Learning Outcomes based Curriculum

Department of Computer Science and Engineering

Tezpur University

Master of Computer Application

Preamble

The Master of Computer Application (MCA) programme was started in July, 1994 with a view to produce high quality information system professionals. Till 2019, the programme was a full time three year (6 semesters) degree programme. From the academic session 2020-21, the All India Council for Technical Education (AICTE) has reduced the duration of MCA degree programme from the three years to current two years. Students having completed bachelor's degree of minimum three years duration in any discipline with at least 50% marks in Major/Honours subject or 55% marks in aggregate for no major/honours, or 55% marks in aggregate in BCA are admitted to the programme. Admission is based on the performance in the University entrance examinations.

1. Introduction

The MCA programme provides intensive theoretical and practical training in various aspects of Computer Science and Information System development. It was started with the intention to train the students in the topics in emerging fields of Computer Science and its applications. With new innovations constantly on the horizon, the demand of skilled computer science professional keeps rising in the industries. Our graduates of the MCA programme can fill up the demand. The curriculum of the MCA programme has been designed and evolved keeping this in mind and according to the model prescribed by the AICTE. Courses offered include Mathematical Foundation for Computer Science, Algorithms and Data Structures, Computer organization, Operating Systems, Computer Networks, Image Processing, Database System etc. The project works in the programme are intended to equip the students to go deeper into area of computer applications. The curriculum is organized with few core courses and many electives to give the students enough exposure to specialization like Soft Computing, Machine learning etc.

2. Qualification descriptors for the graduates

Knowledge & Understanding

1. Graduates develop an in-depth knowledge in the fundamentals of Computer Science and its applications.
2. Graduates develop the ability and confidence to analyse problem and design solution in complex computing environment using state of the art techniques, tools and resources.
3. Graduates acquire expert awareness and competency to supervise and moderate the computer science applications in various domains.

Skills & Techniques

1. Graduates develop skill set for using knowledge in computer science to create the ability to configure and operate complex software systems, packages, tools, and applications for sustainability in various domains.
2. Graduates have the right communication skills required for success in their profession.
3. Graduates equip themselves with techniques of design of experiments, analysis and interpretation of data and synthesis of information to provide a valid conclusion.

Competence

1. Graduates develop the competency to adapt to the changing trends of computer applications.
2. Graduates are ready to work individually as well as in teams, in industry, academia, research, and entrepreneurship.
3. Graduates are ready for pursuing lifelong learning to enhance the adaptability to the changing trends and career opportunities in computer applications.

3. Graduates Attributes

1. Graduate will have an in-depth technical knowledge in the field of computer application.
2. Graduate will have various computing skills like the analysis, design, and development of innovative software products to meet the industry needs

3. Graduate will pursue lifelong learning and to do research as computing professionals and scientists.
4. Graduate will communicate and function effectively in teams in multidisciplinary fields within the global, societal and environmental context.
5. Graduate will develop competency, creativity, and innovativeness in the field, with the ability to adapt to the changing trends and career opportunities in computer application.

4. Program Outcomes

1. Graduates will apply domain knowledge, abstraction and conceptualization of computing models from defined problems and requirements.
2. Graduates will be able to design experiments, analyse and interpret data, solve complex computing problems and synthesize the information to provide valid conclusions by creating, selecting, adapting and applying appropriate techniques, resources, and modern computing tools with an understanding of the limitations.
3. Graduates will be able to demonstrate knowledge and understanding of the computing and management principles and apply these to one's own work, to manage projects effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.
4. Graduates will be able to communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.
5. Graduates will be able to identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

5. Programme structure

Total Credits: 80

Structure of the curriculum

Course category	No of courses	Credits per course	Total Credits
I. Core courses	14	0/1/2/3/4	53

II.	Elective courses	7	3	21
	Open Elective courses	2	3	6
III.	Bridge courses	2	3	0
	Total credits			80

6. SEMESTER-WISE SCHEDULE

Semester	Course	Category of Course	Mode of Delivery & Credit			Total Credit
			L	T	P	C
First	**THEORY**					
	CS405: Discrete Mathematics	Core	2	1	0	3
	CS412: Data structures	Core	3	1	0	4
	EF103: Communicative English	Audit	2	0	1	0
	Elective-1	Elective				3
	Elective-2	Elective				3
	Elective-3	Elective				3
	Elective-1 - (any one of these courses)					
	MS509: Probability & Statistics	Elective	2	1	0	3
	CS204: Mathematical Foundation	Elective	3	0	0	3
	Elective-2 : (any one of these courses)					
	CS422: Numerical Methods	Elective	2	0	1	3
	CS305: Internet Concepts and Web Technology	Elective	2	1	0	3
	Elrctive-3 : (any one of these courses)					
	CS421: Graph Theory	Elective	3	0	0	3
	CS424: Formal Language & Automata	Elective	3	0	0	3
	Bridge Courses					
	CS309: Computing fundamentals	Bridge*	2	0	1	0
	CS410: Computer organization	Bridge*	2	1	0	0
	*Bridge Courses are noncredit courses					
	LABORATORY					
	CS312: OO programming & Data Structures Lab.	Core	0	0	2	2
					Total	**18**
Second	**THEORY**					
	IC361: Accounting & Financial Management	Core	2	1	0	3
	CS413: Database Management Systems	Core	3	0	0	3
	CS513: Software Engineering	Core	3	0	0	3

	Elective-4	Elective				3
	Elective-5	Elective				3
	LABORATORIES					
	CS414: Database Management Systems Lab.	Core	0	0	2	2
	CS518: Software Engineering Lab	Core	0	0	1	1
					Total	**18**
Third	**THEORY**					
	CS519: Computer Networks	Core	3	1	0	4
	CS516: Operating Systems	Core	2	1	0	3
	CS510: Minor Project	Core	0	0	5	5
	Elective -6	Elective				3
	Elective-7	Elective				3
	Open Elective-I	Open Elective				3
	LABORATORIES					
	CS520: Networks Lab	Core	0	1	1	2
	CS517: Operating system Lab	Core	0	1	1	2
					Total	**25**
Fourth	CS515 Major Project	Core				16
	Open Elective-2	Open Elective				3
					Total	**19**

List of Elective Courses

Course Code	**Title**	**Credit Structure**	**Total Credit**
CS204	Mathematical Foundation	3-0-0	3
CS305	Internet Concepts and Web Technology	2-1-0	3
CS309	Computing Fundamentals	2-0-1	3
IC361	Accounting & Financial Management	2-1-0	3
CS410	Computer Organization	2-1-0	3
CS421	Graph Theory	3-0-0	3
CS422	Numerical Methods	3-0-1	4
CS424	Formal Language and Automata	3-0-0	3
CS425	Database administration	3-0-1	4
IT504	E-Commerce	3-0-0	3
IT509	Data Mining & Data Warehousing	3-0-1	4
IT507	Computer Security & Cryptography	3-0-0	3
MS509	Probability & Statistics	2-1-0	4

CS522	Computer Graphics	3-0-1	4
CS523	Enterprise Resource Planning	3-0-0	3
CS524	Theory of Computation	3-0-0	3
CS525	Artificial Intelligence	3-0-0	3
CS541	Mathematical Foundation for Computer Science	3-1-0	4
CS530	Data Analytics and Visualization	3-0-1	4
CS531	Object Oriented Programming & Design	3-1-1	5
CS533	Computational Geometry	3-0-0	3
CS529	Embedded Systems	3-0-1	4
CS601	Design & Analysis of Algorithms	3-0-0	3
CS602	Image Processing	3-0-0	3
CS606	Computer Architecture and Parallel Processing	3-0-0	3
CS609	Geographic Information Systems	3-0-0	3
CS610	Bioinformatics	3-0-0	3
IT611	Distributed Systems	3-0-0	3
CS616	Machine Learning	3-0-0	3
CS621	Mobile Computing	4-0-0	4
CS625	Web Technology	3-0-1	4

7. Mapping of course with program outcomes (POs)

Course title	PO1	PO2	PO3	PO4	PO5
CS405: Discrete Mathematics	50%	25%	25%	-	-
CS412: Data structures	50%	25%	25%	-	-
EF103: Communicative English	-	-	-	70%	30%
CS312: OO programming & Data Structures Lab.	40%	20%	20%	-	20%
CS516**:** Operating Systems	40%	30%	30%	-	-
CS413**:** Database Management Systems	40%	20%	20%	-	20%
CS513: Software Engineering	20%	20%	20%	20%	20%

CS517: Operating system Lab	40%	30%	30%	-	-
CS414: Database Management Systems Lab.	40%	20%	20%	-	20%
CS518**:** Software Engineering Lab	20%	20%	20%	20%	20%
CS519: Computer Networks	40%	30%	10%	-	20%
IC361: Accounting & Financial Management	40%	20%	20%	-	20%
CS520**:** Networks Lab	40%	30%	10%	-	20%
CS510: Minor Project	20%	20%	20%	20%	20%
CS515: Major Project	20%	20%	20%	20%	20%
Elective-1	50%	20%	-	-	30%
Elective-2	40%	20%	20%	-	20%
Elective-3	40%	20%	20%	-	20%
Elective-4	40%	20%	20%	-	20%
Elective-5	40%	20%	20%	-	20%
Elective-6	40%	25%	10%	-	25%
Elective-7	20%	20%	20%	20%	20%

8. **Evaluation plan:**
   There shall be minimum two Sessional Tests and two Examinations for each Theory Course, and two Examinations for a Practical Course having L-T-P structure. Details as follows:

Evaluation plan for Theory Courses:

Sessional Test/ Examination		Course Credit ≤ 2		Course Credit ≥ 3		Semester period
Nomenclature	Type	Marks	Duration	Marks	Duration	
Sessional Test-I	Written	20	30 min	25	45 min	Within 5th week
Mid-Semester Examination	Written	30	90 min	40	2 hours	Within 10th week

Sessional Test-II	Written/Assignment/Seminar etc.	20	XX	25	XX	Within 14th week
End-Semester Examination	Written	50	2 hours	60	3 hours	Within 18th week

Evaluation plan for Practical Courses:

Examination		L-T-P Structure-wise Marks		Semester period
		L-T-P: 0-0-z	L-T-P: x-y-z	
Nomenclature	Type	Marks	Marks	
Mid-Semester (Practical) Examination	Viva, Report	30	-	Before Mid Semester
End-Semester (Practical) Examination	Practical Examination, Viva, Report	70	50	Before End Semester

9. DETAILED SYLLABUS

CS309: Computing Fundamentals

Credit: 3: (2L-0T-1P)

Course Content

Unit 1: Introduction to Computing Systems Hardware:

Basic parts and structure of computer system, Categorizing computers- Desk-top Workstations/ PCs, Mainframe, Super-Computer, Parallel computer, Generation of computers; Client-Server Architecture. Information processing life cycle, Essential computer hardware: Central processing Unit, Input/ output devices, Computer Memory: primary and secondary memory, magnetic and optical storage devices.

Data processing: concepts of data processing, Definition of Information and data, Basic data types, Storage of data/Information as files, Representation of data/Information.

Unit 2: Introduction to Computing Systems Software:

Essential computer software: System and Application. Operating Systems: Single user/ Multiuser, Batch/ Interactive, Real-time, Network OS, File server, Distributed OS; Command Shell, GUI-Windows; Basics of DOS, LINUX/UNIX, Windows operating systems.

Unit 3: Fundamentals of problem solving and Programming:

Algorithm: Concept and Definition, its representations-pseudo code, flowchart and computer program. Example algorithms: string processing, root finding, matrix operations, record processing, searching, sorting etc.

Computer Programming: Programming styles, Program development steps.

Program development Environment: Generation of Languages - Machine, Assembly, HLL, Programming tools - Assembler, Compiler, Interpreter, Linker and Loader, Text Editor,

Features of a high level language : Assignment statement, input-output statements; Expressions; data types; conditional statements, Iterative statements; Array data type and use of arrays; character data type and text processing; functional and procedural abstraction; Recursion; Pointer data type and simple applications of pointers.

Unit 4: Computer & Communication:

Computer Networks- LAN, WAN, Communication services across network protocols, e-mail, Internet facilities through World Wide Web; Communication devices.

Unit 5: Security and Integrity Issues:

Information integrity definition, Ensuring integrity; Computer and communication security, Perverse software, Concepts and component of security, Preventive measures, and treatment.

Unit 6: Laboratory:

Familiarization MS DOS and MS WINDOWS and LINUX/UNIX commands. File creation, editing and directory creation, Mastery of MS DOS commands, learning to use MS office: MS WORD, use of database and spread sheet. Slide creation with PowerPoint

Textbooks Recommended:

1. Trainer T., et al, “Computers” (4th edition) McGraw Hill, 1994 Rajaraman V, “Fundamental of Computers” (2nd edition), Prentice Hall of India, New Delhi. 1996.
2. Dromey, G, “How to solve it by computer” PHI (EEE), 1985.
3. Karnighan and Ritchie, “The C Programming Language” (2nd edition), Pearson

CS204: Mathematical Foundation

Credit: 3 (3L-0T-0P)

<u>Course Content</u>

Unit 1: Matrices:

Matrix and Determinants, Rank of a matrix, Solutions of simultaneous linear equations, existence of solutions and solution by Gaussian elimination, Eigen values and Eigenvectors- diagonalization, computing A^k, matrix exponential for solving difference and differential equation, Hermitian and skew Hermitian matrices, symmetric and positive definite matrix

Unit 2: Vector Space:

Fields (definition with a few examples), definition and examples of vector spaces, Properties of linearly independent and dependent set of vectors, Basis, and dimension of a vector space,

Unit 3: Vector Calculus:

Multi Variable Functions, Continuity, Partial derivatives, Total derivative, Increment theorem, Chain rule, Gradient, Directional derivatives, Back propagation and automatic differentiation, higher order derivatives, maxima, minima, Lagrange multipliers. linearization and Taylor series.

Textbooks Recommended:

1. Krishnamurthy V., Mainra V. P., Arora J. L., "An Introduction to Linear Algebra", EWP
2. M.K. Venkataraman, NPC, "Engineering Mathematics Vol.1 & Vol.2."
3. Grewal B. S., "Engineering Mathematics"

CS305: Internet Concepts and Web Technology
Credit: 3 (2L-1T-0P)

Course Content

Unit 1: World Wide Web and The Internet:

Classification of Networks, Networking models, a brief history of TCP/IP and the Internet, Internet Infrastructures, Internet addressing – structure of internet server address and address space. Protocols and Services on Internet – Domain Name System (DNS), SMTP and Mails, HTTP and WWW, UseNet and Newsgroups, FTP, Telnet; Web page, Web browsers, Introduction to web servers and their architectures Web Security: S-HTTP, Fire Walls, Proxy Servers.

Unit 2: Web Page and basic HTML:

Static and dynamic Web pages, Basic HTML Tags Formatting of Text – headers, formatting tags, PRE and FONT tag, Special characters, Meta Tag and working with images.

Unit 3: Advanced HTML:

Hyperlinks, List – Unordered and Ordered, Definition lists, Tables in HTML, Frames in HTML, Forms in HTML, image maps, style sheets in HTML. DHTML, Introduction to XML, syntax, DTD

Unit 4: Introduction to JavaScript:

Client-side Scripting languages, Basics of JavaScript and Creating interactive documents using JavaScript

Unit 5: Practical Web Design:

Introduction to web design using an Integrated Development Environment like Dreamweaver.

Textbooks Recommended:

1. Margaret Levine Young, "Internet - The Complete Reference", (Millennium Edition), TMT Edition -1999.
2. Bayross, "Web Enable Commercial Application Development Using HTML, DHTML, JavaScript, Perl, CGI", BPB publications, 2000.
3. J. Jawoskri, "Mastering JavaScript" BPB publications, 1999.

CS410: Computer organization

Credit: 3 (3L-0T-0P)

Course Content

Unit-1: Digital Logic Design:

Basic logic gates and combinational circuits, Half Adder, Full adder, Decoder, Encoder, Multiplexer, Demultiplexer, Sequential Circuits, State transistors, Flip-flops, Registers, Counters

Unit-2: Basic Organization of Computer:

Block level organization of the functional units, Basic Memory Model and Interfacing, Programmability concepts and program execution model, Instruction execution cycles.

Unit-3: Memory Organization:

RAM, ROM, Types and Organization, Interfacing of memory with processor, Memory hierarchy, Cache memory, Virtual Memory.

Secondary Storage: Magnetic disk drives, Virtual memory mechanisms, Optical Disks, Magnetic tapes, Solid State Drives.

Unit-4: Instruction Set Architecture:

Instruction set- types and format, instruction cycles, registers and storage, addressing modes

Unit-5: Processing Unit:

Computer arithmetic and their implementation – Integer representation, Floating point number representations.

ALU, Data path organization, Control unit design – Hardwired control, Microprogramming

Unit-6: Input Output Organization:

I/O organization and mapping; Data Transfer Methods- Program controlled, Interrupt driven, DMA. Interfacing with I/O devices and data transfer protocols – Asynchronous, Synchronous, keyboard and display interfaces.

Textbook Recommended:

1. Stallings W., "Computer Organization and Architecture", Pearson (2010)
2. Hamacher, Vranesic, and Zaky, "Computer Organization", McGraw Hill Education (India) Private Limited; 5 edition (4 November 2011)
3. Mano M.M., "Digital Logic Design", Pearson; Third edition (2008)

CS 312: OO programming & Data Structures Lab.

Co-requisite: CS412 Data Structures

Credit: 3(0L-1T-1P)

<u>**Course Content**</u>

Unit-1: Introduction to Object-oriented programming using C++:

Object, Class, Encapsulation, Abstraction, Inheritance, Polymorphism, Inheritance, Operator overloading, Generic programming concept using template

Unit-2: Programming Laboratory using Data Structures:

Programming assignments on the implementations of data structures – list, stack, queues, trees, sets, graph and their applications, Implementations of various sorting and searching techniques.

Textbooks Recommended:

1. Satraj Sahni, “Data Structures, Algorithms and Applications in C++ (2nd Edition)”, University Press.
2. Lipman, “C++ PrimerPai G A V, “Data Structures and Algorithms: Concepts, Techniques and Applications”, 2ndEdn, Tata McGraw-Hill, 2008.

CS412: Data structures

Co-requisite: CS312 OO programming & Data Structures Lab.

Credit: 4 (3L-1L-0P)

<u>Course Content</u>

Unit-1: Basic Concepts;

Data Structures, Algorithms, Complexity of algorithms. Asymptotic Analysis.

Unit-2: Linear Data Structures:

Abstract Data Types (ADT), Stack, Queues, Deques, Applications of stack and Queues, Lists: Singly Linked List, Circularly Linked Lists, Doubly Linked Lists

Unit-3: Non-linear Data Structures:

Trees: Definition and Implementation; Binary trees, Tree traversal, Postfix, Prefix notations. Sets: Implementation; Dictionary, Hash table, Priority queues; Advanced Set Representation Methods - Binary search tree, AVL tree, Balanced tree, Sets with Merge and Find operation.

Unit-4: Graphs:

Directed graphs: Representation; Single source shortest path problem, All pair shortest path problem, Transitive closure. Undirected graph: Minimum spanning tree

Unit-4: Advanced Structures:

B-trees and its variants, k-d trees, quad trees, R-tree and its variants

Textbooks Recommended:

1. Horowitz E., Sahni S., Anderson-Freed S., “Fundamentals of Data Structures in C”, Universities Press; Second edition (2008), Books/References:
2. Aho, Hopcroft and Ullman, “Data Structures and Algorithms”, Addison Wesley; 1st edition (1 January 1983),
3. Horowitz and Sahni, “Fundamentals of Algorithms”, Universities Press; Second edition (2008)
4. Knuth. D., “The Art of Computer Programming, Vol.-I & II”, Dorling Kindersley Pvt Ltd; 3rd edition edition (1 December 2005).

CS320 Probability and Statistics

Credit: 3 (2P-1T-0P)

<u>Course Content</u>

Unit-I: Concepts of Probability:

Experiment and Sample Space, Events, Probability of an Event, Basic Probability Rules, Joint and Marginal probabilities, Conditional Probability, Independent events, Bayes’ Theorem

Unit-2: Random Variables and probability distributions:

Random variable-continuous and discrete**,** Mean, Variance and Standard Deviation of a Random Variable. Expected Value and Variance, moments Distributions – Binomial, Poisson, Normal etc. The Normal Theorem and the Central Limit Theorem.

Unit-3: Sampling:

Purpose and the nature of sampling, its uses and applications, Random sampling - basic method, stratified sampling and variants thereof, cluster sampling, Non-random sampling: purposive methods, sequential sampling, Multivariate distributions, independent random variables

Unit-4: Estimation and Hypothesis Testing:

Estimation of Population Parameters: Parameter and Statistic, Point and Interval Estimation, Interval Estimation of Three Common Parameters. Hypothesis Testing for a Single Population: Concept of a Hypothesis, Tests Involving a Population Mean, Tests Involving a Population Proportion, Tests Involving a Population Standard Deviation. Analysis of variance, one-way ANOVA and two-way ANOVA with fixed effect; interval estimation

Unit5: Correlation and regression:

Correlation and regression, Correlation Coefficient, Approaches to correlation: the linear model approach, the least squares fitting approach, strengths and weaknesses of these and conditions for applicability.

Unit-6: Stochastic processes:

Bernoulli and Poisson processes, Discrete parameter Markov chains: transition probabilities, limiting distributions, Finite Markov chains, program execution times, Queuing: M/M1 and M/G/1, birth and death process

Text Books:

1. R. M. Bethea, B. S. Duran, T. L. Boullion, "Statistical Methods for Engineers and Scientists," Marcell Dekker Inc.
2. H. Frank, S. C. Altheon, "Statistics: Concepts and Applications", Cambridge Low Priced Edition.
3. M. R. Spiegel, "Theory and Problems of Probability and Statistics", Schaum's Outline Series, McGraw Hill.

CS405 Discrete Mathematics
Credit: 3 (2L-1T-0P)

Course Content:

Unit1: Set. relation and functions:

- Set, relations, equivalence relations; mappings-one-one and on to ;
- Definition of an algebraic structure;
- Introduction to groups, subgroups, normal subgroups, isomorphism, homeomorphism; automorphism of groups; semigroups, monoids, rings, vector space.

Unit2: Logic:

- Logic operators, Truth table, Normal forms
- Theory of inference and deduction.
- Mathematical induction.
- Predicate calculus; predicates and quantifiers.

- Boolean algebra.
- Lattice.

Unit3: Combinatorics :
- Basic counting techniques.
- Recurrence relations and their solutions.
- Generating functions.

Unit4: Modular Arithmetic :
- Congruence modulo, Fermat s Theorem, Euler s Theorem, Multiplicative Inverse, Reminder Theorem, FFT, Discrete Logarithm.

Text Book:

1. Kenneth H. Rosen : Discrete Mathematics and Its Applications, Mcgraw-Hill College; 6th edition (January 5, 2006)

References Books:

1. Liu, C. L. : Introduction to Discrete Mathematics. McGraw Hill Education (India) Private Limited (2008)
2. Trembley, Manohar : Discrete Mathematical Structures. McGraw Hill Education (India) Private Limited (2 February 2001)
3. L. Lovász, J. Pelikán , K. Vesztergombi : Discrete Mathematics: Elementary and Beyond (Undergraduate Texts in Mathematics), Springer; 2003 edition (17 February 2003)
4. Jiri Matousek, Invitation to Discrete Mathematics, Clarendon Press (23 July 1998)

CS422 Numerical Methods
Credit: 4 (3L-0T-1P)

<u>Course Content:</u>

Basic Concepts: Limit, continuity, derivative, convergence, difference equations; Absolute error, relative error, round off errors, numerical stability, conditioning;

Solutions of nonlinear and linear equations- Bisection method, Newtons method, Secant method, Gaussian method, Gauss Seidel, Steepest descent, Conjugate gradient.

Matrix eigen value problem, matrix factorization and least square problem. Interpolation and extrapolation.

Numerical differentiation and integration; Numerical solutions of ordinary and partial differential equations.

Text Book:

1. R. L. Burden and J. D. Faires: Numerical Analysis, Cengage Learning India; 09 edition 2012

Reference Books:

1. David Kincaid & Ward Cheney : Numerical Analysis, American Mathematical Society; Third edition (2010)
2. J. Stoer and R. Bulirsch : Introduction To Numerical Analysis , Springer (sie) (2009)

CS421 Graph Theory
Credit: 3 (3L-0T-0P)

Course Content:
Graph : Incidence and degree; Handshaking Lemma; Isomorphism; Subgraphs and Union of graphs; Connenctedness; Walks, Paths and Circuits; Components and Connectednes; Walks, Walks, Paths and Circuits; Components and Connectednes algorithms; Shortest Path Algorithms, Eulerian graph, Fleury s algorithm and Chinese postman problem; Hamiltonian graph - necessary and sufficient conditions; Traveling salesman; Bipartite graph.

Tree : Properties of trees; Pedant vertices in a tree; Center of a tree; Rooted binary trees; Spanning trees - Spanning tree algorithms; Fundamental circuits; Spanning trees of a weighted graph; cut-sets and cut-vertices; Fundamental cut-sets; Connectivity and separativity; network flow; max-flow min-cut theorem.

Planner graph : Combinatorial and geometric dual; Kuratowski s graph; detection of planarity; Thickness and crossings.
Matrix representations of graph: Incidence; Adjacency; matrices and their properties.

Colourings: Chromatic number : Chromatic polynomial; The six and five colour theorems; The four colour problem.
Directed graphs: Binary relations; Directed graphs and connectedness; directed trees;

Aborecence; Polish method; Tournaments.
Counting of labeled trees: Cayley s theorem; Counting methods; Polya theory. Application of graphs in computer science.

Text Book:
1. West Douglas B., Introduction to Graph Theory, PHI; 2nd Edition (2002)

Reference Books:
1. Deo, N.: Graph Theory with Applications to Engineering and Computer Science.
2. Harary : Graph Theory, PHI (EEE)
3. Adrian Bondy, U.S.R. Murty, Graph Theory (Graduate Texts in Mathematics), Springer; 27 September 2011
4. Reinhard Diestel, Graph Theory, Springer (India) Pvt. Ltd. (2008)

CS424 Formal Language and Automata
Credit: 3 (3L-0T-0P)

Course Content:
Alphabets, Languages, Grammars.
Finite automata: regular expressions, regular languages.
Context free languages: pushdown automata, DCFLs, LL(k) and LALR grammars. Context sensitive languages: linear bound automata.
Turing machines: recursively enumerable languages. Operations on formal languages and their properties. Decision questions on languages, Undecidable problems.

Text Books:

1. Michael Sipser, Introduction to Theory of Computation, Cengage; 03 edition (1 October 2014)

Reference Books:

1. Hopcroft J. E., Ullman J. D., Automata Theory Language & Computation, Pearson; Third edition (2008)
2. Martin J. C., Introduction to Languages and the Theory of Computation, McGraw-Hill 3rd edition (6 June 2007)
3. Buchi A., Finite Automata, Their Algebras and Grammars: Towards a Theory of Formal Expressions, Springer; 1989 edition (12 October 1989)

IC:361Accounting And Financial Management
Credit 3 (2L-1T-0P)

Course Content:

Unit 1: Introduction of Accounting
Meaning and Scope of accounting; Objectives, nature and functions of accounting; Advantages and limitations of accounting; Accounting as a measurement and valuation principle; Accounting Principles; Accounting as an Information System; Basis of Accounting – Cash and accrual system of Accounting; Branches of accounting; Accounting and management control.

Unit 2: Basic Accounting Process
Accounting process from recording of transactions till preparation of Trial Balance-Concept of assets, liabilities, capital, income and expenses; Balance Sheet equation; Classification of receipts/income and payments/expenditure into capital and revenue; Rules for Debit and credit; recording of transactions; The Journal and subsidiary books, ledger accounts- posting of transactions; Adjusting entry; Bank Reconciliation Statement.

Unit 3: Trial Balance and Final Accounts
Trial Balance – meaning and importance, adjusted trial balance, Difference in Trial Balance; Errorsand rectification entries thereof. Need for measurement of income, Realization principle vs. Accrual principle; accounting period, Matching revenue and expenses. Manufacturing Account, Trading Account, Concept of Gross profit and Net profit,, Need and meaning of Profit and Loss Account, Forms and contents of Profit and Loss Account, Concept of Balance Sheet, Classification of items in a balance sheet; Format of Company Balance Sheet, Preparation of Final Accounts; Cash Flow statement. Accounting for depreciation; method of inventory valuation35

Unit 4: Accounting Standards and emerging concepts in Accounting
Introduction to Accounting Standards and IFRS converged Ind AS, Human Resource Accounting, Corporate Social Accounting etc.; computerized Accounting System and accounting software,

Unit 5: Study of Annual Reports of Companies; Analysis, interpretation and Judgment building
(Assignment based)

Text Books:

1. Ramachandran, N. and Kakani, R.K. Financial Accounting for Management. 3/e, TATAMcGraw-Hill Education Pvt. Ltd: Noida, 2011.
2. BhattacharjeeAshis K. Financial Accounting for Business Management. Prentice Hall India:New Delhi, 2006.

Reference Books:
1. Anthony Robert N., Hawkins David, Merchant Kenneth A. ,Financial Accounting-Text andCases, McGraw-Hill Higher Education; 13 edition (1 June 2010

CS413 Database Management Systems
Credit: 3 (3L-0T-0P)

Course Content:

*Overview:*Concept of database, data independence, redundancy Control; Database architecture - ANSI model.
Modeling of real world situation: Entity-relationship model; Data models: Network, Hierarchical, Ralational.
Relational data model: DDL, DML: relational algebra and calculus; functional dependencies, normal forms, decomposition, integrity rules; Query languages for relational systems: SQL, QBE, query optimization, embedded SQL.
Database transactions, concurrency control, recovery and security issues in databases.
Brief treatment of: Client-server models, distributed databases, object-oriented databases, deductive databases, multimedia databases, active databases.

Text Book:
1. Elmasri and Navathe, Fundamentals of database systems; Pearson (1 May 2008)

Reference Books:
1. Silberschatz and Korth, Database system concepts, Mc Graw Hill (1 January 2011)
2. C. J. Date, A. Kannan, S. Swamynathan: An Introduction to Database Systems, Pearson Education; eighth edition (1 January 2012)

CS513 Software Engineering
Credit: 4 (3L-0T-0P)

Course Content:

Introduction to software engineering, concept of a software project, size factor, quality and productivity factor, different phase of a software development life cycle, managerial issues.

Software project planning: Problem definition, development of a solution strategy, development process planning, software development models and their comparative study; Organizational structure planning, project formats and team structures; Planning for quality assurance and configuration management; Planning for verification and validation.
Software economics: Cost estimation and evaluation techniques, cost estimation based on COCOMO model and Raleigh model.
Software requirements analysis and specifications techniques- their notations & languages . Software design: Concept of fundamental design; Design approaches- top-down & bottom-up, structured, object-based & object oriented design; Design specification and notations.
Software implementation: Structured coding techniques, coding styles, and standards; Guidelines for coding and documentation.

Software verification and validation: Theoretical foundation, black box and white box approaches; Integration and system testing.
Software reliability: Definition and concept of reliability, software faults, errors, repair and availability, reliability and availability models.
Case studies.

Text book:
1. Pressman, R.S., Software Engineering: A Practitioner's Approach, McGraw Hill.

References:

1. Rajib Mall**:** Fundamentals of Software Engineering, Prentice Hall India Learning Private Limited; Fourth edition (2 April 2014)
2. Ian Sommerville: Software Engineering, Pearson Education; Nineth edition (2013)
3. Fairley, R.E., Software Engineering Concepts, McGraw Hill Education (India) Private Limited (23 April 2001)

CS516 Operating Systems

Credits: 3 (2L-1T-0P)

Abstract: The course Operating Systems intends to build the basic concepts of the operating system software and its implementation details for computer system. It also covers topics with case studies giving the students opportunity to explore practical operating systems like - MS Windows, Unix, Linux. The course will be instrumental to build the confidence in the students to develop the introductory knowledge of design and develop an operating system.

Course Outcomes:
Towards the end of the course the student would be conversant with

1. Operating systems and OS design issues
2. OS based programming fundamentals
3. be familiar with OS system software design concepts
4. be able to design and develop OS features
5. OS based application development

Course Contents:
Overview: Evolution of Operating Systems, current status and future trends. Structural overview, system calls, functions of OS, Hardware requirements: protection, context switching, privileged mode
Concept of a process: states, operations with examples from UNIX/Linux (fork, exec) and/or Windows. Process scheduling, interprocess communication (shared memory and message passing), UNIX/Linux signals, cooperating and concurrent processes, tools, and constructs for concurrency, Threads: thread management, multithreaded model, scheduler activations, examples of threaded programs and applications.

Scheduling: multi-programming and time sharing, scheduling algorithms, multiprocessor scheduling, thread scheduling (examples using POSIX threads).

Process synchronization: mutual exclusion, shared data, critical sections, classical two process and n-process solutions, hardware primitives for synchronization, lock, semaphores, monitors, block and wakeup, classical problems in synchronization (producer-consumer, readers-writer, dining philosophers, etc.).

Deadlocks: modeling, characterization, prevention and avoidance, detection, and recovery.

Memory management: with and without swapping, MMU, Contiguous and non-contiguous allocation, paging and segmentation, demand paging, virtual memory, page replacement algorithms, working set model, thrashing, and implementations from operating systems such as UNIX, Windows. Current Hardware support for paging: e.g., Pentium/ MIPS processor etc.

Secondary storage and Input/Output: device controllers and device drivers, disks, scheduling algorithms, file systems, directory structure, device controllers and device drivers, disks, disk space management, disk scheduling, NFS, RAID, other devices and operations on them, UNIX FS, UFS protection and security.

Virtualization: Virtual Machine (VM), concept of hypervisor and virtual machine manager (VMM), types of hypervisor: kernel-based and hosted hypervisor, open source virtual machine design in Linux: Kernel-based Virtual Machine (KVM).

Protection and security: Illustrations of security model of UNIX and other OSs. Examples of attacks.

Pointers to advanced topics (distributed OS, multimedia OS, embedded OS, real-time OS, OS for multiprocessor machines, mobile OS, cluster OS).
Case study: Design of UNIX, Linux, Windows, Android

Textbooks:

1. Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 9th Ed., John Wiley, 2018.
2. Operating Systems: Internals and Design Principles, William Stallings, Pearson, 9th Ed., 2019.

Reference Books:

3. Operating systems: Concepts and Design, M. Milenkovic (McGraw Hill, 2001)
4. Modern Operating Systems, A. S. Tanenbaum (Pearson, 2009)
5. Design of the Unix Operating System, M. J. Bach (Prentice Hall of India, 1986)
6. Operating Systems: Three Easy Pieces, Remzi and Andrea Arpaci-Dusseau http://pages.cs.wisc.edu/~remzi/OSTEP/
6. Understanding Linux Kernel 2.6, Bovet and Chesti (Orelly), 2005)
7. Professional Linux Kernel Architecture, Wolfgang Mauerer, (Wiley)
8. Linux Kernel Development, R. Love (Addison Wesely, 2010)
9. Professional Android Application Development, R. Meier (John Wiley & Sons)

CS517 Operating Systems Lab
Credits: 1(0L-0T-1P)

Abstract: The course Operating System Lab intends to build hand on understanding about some of the important topics covered in Operating System course. It will familiarize with UNIX system calls for process management and inter-process communication, process synchronization, memory management and file system management; experiments on process scheduling and other operating system tasks through programming, simulation / implementation under a simulated environment.

Course Outcomes:
After completion of course, students would be able

1. to grasp knowledge for implementation of operating system software features,

2. to understand the working of an operating system kernel,
3. to master on using system level programming especially the kernel coding using C and java /C++ languages,
4. to master on develop algorithm for operating system design
5. to grasp knowledge for implementing file system concepts
6. to master in using simulation tools, system utilities

Course Contents:

Shell scripting primer using Bourne shell, Bash scripting for beginner, use of awk etc.
Create process (use of fork(), exec() etc. system calls), implement a process ownselves
Use system calls signal(), kill(), creating POSIX threads, using thread library Pthread library using system calls pthread_create() and pthread_exit()
Implementation of file locks using fcntl for basic file access synchronization
Use of basic IPC mechanism with pipe(), mknod(), using message queue, shared memory.
Learn to use synchronization of processes with semaphore and other tools,
Dynamic memory allocation, LKM programming, Device driver for char and block devices
Open source Linux kernel source code browsing and understanding.
Android based application development.
Open source hypervisor development

Reference Books:

1. Unix Network Programming, Stevens, Vol-1, Addison-Wesley, 3rd Ed, 2003 & Vol-2, Prentice Hall, 2nd ed, 1998
2. Design of Unix Operating System, M. J. Bach (Prentice Hall of India, 1986)
3. Linux Device Drivers, J. Corbet and A. Rubini, Orelly, 2005
4. Professional Android Application Development, R. Meier (John Wiley & Sons, 2014)
5. Writing a simple Operating System from Scratch, N. Blundell, University of Birmingham, UK, 2010
6. http://www.ee.surrey.ac.uk/Teaching/Unix/
7. http://developer.android.com/guide/index.html
8. Operating Systems: Three Easy Pieces, Remzi and Andrea Arpaci-Dusseau http://pages.cs.wisc.edu/~remzi/OSTEP/
9. Web links and tutorial materials will be provided time to time

CS413 Database Management Systems
Credits: 3(3L-0T-0P)

Abstract: The course Database System is an introductory course on database systems. The course covers the basic concepts of database, data models, database architecture, relational database languages, SQL, functional dependency and normalization, database transactions

Course Outcomes:
After completion of this course:

1. The students should clearly understand the concepts of database systems, their advantages, and applications.
2. The students should be able to design a database system for a given database problem.
3. The students should be familiar with the emerging database application areas.

Course Contents:

Introduction & Overview: Concept of database, Characteristics of database, Advantages, data independence, redundancy Control; Database architecture - ANSI model.
Modelling of real-world situation (data models): ER model, EER model
Relational data model: relational model concepts, relational algebra and calculus, SQL, ER/EER to relational model mapping,
Functional dependencies and normalization: functional dependencies, normal forms, decomposition, multi-valued functional dependency, and higher normal forms
Database Indexing and hashing: B-Tree, B+ Tree, static and dynamic hashing
Database Transaction concepts, query evaluation overview, security, and recovery
Distributed Database
Brief introduction to emerging database applications (like Hadoop, NoSQL etc.)

Text Books:

1. Fundamentals of Database Systems, Sixth(2011)/Seventh(2017)Edition, ELMASRI and NAVATHE, Pearson
2. Database Systems Concepts, Sixth (2010)/Seventh(2019) Edition, A. SILBERSCHATZ, H. F. KORTH, S SUDARSHAN, McGraw Hill,

References:

1. Database Management Systems, 3rd Edition, - Raghu Ramakrishnan, Johannes Gehrke, McGraw Hill, 2014
2. An Introduction to Database Systems, 8th Edition, C.J Date, Pearson, 2003
3. Fundamentals of Database Systems, by Leon & Leon, Tata McGraw Hil, 2008
4. SQL & NoSQL Databases, Meier, Andreas and Kaufmann, Michael, eBook, Springer 2019
5. Getting Started with NoSQL, Gaurav Vaish, Packt Publishing, March 2013
6. Hadoop: The Definitive Guide, 4th Edition, Tom White, O'Reilly Media, Inc., 2015

CS414 Database Management Systems Lab.

Credits: 2(0L – 1T – 1P)

Course Outcomes:

1. After completion of this course the students will be able to:
2. Create and manage a database using RDBMS like Oracle.
3. Perform queries on the database.
4. Build user-defined functions and procedures using PL/SQL.
5. Develop database applications.

Course Contents:

Introduction: Introduction to a RDBMS like Oracle
SQL: data types, DDL, DML
SQL functions, PL/SQL and user-defined function, triggers
DBA commands
Role-based authorization
Development of database applications using tools/languages like Forms & Reports, PHP, Java, JavaScript etc.

Laboratory works / experiments:

Basic commands of SQL and SQL Plus
Creating and modifying database tables, inserting, deleting and updating data in database tables using SQL
SQL commands for retrieving data from database tables
Creating and updating database triggers
Writing and executing SQL scripts
SQL scripts for building simple reports
Basics of PL/SQL
User-defined functions and procedures
DBA commands and database authorization
Developing GUIs using Oracle Forms & reports/PHP/Java etc.
Developing reports using Oracle Forms & reports/PHP/Java etc.

Text Books:

1. SQL The Complete Reference, 3rd Edition, James Groff, Paul Weinberg, McGraw Hill Education, 2017
2. Oracle 12c: The Complete Reference, Kevin Loney, George Koch McGraw Hill/Osborne, 2017
3. Learning PHP, MySQL & JavaScript 5e (Learning PHP, MYSQL, Javascript, CSS & HTML5), Robin Nixon, O'Reilly, 2018

Reference Books:

1. Beginning PHP and Oracle, W. J. Gilmore and B. Bryla, Apress, Berkley, CA, USA, 2007
2. The Underground PHP and Oracle Manual, Release 2 Christopher Jones and Alison Holloway, Oracle, 2012
3. Mastering Oracle SQL, 2nd Edition Sanjay Mishra and Alan Beaulieu, O'Reilly Media, 2004
4. Oracle PL/SQL Programming, 5/6th Edition, By Steven Feuerstein, Bill Pribyl, O'Reilly Media, 2016
5. Oracle Forms developer's handbook, Albert Lulushi, Pearson Education, 2012
6. Oracle 9i development by example, Dan Hotka, Pearson Education, 2001

CS 519 Computer Networks
Credit: 4(3L-1T-0P)

<u>Course Content:</u>

Review of Computer Network Architecture and the Subnet layers.
Data Transport: Connection management, Quality of Service, TCP/IP Protocol, ATM.
Session Management: Session establishment and maintenance, Dialogue management,
Recovery. End-to-end Data: Presentation formatting issues and methods: XDR, ASN.1, NDR; Data Compression, Lossless Compression Algorithms- Run length encoding, DPCM, Dictionary- based methods, Image compression- JPEG, Video compression- MPEG; Security and authentication techniques, Encryption algorithms.

Applications: E-mail, Remote login, File transfer, Network file system, Network management. UNIX network programming with TCP/IP; Network File System, Novell Netware, and Windows NT installation, configuration and use.

Text Book:

1. Behrouz A Forouzan, DeAnza CollegeFirouz Mosharraf: Computer Networks: A Top-Down Approach, McGraw Hill Education (India) Private Limited (11 November 2011)

References:

1. Tanenbaum A.S., David J. Wetherall : Computer Network, Pearson; Pearson; 5 edition (17 January 2012)
2. Stalling W.: Data and Computer Communication, Pearson; Nineth edition (2013)
3. Peterson L L, Davie B S, Computer Networks: A Systems Approach, Morgan Kaufmann Publishers In; 5th Revised edition edition (20 April 2011)
4. Stevens, UNIX Network Programming, Pearson Education; 1ST edition (2003)
5. Comer D E., Internetworking With TCP/IP Principles, Protocols, And Architecture, PHI (2013)

CS520 Networks Lab
Credits: 1(0L-0T-1P)

Course Outcomes:
At the end of this course the student should be able to:

1. Apply packet sniffing tools to capture packets and analyse
2. Write client-server applications in C/C++/Java
3. Use different OS tools to configure network and network protocols
4. Set up LAN using networking devices
5. Use network simulators to study behaviour of different protocols
6. Analyse performance of various communication protocols using packet sniffers

Course Contents:
Experimental study of application protocols such as HTTP, FTP, SMTP, using network packet sniffers and analyzers such as Ethereal, tcpdump, Wireshark etc.
Client-server based application development using socket programming in C/C++/Java (both TCP and UDP sockets)
Experiments with packet sniffers to study the behaviour of TCP protocol.
Using OS tools (netstat etc.) to understand TCP protocol (connection establishment, connection termination, retransmission timer behaviour, congestion control behaviour)
Setting up a small IP network - configuring interfaces, IP addresses and routing protocols to set up a small IP network
Introduction to network simulator tools (NS-2/3, GNS3, Mininet, QualNet etc.) to study behaviour of MAC (IEEE 802.3, IEEE 802.11) and other protocols.

Books:

1. WR Stevens, UNIX Network Programming, 2nd Ed., 2015, Pearson
2. Kirch and Dawson, Linux Network Administrator's Guide, O'Reilly

References:

1. Online and weblink reference manuals for network simulators

***