Syllabi of MCA

Bridge Courses

CO103 Introductory Computing

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

Course Outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Formulate program requirements.

CO2. Develop efficient algorithm for solving real life problem.

CO3. Implement the algorithm using C programming language.

CO4. Build sets of test data to validate the correctness of the program.

Course Contents:

Computer Fundamentals: History, Generations, Classification of Computers; Organization of a Computer; Concept of Programming and Programming Languages.

Introduction to Programming: Concept of Algorithm, Flow Chart, Pseudocode, Illustrative problem-solving Examples.

Features of a Programming Language: Character Set, Identifiers, Keywords, Data Types, Variables, Declarations, Operators & Expressions;

Statements: Assignment, Input/Output; Flow Control- Conditionals and Branching; Iteration; Functions, Function Types, Scope Rule; Recursion;

Arrays, Pointers, Structures.

(A programming language like C/C++ shall be used as a basis language. The same language is to be used for the laboratory).

Text Books:

1. Programming in C, Balaguruswamy.
2. Let us C, Kanetkar Y.
3. Programming in C, Gotfreid, McGrawHill
4. Fundamentals of Computers, Rajaram, V.

Reference Books:

1. The Elements of Programming Style, Kerningham, B. W.
2. Techniques of Program Structures and Design, Yourdon, E.
3. Theory and Problems of Computers and Programming, Schied, F. S.
4. The C Programming Language, Kerningham & Ritchie.

CO104 Computing Laboratory

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

Course outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Develop sound knowledge on software development tools such as algorithms, pseudocode, flowchart etc.

CO2. Gain understanding of the steps in the program development process such as source code generation, compilation, linking, execution and debugging.

CO3. Ability to write programs for simple computing problems such as factorial, recursion, problems involving use of structures etc.

CO4. Ability to implement some basic data structures, using the C language, such as- Array, Stack, Queue, and Linked List.

Course Contents:

Laboratory exercises shall involve the following:

1. Familiarization of a computer and the environment and execution of sample programs

2. Expression evaluation

3. Conditionals and branching

4. Iteration

5. Functions

6. Recursion

7. Arrays

8. Structures

9. Linked lists

10. Data structures

It is suggested that some problems related to continuous domain problems in engineering and their numerical solutions are given as laboratory assignments. It may be noted that some of basic numerical methods are taught in the Mathematics course.

Text Books:

1. The Elements of Programming Style, Kerningham, B. W.

2. The C Programming Language, Kerningham & Ritchie.

Reference Books:

1. Programming in C, Balaguruswamy.
2. Let us C, Kanetkar Y.
3. Programming in C, Gotfreid, McGrawHill

CO202: Digital Logic Design

Credits: 4 (3L-0T-1P)

Course outcomes:

At the completion of this course, the students should be able to do the following:

CO1. A thorough understanding of the fundamental concepts and techniques used in digital Logic circuits

CO2. Good hold over Boolean algebra and its application in combinational digital logic circuits.

CO3. Understanding of digital sequential circuits and classification of state machines

CO4. Ability to design, analyse and synthesize functional level digital logic circuits (combinational & sequential)

CO5. Ability to diagnose and resolve various hazards and timing problems in a digital circuit

Course Contents:

History & overview: Reasons for studying digital logic, people who influenced/contributed to the area of digital logic, applications of Digital Logic and introduction to a digital system.

Switching theory: Number systems and codes, Binary arithmetic, Complements, Boolean and switching algebra, Representation and manipulation of switching functions, Minimization of switching functions using algebraic method, K-map (2-, 3-, 4-, 5-variable), Quine McCluskey method.

Combinational logic circuits: Basic logic gates (AND, OR, NOT, NAND, NOR, XOR), Realization of switching functions with networks of logic gates, 2-level networks: AND-OR, OR-AND, NAND-NAND, NOR-NOR, Multi-level networks, Physical properties of logic gates (technology, fan-in, fan-out, propagation delay), Elimination of timing hazards/glitches.

Modular design of combinational circuits: Design of medium scale combinational logic modules - Multiplexers, demultiplexers, decoders, encoders, comparators, Arithmetic functions (adders, subtracter, carry look ahead), Multipliers, dividers, Arithmetic and logic units (ALUs), Hierarchical design of combinational circuits using logic modules.

Memory elements: Unclocked and clocked memory devices (latches, flip flops), Level vs. edge-sensitive, and master-slave devices, Basic flip flops (SR, D, JK, T), Asynchronous flip flop inputs

(preset, clear), Timing constraints (setup time, hold time) and propagation delays, Data registers (selection, clocking, timing), Random-access memory (RAM).

Sequential logic circuits: Finite state machines (FSMs), clocked and unclocked, Mealy vs. Moore models of FSMs, Modelling FSM behaviour: State diagrams and state tables, timing diagrams, algorithmic state machine charts, Analysis of synchronous and asynchronous circuits, Design of synchronous sequential circuits: State minimization, state assignment, next state and output equation realization, Sequential functional units: Data registers, shift registers, counters, sequence detectors, synchronizers, debouncers, controllers.

Fault detection and Location: Fault models for combinational and sequential circuits, Fault detection in combinational circuits; Homing experiments, Distinguishing experiments, machine identification and fault detection experiments in sequential circuits.

Laboratory component:

Study of TTL gate characteristics, Open collector and Tri-state gates, Clock generator and timer circuit.

Synthesis of combinational circuits using NAND, NOR and Multiplexers, Decoder and driver circuits for 7- segment LED displays, D/A converter and 4-bit ALU realization. Synthesis of sequential circuits – study of various types of flip-flops, realization of counters, shift registers and sequence generators.

ASM chart based synthesis such as Traffic light controller, Blackjack dealer and dice game ASM synthesis, etc.

Text Books:

1. M.M. Mano: Digital Logic and Computer Design, PHI (EEE)

2. Floyd and Jain: Digital Fundamentals, Pearson Education

3. Z. Kohavi: Switching and Finite Automata Theory, TMH.

Reference Books:

1. R.P. Jain: Modern Digital Electronics, Tata McGraw-Hill Education

2. J.F. Wakerly: Digital Design – Principles and Practices, Pearson Education.

3. S. Lee: Digital Circuits and Logic Design, PHI

CO209 Computing Workshop

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

Course outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Understanding of the main features of MATLAB/SciLab/R, and Python program development environments for their use in system simulation/ analysis, data analysis, machine learning etc.

CO2. Knowledge of the syntax semantics, data types supported in these environments.

CO3. Ability to implement simple mathematical functions/equations in these environments.

CO4. Ability to write codes to interpret and visualize simple mathematical functions and operations thereon using plots/displays.

CO5. Ability to write code in MATLAB/ Python for simulation/implementation of functions for real time environment application**s.**

Abstract:

This course aims to familiarize students with the basic concepts of two programming environments, MATLAB and Python, and their design trade-offs. The course covers basic programming aspects, and discusses the implementation of MATLAB and Python for different numerical methods. It also summarizes how both these environments provide a wide platform for problem solving and encourages students to explore this course for their upcoming project assignments. A group project has to be taken up by the students during this course.

Course Contents:

Introduction to MATLAB: MATLAB interface; variables; keywords; commands;

Operators: arithmetic, relational, logical, bitwise.

Vectors and Matrices in MATLAB: Vectors and matrices: creation, deletion, access, manipulation; Special matrices; complex matrix; matrix commands; matrix operations: determinant, minor, inverse, rank, eigen value and vectors.

MATLAB Scripts: M-files; Function files: primary function, sub-function; ways of creating script files; input/output functions.

Conditional statements in MATLAB: Statements: IF, IF-ELSE, nested IF-ELSE, SWITCH case; IS-function.

Iteration and Loops: Loops: FOR loop, WHILE loop, Nested loops; control statements: break, continue; Vectorizing.

Cell arrays: Creation, deletion, access, manipulation and operations in cell arrays.

Numerical methods using MATLAB: Set operations; Solving of linear equations; Non-linear equations; differentiation and integration.

Plotting in MATLAB: Visualizing results using plot; subplot; histogram; bar graph; pie chart etc.

Introduction to Python: Python overview; Interactive mode and Script mode; variables; keywords; datatypes: numeric, dictionary, Boolean, set, list, tuple, string; creation, deletion, access in different datatypes; operators: arithmetic, relational, assignment, logical, bitwise, membership, identity; input/output functions.

Conditional statements in Python: Statements: IF, IF-ELSE, nested IF-ELSE.

Iteration and Loops: Loops: FOR loop, WHILE loop, Nested loops; control statements: break, continue, pass.

Functions in Python: arguments: required, keyword, default, variable-length; creating function; return statements.

Matrix in Python: Numpy module for matrix in Python; creation, deletion, access, manipulation; types of matrices; matrix operations: determinant, minor, inverse, rank, eigen value and vectors; solving linear equations.

Plotting in Python: Matplotlib module for plotting in Python: plot, bar graph, pie chart, histogram, scatter plot, contour plot etc.

Text books:

1. Steven I. Gordon and Brian Guilfoos, Introduction to Modeling and Simulation with MATLAB and Python, Chapman & Hall, CRC Press, Computational Science Series, 2017.

2. Stormy Attaway, MATLAB: A Practical Introduction to Programming and Problem Solving, College of Engineering, Bosten University, Elsevier, 2009.

Reference books:

1. Mathworks, MATLAB: The Language of Technical Programming.
2. M. C. Brown, Python: The Complete Reference, Mc Graw Hills, 4th Edition, 2018.
3. A. Gilat, MATLAB: An Introduction with Applications, Wiley, 4th Edition, 2012

CO208 Object Oriented Programming

Credits: 4 (3L-0T-1P)

Course outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Understand different styles of programming and their differences.

CO2. Understand the principles of object-oriented problem solving and programming.

CO3. Outline the essential features and elements of the C++ programming language.

CO4. Apply the concepts of class, method, constructor, instance, data abstraction, function abstraction, inheritance, overriding, overloading, and polymorphism.

CO5. Analyze problems and implement C++ applications using an object-oriented approach.

Course Contents:

Data Abstraction: Class, object, constructors, destructors, memory allocations for objects, member functions, friend functions, templates.

Inheritance: Single & multiple inheritance, virtual base class.

Polymorphism: Compile time polymorphism- operator overloading, function overloading, static binding.

Run-time polymorphism: Virtual function, pure virtual function, abstract class, dynamic binding.

Exception handling.

Text Books:

1. Herbert Schild : The Complete Reference to C++, Osborne McGrawHill.
2. Bjarne Stroustrup: The C++ Programming Language, Addison Wesley
3. Rambaugh et al. : Object Oriented Modeling and Design, PHI(EEE).
4. Grady Booch: Object Oriented Analysis and Design, Pearson Education.

CO214 Computer Architecture & Organization

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

Course outcomes:

At the completion of this course, the students should be able to do the following:

CO1. A good understanding of the architectural and organizational aspects of computer systems at the machine level.

CO2. Understanding of the basic principles used for achieving the devices and the various functional modules.

CO3. Knowledge of the various design options and their tradeoffs for the functional units as well as the machine.

CO4 Understanding of the quantitative performance evaluation of the modules and the machine.

CO5 Understanding of the mechanisms built into the machine to provide for the advance digital systems, operating system etc. and a perspective of the architectures used in high performance computers.

Course Contents:

Basic organization of the computer and block level description of the functional units from program execution point of view; Fetch, decode and execute cycle;

Assembly language programming: Instruction set, instruction cycles, registers and storage, addressing modes; discussions about RISC versus CISC architectures;

Inside a CPU: information representation, computer arithmetic and their implementation; control and data path, data path components, design of ALU and data path, control unit design;

Memory and I/O access: Memory maps, Read Write operations, Programmed I/O, Concept of handshaking, Polled and Interrupt driven I/O, DMA data transfer; I/O subsystems: Input-Output devices such as Disk, CD- ROM, Printer etc.; Interfacing with IO devices, keyboard and display interfaces;

Inside the Memory: memory organization, static and dynamic memory; Cache memory and Memory Hierarchy – Cache memory access techniques; Virtual memory;

Introduction to Parallel Architectures: Instruction Level Parallel Processors- Pipelined, VLIW, Superscalar; Multiprocessors & Multicomputer Architectures, Vector Processing.

Laboratory experiments:

The assignments should cover the following:

1. Assignments on assembly language programming;
2. Experiments on synthesis / design of simple data paths and control unit;
3. Assignments on interfacing devices and systems like data acquisition systems;

Development kits as well as PCs/Workstations may be used for the laboratory, along with design / simulation tools as and when necessary.

Books:

1. Computer Architecture and Organization, Hayes J. P., McGrawHill
2. Computer Organization, Hamacher, Zaky, Vranesic, McGrawHill
3. Computer System Architecture, Mano M. M.

CO215: Computer Organization Lab

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

Prerequsites: CO103, CO104

Course outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Get a better understanding of constructing processor functional modules from digital devices.

CO2. Get hands on experience of writing and executing programs in Machine Level and Assembly Level programming.

CO3. A good understanding of the issues of writing computer programs at machine level and Assembly level.

CO4. Better understanding of design options and trade-offs.

CO5. Ability to develop digital systems.

Abstract:

This course is comprised of a laboratory component that is to be covered in parallel to Computer Architecture and Organization (CO212). The course is aimed at providing a practical understanding on building of the functional units and their integration and for appreciation of the issues on programming at the machine level.

Course Contents:

Circuit Design and Simulation: Register, Counter, Adder, Multiplier, Data Paths, Control Unit, ALU etc.

Introduction to 8086 family microprocessors, Architecture of 8086 – Register set, Concept of segments, use of the register set, use of stack, Instruction set, PSW Flags.

8086 assembly language programming: Software interrupts, Data Input/Output, Arithmetic Operations, String Handling, Branching, Looping, showing conditional and unconditional branches, and LOOP instruction, Creating Subroutines.

List of Laboratory Assignments:

Set 1: Designing CPU components using a Simulator:

- Register, Counter
- Adder, Multiplier
- Data Paths, Control Unit
- ALU, Memory unit

Set 2: 8086 Assembly Language Programming:

- Taking keyboard input for characters and numbers, Displaying and working with multi-digit number, Arithmetic operations
- Comparison operators, conditional and unconditional jumps, loops
- Working with array of numbers and strings of characters, finding average/mean of numbers, searching
- Writing Procedures, passing and returning values, use of stack

Text Books:

1. Computer System Architecture, Mano M. M, Pearson.
2. Guide to Assembly Language Programming in Linux, Sivarama Dandamudi, Springer

Reference Books:

1. IBM PC Assembly Language and Programming, Peter Abel, 3e, PHI.
2. Computer Organization and Design: The Hardware/ Software Interface, Patterson and Hennessy, Elsevier.
3. Computer Organization and Architecture: Designing for Performance 9E, William Stallings, Pearson Education.
4. Computer Organization, Hamacher, Zaky, Vranesic, McGrawHill.

CO218 Data Communication

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

Course outcomes:

At the completion of this course, the students should be able to do the following:

CO1 A good understanding of the various issues involved in the process of data communication and the layered architectural solutions in the computer communication systems.

CO2 Knowledge of the various communication media, their physical parameters, characteristics, strengths, and weaknesses.

CO3 Understanding on the underlying issues and concepts, principles, techniques and protocols used at the different layers of the data communication systems.

CO4 Understanding of the available design options and the tradeoffs.

CO5 Understanding of the performance parameters and techniques for performanc evaluation communication systems and application of the Mechanisms and Protocols

Course Contents:

Overview: Objectives and Applications of Computer Communication.

Computer Communication and Network Architecture: ISO-OSI reference model, design philosophy, layer, protocol, interface, and service concepts. Layer wise functionality.

Physical Layer: Concepts of data transmission: signal, communication channel, channel capacity, distortion & noise, line coding; modulation(analog and digital), modem; multiplexing- FDM, TDM, WDM, CDM etc; OFDM & spread spectrum techniques; switching, communication media-guides & unguided; standard protocols, RS- 232C, RS-449, X.21, xDSL, SONET, Frame relay, ATM etc.

Medium Assess Control in broadcast networks: ALOHA, CSMA, CSMA/CD, CSMA/CA, token ring, token bus etc, Standard LAN Protocols: (IEEE 802.X), FDDI, satellite networks, LAN switching, VLAN, WLAN, PAN and WiMax.

Data link layer: Framing, Error control techniques, Data link protocols and their performance, HDLC and PPP protocol.

Network layer: Introductory concepts and issue: Routing, Congestion and deadlock control Algorithms, Internetworking issues and devices, gateways, bridges and routers, IP & X.25 protocols.

Communication Laboratory:

Laboratory: Generation, testing, of AM, FM, and PM, Transmitter and receiver, PCM codec; Flow control, Error Control and MAC protocols on LAN trainer kit.

Books:

1. Stalling, Data and Computer Communication, 8e, PHI (EEE) [TAN]

2. Tanenbaum A.S., Computer Network, 5e, PHI (EEE)

References:

1. Forouzan B. A, Data Communication and Networking, 5e, Tata McGrawHill [LEG] Leon-Garcia, Widijaja, I., Communication, 5e, PHI (EEE)
2. B. P. Lathi, "Modern Analog and Digital Communication Systems", 3/e, Oxford University Press, 1998.

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.

Text books:

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.

Two-Year MCA Curriculum

EF103 Communicative English

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

Course contents:

A. Vocabulary and grammar

Discussion on the following before and/or after the activities mentioned in B, C & D

- Structure of simple sentences;
- Agreement of verb and subject;
- Useof adverbials; Tenses,
- Use of passive in scientific discourse, various types of questions,
- Directand indirect narration,
- Articles,
- Prepositions,
- English modal verbs,
- Errors in the use of individual words

B. Reading

- Reading and comprehension: global and local comprehension, drawing inferences
- Materials: Stories and essays(preferably a collection of comparatively short essays on scientific, interestingly written topics, short stories-adventure and scientific fiction)
- Reading silently in class followed by short comprehension questions, brief writing exercises, summaries in brief, personal responses (not typical question-answer type)-both oral and written. Reading material from Internet and talking and writing about them; reading scientific reports, articles collected from newspapers and magazines, Internet etc. and writing notes etc on them.

C. Writing

- Preparing project proposal and reports
- Writing applications of various types and for various purposes
- Curriculum vitae/Resume
- Letters to the editors, letters to various agencies
- Essay and Précis
- Notice both formal and informal/friendly
- Memo/notes

D. Speaking: Oral Communicative Activities

- Listening Comprehension: Information transfer activities: Pair and group works involving transfer of information: Gleaning information from different types of written materials including articles etc. and talking about them
- Pronunciation, Intonation, Stress and Rhythm
- Common Everyday Situations: Conversations and Dialogues
- Communication at Workplace
- Interviews
- Formal Presentations: Use of Graphic presentation, Presentation aids
- Formal group discussion
- Formal Speech

Course Outcomes

The students will

➢ Acquire proficiency in English enabling them to use English for communication and for study purposes;

➢ Develop their interactive/speaking skills by developing their ability to listen to English for formal, as in class lectures and informal, as in face to face interactive, situations) with a high degree of understanding, and help them to speak English with a reasonable degree of fluency and with an acceptable pronunciation of the sounds of English;

➢ Develop the basic skill for academic as well as non-academic writing.

Reference Books:

1. Sharma, S. and B. Mishra (2009).*Communication Skills for Engineers and Scientists.* PHI, New Delhi.
2. Wood, F. T. (2010) *A Remedial English Grammar for Foreign Students*. Macmillan, Delhi.

3. Greenbaum, Sidney.(2005).*Oxford English Grammar*. Oxford University Press, New Delhi, Indian Edition.
4. Kenneth, Anderson, Tony Lynch, and Joan Mac Lean. (2008). *Study Speaking*. CUP, New Delhi.
5. Lynch, Tony. (2008). *Study Listening*. CUP, New Delhi.
6. Thomson and Martinet. (2008).*A Practical English Grammar.* Oxford ELBS, Delhi.
7. Swan, Michael. *Practical English Usage*. OUP, New Delhi . 1995.
8. Exercises in Spoken English. Parts. I-III. CIEFL, Hyderabad. Oxford University Press

CS405 Discrete Mathematics

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

Course Outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Get sound understanding of the concepts of mathematical logic, sets, relations, recurrence relations, functions, graphs, trees.
CO2. Ability to construct and verify the correctness of mathematical arguments.
CO3. Ability to solve problems involving recurrence relations and generating functions.
CO4. Get sound grip on the operations on discrete structures such as sets, functions, relations.
CO5 Ability to use graphs and trees as problem solving tools

Course Content:

Part-I: 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.

Part-II: Logic:

- Logic operators, Truth table, Normal forms
- Theory of inference and deduction.

- Mathematical induction.
- Predicate calculus; predicates and quantifiers.
- Boolean algebra.
- Lattice.

Part-III: Combinatorics:

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

Part-IV: Modular Arithmetic:

- Congruence modulo, Fermat s Theorem, Euler s Theorem, Multiplicative Inverse, Reminder Theorem, FFT, Discrete Logarithm.

Text Books:

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)

CS412 Data structures

Co-requisite: CS416 OO programming & Data Structures Lab.

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

Course outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Apply knowledge of data structures s for abstraction of the data to systematically design efficient computational solutions.

CO2. Explain how basic data structures including: arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms.

CO3. Ability to analyze/characterize computational problems and choose appropriate data structures for solving them efficiently. .

CO4. Demonstrate understanding of the various sorting and searching techniques and their efficiency.

CO5 Ability to analyze efficiency of various data structures in terms of space and time complexity.

Course Contents:

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-5: 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).

CS416 OO programming & Data Structures Lab.

Co-requisite: CS412 Data Structures

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

Course Outcomes :

At the completion of this course, the students should be able to do the following:

CO1**.** Understand different programming paradigms, their advantages and disadvantages.

CO2.Understanding of the basic OO principles and their application in the design and implementation of different OO computing solutions.

CO3. Write C++ programs by choosing appropriate data structures to solve a problem.

CO4. Master the standard data structure library of C++ programming language

CO5 Develop skills in implementations and applications of data structures and different sorting and searching techniques.

Course Contents:

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.

Text books:

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.

IC361 Accounting and Financial Management

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

Course Contents:

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. Bhattacharjee Ashis 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

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

Course outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Clear understanding of the concepts of database systems, their advantages, and applications.

CO2. Ability to carry out the conceptual modelling of the data for a given application.

CO3. Ability to evaluate functional dependencies in the data and model the data in a suitable normal form for a relational database.

CO4. Ability to carry out the database design using appropriate database model and to write optimized queries.

CO5 Familiarity with the database modelling techniques for emerging application areas

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 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,

Reference Books:

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–0T–2P)

Course outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Ability to create and manage a database using RDBMS like Oracle.

CO2. Ability to write queries in SQL for retrieval of information from a database.

CO3. Ability to build user-defined functions and procedures using PL/SQL.

CO4. Ability to develop database application

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:

1. Basic commands of SQL and SQL Plus
2. Creating and modifying database tables, inserting, deleting and updating data in database tables using SQL
3. SQL commands for retrieving data from database tables
4. Creating and updating database triggers
5. Writing and executing SQL scripts
6. SQL scripts for building simple reports
7. Basics of PL/SQL
8. User-defined functions and procedures
9. DBA commands and database authorization
10. Developing GUIs using Oracle Forms & reports/PHP/Java etc.
11. 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

CS417 Operating Systems

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

Course Outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Clear understanding of the operating system policies and algorithms for the management of computer system resources.

CO2. Good knowledge of the of Operating system module functionalities, their working and the design issues

CO3. Ability to write programs with OS system calls for IPC, resource utilization etc.

CO4. Ability work with OS kernel.

CO5. Ability to develop device drivers.

Prerequisites: CS412 (Data Structures), CO214(CAO)

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 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:

1. Operating systems: Concepts and Design, M. Milenkovic (McGraw Hill, 2001)
2. Modern Operating Systems, A. S. Tanenbaum (Pearson, 2009)
3. Design of the Unix Operating System, M. J. Bach (Prentice Hall of India, 1986)
4. Operating Systems: Three Easy Pieces, Remzi and Andrea Arpaci-Dusseau
5. 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)

CS418 Operating Systems Lab

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

Prerequisites: CS417(OS), CO103 (IC), CO104 (CL), CS412 (Data Structures)

Course outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Ability to implement operating system software features,

CO2. Good understanding of the working of an operating system kernel

CO3. Ability to carry out system level programming especially the kernel coding.

CO4. Ability to develop algorithm for operating system and for implementing file system.

CO5. Ability to use the OS simulation tools, the system utilities.

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 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
7. http://www.ee.surrey.ac.uk/Teaching/Unix/
8. http://developer.android.com/guide/index.html
9. Operating Systems: Three Easy Pieces, Remzi and Andrea Arpaci-Dusseau

http://pages.cs.wisc.edu/~remzi/OSTEP/

10 Web links and tutorial materials will be provided time to time

CS513 Software Engineering

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

Course outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Clear understanding of the software life cycle models and the process of project planning and management techniques

CO2. Good understanding of the various software design methodologies.

CO3. Ability to carry out software requirements analysis for a given project.

CO4. Ability to convert software requirements to a design model.

CO5. Good understanding of the various testing and maintenance requirements and the ability to develop appropriate testing procedures for a given project.

Course Contents:

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.

Reference Books:

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)

CS519 Computer Networks

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

Prerequisites: CO218 (Data Communication)

Course Outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Clear understanding of the principle and the design of layered the Computer Network Architecture.

CO2. Knowledge of the key functions performed by different widely known mechanisms and protocols at Network, Transport and Application Layers and the functionalities and use of the different networking devices.

CO3. Understanding the basic working behaviour of TCP, UDP and other Internetworking protocols.

CO4. Understanding of the needs of network security and the of different network security and authentication mechanisms.

CO5. Knowledge of the needs and protocols used in different network applications and the basic mechanisms used in data compression

Course Contents:

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 Books:

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

Reference books:

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)

Prerequisites: CO218 (Data Communication)

Course Outcomes:

At the completion of this course, the students should be able to do the following:

CO1. Ability to design and configure networks with Router, ARP, DHCP, DNS, and Gateway etc.

CO2. Ability to write client-server applications in C/C++/Java

CO3. Ability to use different OS tools to configure network and network protocols.

CO4 Ability to use network simulators to study behaviour of different protocols and to analyse their performance 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.

Text Books:

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

Reference Books:

1. Online and weblink reference manuals for network simulators