to Interpretability: Motivation of interpretability; taxonomy of interpretability methods; scope of interpretability- local explanation vs global explanation; properties of explanation methods and individual explanations; model agnosticity: model-specific vs model-agnostic; dataset type- image, textual, speech, tabular, and graph-based; type of explanations.

Unit-II Inherently Explainable Models: Linear regression, logistic regression, Generalized Linear Models (GLM), Generalized Additive Models (GAM), decision tree, rule-based classifier, naive Bayes classifier, k-nearest neighbours.

Unit-III Model-Agnostic Post-hoc Methods: Global model-agnostic methods: Partial Dependence Plot (PDP), Accumulated Local Effect Plot (ALE), feature interaction, functional decomposition, permutation feature importance; local model-agnostic methods: SHAP (SHapley Additive exPlanations), LIME (Local Interpretable Model-Agnostic Explanations), Counter-Factual Explanations.

Unit-IV Model-Specific Post-hoc Methods: Gradient, gradient*input, smooth-gradient, guided backpropagation, Layerwise Relevance Propagation (LRP), Layerwise Relevance Propagation-Epsilon (LRP-ε), deep Taylor, integrated gradient, DeepLIFT (Deep Learning Important FeaTures), deconvolution, Class Activation Mapping (CAM), Gradient-weighted Class Activation Mapping (Grad-CAM), Guided Gradient-weighted Class Activation Mapping (Guided Grad-CAM).

Unit-V Limitations of Explainable Methods: Faithfulness/fidelity, fragility, stability, usefulness in practice

Unit-VI Future of Machine Learning and Explainability: Open issues: Modelling uncertainty, generating stable explanations, issue of over-trust on model's explanations by domain experts and end-users, data leakage exposing sensitive information from the dataset, usefulness of the explanations in real-world.

Readings:

1. Molnar, Christoph. **Interpretable machine learning. A Guide for Making Black Box Models Explainable**, 2019, Lulu.com, https://christophm.github.io/interpretable-ml-book/.
2. Rothman, Denis, **Hands-On Explainable AI (XAI) with Python: Interpret, visualize, explain, and integrate reliable AI for fair, secure, and trustworthy AI apps**, Packt Publishing Ltd, 2020.

MCAE 307: NATURAL LANGUAGE PROCESSING [4-1-0]
Course Outcomes:

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

CO1	:	Understand and describe the evaluation of NLP systems.
CO2	:	Understand deep learning techniques used in NLP and apply them to solve machine translation and conversation problems.
CO3	:	Learn about major NLP issues and identify possible future areas of NLP research.

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

MCAE 308: GRAPH THEORY [4-0-1]

Course Outcomes:

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

CO1: model problems using different types of basic graphs like trees, bipartite graphs and planar graphs.

CO2: understand and identify special graphs like Euler graphs and Hamiltonian graphs.

CO3: have increased ability to understand various forms of connectedness in a graph

CO4: appreciate different graph-coloring problems and their solutions.

CO5: to model simple problems from real life as graph-coloring problems.

Syllabus:

Unit-I Fundamental Concepts: Definitions, 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.

Unit-II 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 labeling, minimum spanning trees (Kruskal's algorithm), shortest paths (Dijkstra's algorithm).

Unit-III 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.

Unit-IV 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.

Unit-V 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.

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

<u>Readings:</u>

1. Douglas B West, **Introduction to Graph Theory**, II Edition, 2017, Pearson.

2. Gary Chartrand and Ping Zhang, **Introduction to Graph Theory**, 2017, Tata McGraw Hill.

3. Jonathan L. Gross and Jay Yellen, **Graph Theory and Its Applications**, 2nd Edition, 2005, Chapman Hall (CRC).

4. The course will also be taught through various research papers.

<u>MCAE 309: TEXT ANALYTICS</u> [4-1-0]

<u>Course Outcomes:</u>

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

CO1: identify relevance of text to the information needs of diverse individuals, communities and organizations.

CO2: select appropriate technique to automatically process the text.

CO3: use open source text analytic tools to perform various text processing and analysis tasks.

CO4: develop simple text analysis tools and solve real world problems.

<u>Syllabus:</u>

Unit-I Introduction : Introduction to Natural Language Processing (NLP) and Text Analytics, Tokenization, Part-of-speech tagging, Stemming and lemmatization, NLP toolkits, Indexing text using inverted file index, Posting lists , Information retrieval, Authorship attribution, Sentiment analysis, Named-entity recognition, Document summarization.

Unit-II Document representation and Language models : Representation of the unstructured text documents, Heaps' and Zipf's Laws, Boolean vector representation, Bag-of-words model, tf-idf model, Term weighting , Scoring and ranking, Vector space model, Visualization techniques, Unigrams, bi-grams and n-grams, language models.

Unit-III Text categorization: Supervised text categorization algorithms, Naive Bayes, k Nearest Neighbor (kNN), Logistic Regression.

Unit-IV Text clustering: Clustering structure of a corpus of text documents, Hierarchical clustering, Centroid-based clustering. Topic Modeling- Latent Semantic Indexing (LSI).

Readings:

1. Ricardo Baeza – Yates, Berthier Ribeiro – Neto, **Modern Information Retrieval: The concepts and Technology behind Search**, (ACM Press Books), Second Edition, 2011.

2. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze, **Introduction to Information Retrieval**, Cambridge University Press, First South Asian Edition, 2008.

3. Steven Struhl, **Practical Text Analytics: Interpreting Text and Unstructured Data for Business Intelligence**, Kogan Page, 2015.

4. Matthew A. Russell, **Mining the Social Web**, O'Reilly Media, 2013.

MCAE310: AUTOMATA THEORY [4-0-1]

Course Outcomes:

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

CO1: describe the mathematical model of machines.

CO2: describe the concept of formal language and corresponding automaton.

CO3: identify ambiguity in grammar

CO4: construct a parse tree for the given grammar.

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 ed.), Pearson Education, 2015

3. P. Linz, **Introduction to Automata Theory, Languages, and Computation**, Jones & Bartlett, 2016.

MCAE 311: COMBINATORIAL OPTIMIZATION [4-1-0]

Course Outcomes:

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

CO1: model problems using linear and integer programs.

CO2: differentiate between the computational complexities of LP and IP.

CO3: apply polyhedral analysis to develop algorithms.

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

CO5: explain the mathematical theory forming the basis of algorithms for combinatorial optimization (particularly graph theoretic).

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:

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

2. Matousek and Gartner, **Understanding and Using Linear Programming**, 2007, Springer.

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

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

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

MCSE 312: DIGITAL WATERMARKING AND STEGANOGRAPHY [4-1-0]

Course Outcomes:

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

CO1: implement algorithms for steganographic and watermarking operations.

CO2: evaluate steganographic and watermarking algorithms for robustness.

CO3: describe various finger printing 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, Research history, 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 313: QUANTUM COMPUTING [4-1-0]

Course Outcomes:

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

CO1: describe the architecture of a quantum system.

CO2: describe a quantum Turing machine and compare it with the standard Turing Machine.

CO3: distinguish between a probabilistic and quantum algorithms.

CO4: describe the notion of error correction in quantum systems.

CO5: implement quantum algorithms like kickback algorithm, Deutsch-Jozsa algorithm, Simon's algorithm, fast factorization algorithms, quantum Fourier transform, Shor's Algorithm, Grover's search algorithm.

Syllabus:

Unit-I Introduction & Background: A brief history of computing including 'Turing Machines', Probabilistic systems, Quantum Systems.

Unit-II QUBITS and Framework of Quantum Systems: The State of a Quantum System, Quantum Bits, Quantum Registers, Quantum information, Quantum Turing Machine.

Unit-III Quantum Circuits: Boolean Circuits, Reversible Circuits, Quantum Circuit Model, Quantum Gates, Universal Sets of Quantum Gates, Efficiency in approximating unitary transformation, Implementing measurements with Quantum Gates.

Unit-IV Introduction to Quantum Algorithm: Probabilistic versus Quantum Algorithm, Phase Kick- Back Algorithm, The Deutsch Algorithm, The Deutsch-Jozsa Algorithm, Simon's Algorithm.

Unit-V Fast Factorization & Search Algorithms: Quantum Fourier Transform, Shor's Algorithm, The Correctness Probability. Grover's Search Algorithm.

Unit-VI Computational lower bound complexity for quantum circuits: General ideas, Polynomial representations, Quantum Circuit Lower Bound.

Unit-VII Introduction to quantum error Correction: Classical Error correction, Fault tolerance, Quantum Error Correction, Fault Tolerance Quantum computation.

Readings:

1. S Arora and B.Barak, **Computational complexity, A Modern approach**; Cambridge ,Cambridge University Press, 2009

2. Michael A. Nielsen, Isaac L. Chuang, **Quantum Computation and Quantum Information**: 10th Anniversary Edition, Cambridge University Press, 2011

3. Mika Hirvensalo, **Quantum Computing**, Springer, 2001.

4. P Kaye, R Laflamme, and M Mosca, **An Introduction to Quantum Computing,** Oxford University Press, 2007.

MCAE 314: COMPUTER GRAPHICS [4-1-0]
Course Outcomes:

On completion of 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.
CO7	:	to understand a typical graphics pipeline.

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: The 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, Warping (Only 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, Open GL Rendering Pipeline. Applications of Virtual Reality.

Readings:

1. Donald Hearn and M. Pauline Baker, "Computer Graphics with Open GL", Prentice Hall.
2. R. K Maurya, "Computer Graphics with Virtual Reality", Wiley

<u>MCAE 315: PARALLEL AND DISTRIBUTED COMPUTING</u> [4-1-0]

<u>Course Outcomes:</u>

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

CO1 describe architectures for parallel and distributed systems.

CO2 develop elementary parallel algorithms.

CO3 develop an application involving synchronization of communicating distributed process.

Syllabus:

Unit-I 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, global versus distributed memory, the PRAM shared-memory model, distributed-memory or graph models, basic algorithms for some simple architectures: linear array, binary tree, 2D mesh with shared variables.

Unit II Distributed Computing Architectures: Characteristics and goals of distributed computing, architectural styles: centralized, decentralized, and hybrid architectures, layered, object-based and service oriented, resource-based, publish-subscribe architectures, middleware organization: wrappers, interceptors, and modifiable middleware, system architecture, example architectures: network file system and web.

Unit-III Distributed Processes and Communication in Distributed Systems: Threads in distributed systems, principle of virtualization, clients: network user interfaces and client-side software for distribution transparency; servers: design issues, object servers, server clusters; code migration; Layered protocols, remote procedure call: RPC operation, parameter passing; message- oriented communication: transient messaging with sockets, message-oriented persistent communication; multicast communication: application-level tree-based multicasting, flooding-based multicasting, gossip-based data dissemination.

Unit IV Naming and Coordination in Distributed Systems: Names, identifiers, and addresses, flat naming, Structured naming, and attribute-based naming; coordination: clock synchronization, logical clocks, mutual exclusion: centralized, distributed, token-ring, and decentralized algorithms; election algorithms, location systems, distributed event matching, gossip-based coordination.

Unit-V Consistency and replication, Fault Tolerance, and Security: Introduction to consistency models and protocols, fault tolerance, and security issues in distributed systems.

Readings:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, 2nd edition, 2003, Addison-Wesley.

2. B. Parhami, Introduction to Parallel Processing: Algorithms and Architectures, Plenum, 1999, Springer.

3. M. van Steen, A. S. Tanenbaum, Distributed Systems, CreateSpace Independent Publishing Platform, 2017.

MCAE316: DATA MINING [4-1-0]

Course Outcomes:

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

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

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

CO3: implement and apply appropriate data mining algorithm for the given problem.

CO4: use programming tools (Weka/Python/R) for solving data mining problems.

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: Supervised learning/mining tasks, Decision trees, Decision rules, Bayesian classification, Instance-based methods, evaluation and Validation methods.

Unit-IV Clustering: Basic issues in clustering, k-means clustering, expectation maximization, 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 ed.), Morgan Kaufmann, 2011.
3. Charu C Agrawal, **Data Mining: The Textbook**, Springer, 2015.
4. Luis Torgo, **Data Mining with R Learning with Case Studies,** Second Edition, CRC Press, 2017.
5. Robert Layton, **Learning Data Mining with Python**, Second Edition,

MCAE 317: GPU PROGRAMMING[4-1-0]

Course Outcomes:

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

CO1: describe GPU architecture and parallel programming models.

CO2: implement fundamental GPU Algorithms – reduce, scan, and histogram.

CO3: analyze, and figure out the portion of programs being parallelizable.

CO4: write an efficient parallel algorithm to solve a given problem.

CO5: optimize GPU programs.

Syllabus:

Unit-I Introduction: Introduction to heterogeneous computing, overview of CUDA C/Python, and kernel-based parallel programming.

Unit-II Performance Issues: Memory model for locality, tiling for conserving memory bandwidth, handling boundary conditions, and performance considerations, simple matrix-matrix multiplication in CUDA environment.

Unit-III Introduction to OpenCL, operations such as vector addition using streams.

Unit-IV Applications: Parallel convolution pattern, parallel scan pattern, parallel histogram pattern and atomic operations, data transfer and task parallelism.

Readings:

1. Shane Cook, **CUDA Programming: A Developer's Guide to Parallel Computing with GPUs**, Elsevier; 2014.

2. Norman Matloff, **Parallel Computing for Data Science: With Examples in R, C++ and CUDA**, Chapman & Hall/CRC, 2015.

MCAE 318: Neural Networks [4-1-0]

Course Outcomes:

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

CO1: describe the role of neural networks in engineering, artificial intelligence, and cognitive modelling.

CO2: design single and multi-layer feed-forward neural networks for practical applications,

CO3: analyse performance of neural networks and tune various hyper-parameters.

Syllabus

Unit-I Introduction: Neuron as basic unit of neurobiology, McCulloch-Pitts model, Hebbian Hypothesis; limitations of single-layered neural networks.

Unit-II Supervised Learning: Single-layered neural networks, perceptron rule, review of gradient-descent algorithms; multi-layered neural networks: first order methods, backpropagation algorithm, second order methods, modelling sequences using recurrent neural networks, Hopefield networks, Boltzmann machines, restricted Boltzmann machines.

Unit-III Kernel methods and support vector machines: soft margin techniques.

Readings:

1. C. M. Bishop, **Pattern Recognition and Machine Learning**, Springer, 2010.

2. Simon O. Haykin, **Neural Networks and Learning Machines,** Pearson Education, 2016

MCAE319: MACHINE LEARNING APPLICATIONS[4-1-0]

Course Outcomes:

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

CO1: appreciate fundamental issues and challenges of supervised and unsupervised learning techniques.

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

CO3: appreciate the underlying mathematical relationships within and across Machine Learning algorithms.

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

Syllabus:

Unit-I Introduction: Learning theory, Hypothesis and target class, Inductive bias and bias-variance tradeoff, Occam's razor, Limitations of inference machines, Approximation and estimation errors, Curse of dimensionality, dimensionality reduction, feature scaling, feature selection methods.

Unit-II Regression: Linear regression with one variable, Linear regression with multiple variable, Gradient Descent, Logistic Regression, Polynomial regression, over-fitting, regularization. performance evaluation metrics, validation methods.

Unit-III Classification: Decision trees, Naive Bayes classifier, k-nearest neighbor classifier, Perceptron, multilayer perceptron, Neural network, back-propagation Algorithm, Support Vector Machine, Kernel functions.

Unit IV Evaluation: Performance evaluation metrics, ROC Curves, Validation methods, Bias- variance decomposition, Model complexity.

Unit-V Unsupervised learning: Clustering, distance metrics, Mixture models, Expectation Maximization, Cluster validation methods.

Readings:

1. Alpaydin, Ethem, **Introduction to machine learning,** MIT press, 2014.

2. Christopher, M. Bishop, **Pattern Recognition And Machine Learning**, Springer-Verlag, 2016.

3. Shai Shalev-Shwartz, Shai Ben-David, **Understanding Machine Learning: From Theory to Algorithms,** Cambridge Press, 2014.

4. Michalski, Ryszard S., Jaime G. Carbonell, and Tom M. Mitchell, eds. **Machine learning: An artificial intelligence approach**, Springer Science & Business Media, 2013.

MCAE320: DIGITAL IMAGE PROCESSING [4-1-0]

Course Outcomes:

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

CO1: enhance the quality of an image using various transformations.

CO2: transform an image in spatial domain to frequency domain, and DWT domain.

CO3: apply required morphological operations to an image.

CO4: segment an image using various approaches.

Syllabus:

Unit-I Introduction: Applications of digital image processing, steps in digital image processing: image acquisition, image sampling and quantization, basic relationships between pixel.

Unit-II Image enhancement in the spatial domain and frequency domain: Gray level transformations, histogram processing, local enhancement, image subtraction, image averaging, spatial filtering: smoothing and sharpening filters, Discrete Fourier transformation, filtering in the frequency domain: smoothening and sharpening filters, image restoration in spatial and frequency domains.

Unit-III Morphological image processing: erosion and dilation, opening and closing, hit-or-miss transformation, some basic morphological algorithms.

Unit-IV Image segmentation: Point, line and edge detection, gradient operator, edge linking and boundary detection, thresholding, region-based segmentation, representation schemes like chain codes, polygonal approximations, boundary segments, skeleton of a region, boundary descriptor, regional descriptor.

Unit-V Introduction to Image Compression: Image compression models, Error free compression techniques, lossy compression techniques, JPEG, MPEG.

Readings:

1. Rafael C. Gonzalez and Richard E.Woods**, Digital Image Processing** (3rd edition), Prentice–Hall of India, 2016.

2. Bernd Jahne**, Digital Image Processing**, (6th edition), Springer, 2005.

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

4. M.A. Joshi, **Digital Image Processing: An Algorithmic Approach** (2nd edition), Prentice-Hall of India.

5. B. Chandra and D.D. Majumder, **Digital Image Processing and Analysis**, Prentice-Hall of India, 2011.

MCAE321: INTERNET OF THINGS [4-1-0]

Course Outcomes:

On completion of the course, the student is expected to:

CO1 : describe the basics of IoT and how it works.

CO2 : understand the key components that make up an IoT system

CO3 : recognise the factors that contributed to the emergence of IoT

CO4: use real IoT protocols for communication

Syllabus:

Unit I Basics of IoT: Introduction to Internet of things, characteristics, architecture, Various sensors and sensing techniques. Technological trends in IoT, Impact of IoT on society, Application scenarios, vision, scalability, current solutions, issues.

Unit II – Microcontroller and Interfacing Techniques for IoT Devices

Introduction to IoT and architecture layers, IoT smart devices, Typical embedded computing systems, Introduction to ARM architecture and programming method, Embedded system development: a case study, Introduction to interfacing techniques.

Unit III – IoT Protocols & Security

Networking and basic networking hardware. Networking protocols, Interaction between software and hardware in an IoT device. IoT components and technologies to secure systems and devices. Various security issues related to the IoT and security architectures. Hardware security threats and security vulnerabilities; protecting physical hardware

Unit IV – Location Tracking

Introduction to device localization and tracking; different types of localization techniques: time-of-arrival (TOA) based, time-difference-of-arrival (TDOA) based, angle-of-arrival (AOA) based, received signal strength (RSS) based, Radio-Frequency IDentification (RFID) based and fingerprinting based; Monte-Carlo tracking; Kalman filter based tracking; Cramer-Rao lower bound (CRLB) for device location estimator; Device diversity/heterogeneity issue in IoT networks.

Unit VI - IoT Applications: Application of IoT in Agriculture, Healthcare, Activity Monitoring.

Readings:

1. The Internet of Things: Enabling Technologies, Platforms, and Use Cases", by Pethuru Raj and Anupama C. Raman (CRC Press)

2. "Internet of Things: A Hands-on Approach", by Arshdeep Bahga and Vijay Madisetti (Universities Press)

MCAO301: JAVA PROGRAMMING [3-1-0]

Course Outcomes:

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

CO1: understand 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 Exception, 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 7 Edition**, Addison-Wesley, 2013**.**

2. Cay S. Horstmann, Core Java - Vol. I – Fundamentals, 10th Edition, Pearson, 2017**.**

3. Deitel & Deitel, **Java-How to Program** (9th ed.), Pearson Education, 2012.

4. Richard Johnson, An 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.

MCAO 302: DATA MINING FOR BUSINESS APPLICATIONS (4+1)

Course Outcomes

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

CO1: Visualize data from 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 parts of a linear model

Unit-V Classification: Logistic Regression, Decision Trees, Naïve Bayes, Voting, confusion matrix, interpreting results, select a model to guide decisions

Unit-VI Clustering: Clustering methods: centroid-based clustering, expectation maximisation, 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** (3nd ed.), Morgan Kaufmann, 2011.
4. Robert Layton, **Learning Data Mining with Python**, Second Edition.

MCAO 303: DATA SCIENCE [4-1-0]

Course Outcomes:

Unit-1: Data Scientist's Toolbox: Turning data into actionable knowledge, introduction to the tools that will be used in building data analysis software: version control, markdown, git, GitHub.
Unit-2 Getting and Cleaning Data: Obtaining data from the web, from APIs, from databases and from colleagues in various formats. basics of data cleaning and making data "tidy".
Unit-3: Exploratory Data Analysis: Essential exploratory techniques for summarizing data, applied before formal modeling commences, eliminating or sharpening potential hypotheses about the world that can be addressed by the data, common multivariate statistical techniques used to visualize high dimensional data.
Unit-4: Reproducible Research: Concepts and tools behind reporting modern data analysis in a reproducible manner.

Readings

1. Rachel Schutt, Cathy O'Neil, "Doing Data Science: Straight Talk from the Frontiline" by Schroff/O'Reilly, 2013.
2. Foster Provost, Tom Fawcett, "Data Science for Business" What You Need to Know About Data Mining and Data-Analytic Thinking" by O'Reilly, 2013.
3. John W. Foreman, "Data Smart: Using Data Science to Transform Information into Insight" by John Wiley & Sons, 2013.
4. Eric Seigel, "Predictive Analytics: The Power to Predict who Will Click, Buy, Lie, or Die", 1stEdition, by Wiley, 2013.