DEPARTMENT OF COMPUTER APPLICATIONS, SMIT, MAJITAR

SYLLABUS FOR MCA COURSE CURRICULUM

Total Credits:
24.5+24.5+24.5+24.5+26+16=140

(2016-2017 BATCH)

DEPARTMENT OF COMPUTER APPLICATIONS

Program Educational Objectives for MCA Course

The AICTE-approved MCA curriculum at SMIT/SMU provides the education and training necessary to accommodate students who want professional preparation in computer science applications but do not necessarily have a strong interest in computer systems hardware. Within the curriculum, students study subject matter in programming languages, software systems, information and data management, software engineering, principles of programming languages, data structures, computer architecture, theory of computation and formal languages, operating systems, distributed systems, computer modeling, computer networks.

The MCA students are prepared for employment in a wide spectrum industrial and business environments and successful professionals in the field with a solid foundation in computer applications and the knowledge of culture and vocabulary of technology to work with peoples from other disciplines. They are sufficiently prepared to continue life-long learning including to be accepted into or complete advanced degree programs such as M. Tech/M.Phil./Ph.D.

<u>MCA program objectives are:</u>

- Our MCA students will apply their knowledge and skills to succeed in their career and/or obtain an advanced degree.

- They will function ethically and responsibly, and will remain in formed and involved as full participants in our profession and our society.

- Our Postgraduates will demonstrate strong communication skills and the ability to function effectively in multi-disciplinary teams.

- Our Postgraduates will apply basic principles and practices of computing grounded in mathematics and science to successfully complete software related projects to meet customer business objectives and/or productively engage in research.

DEPARTMENT OF COMPUTER APPLICATIONS

MCA COURSE STRUCTURE – SEMESTER WISE

I ST. YEAR - I ST. SEMESTER

Subject Code	Subject Name	Teaching Department	Theory Hours	Practical/ Tutorial Hours	Credit Points
MA2108	DISCRETE MATHEMATICS	MATHS	3L	1T	4
CA2102	WEB PAGE DESIGN	CA	3L	1T	4
CA2103	DIGITAL LOGIC	CA	3L	1T	4
CA2104	PROGRAMMING CONCEPTS AND ‘C’ LANGUAGE	CA	3L	1T	4
BA2110	ACCOUNTING AND MANAGERIAL ECONOMICS	MGMT	3L	1T	4
CA2161	WEP PAGE DESIGN LAB	CA	-	3P	1.5
CA2162	C PROGRAMMING LAB	CA	-	3P	1.5
CA2163	DIGITAL LOGIC LAB	CA	-	3P	1.5
Total Credits					**24.5**

I ST. YEAR - II ND. SEMESTER

Subject Code	Subject Name	Teaching Department	Theory Hours	Practical/ Tutorial Hours	Credit Points
MA2208	QUANTITATIVE ANALYSIS FOR COMPUTER APPLICATIONS	MATHS	3L	1T	4
CA2202	COMPUTER GRAPHICS	CA	3L	1T	4
CA2203	DATA STRUCTURES	CA	3L	1T	4
CA2204	OBJECT ORIENTED PROGRAMMING WITH C++	CA	3L	1T	4
CA2205	COMPUTER ORGANIZATION AND ARCHITECTURE	CA	3L	1T	4
CA2261	COMPUTER GRAPHICS LAB	CA	-	3P	1.5
CA2262	DATA STRUCTURES LAB	CA	-	3P	1.5
CA2263	C++ LAB	CA	-	3P	1.5
Total Credits					**24.5**

Prof.(Dr.) R. Pradhan
CA, SMIT, Member (BOS)

Prof.(Dr.) K. Sharma
CSE, SMIT, Member (BOS)

Dr. S. Borah
Associate Professor,
CA, SMIT (BOS)

Mr. Pijush Barthakur
Associate Professor,
CA, SMIT (BOS)

Prof.(Dr.) M.K. Ghose
CSE, SMIT,
Chairman, (BOS)

Prof. (Dr.) R.K. Samanta
Professor, NBU
(External Member)

Prof. (Dr.) Nabarun Bhattacharya
Associate Director, C-DAC, Kolkata
(External Member)

DEPARTMENT OF COMPUTER APPLICATIONS

II ND. YEAR - III RD. SEMESTER

Subject Code	Subject Code	Subject Code	Subject Code	Subject Code	Subject Code
CA2308	NUMERICAL ANALYSIS- A SOFTWARE APPROACH	CA	3L	1T	4
CA2302	SYSTEM PROGRAMMING AND MICROPROCESSOR	CA	3L	1T	4
CA2303	DATABASE MANAGEMENT SYSTEM	CA	3L	1T	4
CA2304	OPERATING SYSTEMS	CA	3L	1T	4
CA2305	JAVA PROGRAMMING	CA	3L	1T	4
CA2361	OPERATING SYSTEMS LAB	CA	-	3P	1.5
CA2362	JAVA PROGRAMMING LAB	CA	-	3P	1.5
CA2363	DATABASE MANAGEMENT SYSTEM LAB	CA	-	3P	1.5
CA2306	[BRIDGE COURSE] STATISTICAL METHODS AND SIMULATION	CA	3L	1T	4
Total Credits					**24.5**

II ND. YEAR - IV TH. SEMESTER

Subject Code	Subject Name	Teaching Department	Theory Hours	Practical/Tutorial Hours	Credit Points
CA2401	SOFTWARE ENGINEERING	CA	3L	1T	4
CA24**	ELECTIVE –I	CA	3L	1T	4
CA2403	ASP.NET APPLICATION	CA	3L	1T	4
CA2404	COMPUTER NETWORK	CA	3L	1T	4
CA2405	IT LAW AND PRACTICES	CA	3L	1T	4
CA2461	SOFTWARE ENGINEERING LAB	CA	-	3P	1.5
CA2462	COMPUTER NETWORK LAB	CA	-	3P	1.5
CA2463	ASP.NET APPLICATION LAB	CA	-	3P	1.5
Total Credits					**24.5**

Prof.(Dr.) R. Pradhan
CA, SMIT, Member (BOS)

Prof.(Dr.) K. Sharma
CSE, SMIT, Member (BOS)

Dr. S. Borah
Associate Professor,
CA, SMIT (BOS)

Mr. Pijush Barthakur
Associate Professor,
CA, SMIT (BOS)

Prof.(Dr.) M.K. Ghose
CSE, SMIT,
Chairman, (BOS)

Prof. (Dr.) R.K. Samanta
Professor, NBU
(External Member)

Prof. (Dr.) Nabarun Bhattacharya
Associate Director, C-DAC, Kolkata
(External Member)

DEPARTMENT OF COMPUTER APPLICATIONS

III RD. YEAR - V TH. SEMESTER					
Subject Code	**Subject Name**	**Teaching Department**	**Theory Hours**	**Practical/Tutorial Hours**	**Credit Points**
CA2504	FORMAL LANGUAGES AND AUTOMATA THEORY	CA	3L	1T	4
CA2502	ANALYSIS AND DESIGN OF ALGORITHMS	CA	3L	1T	4
CA2503	OBJECT ORIENTED DESIGN AND ANALYSIS USING UML	CA	3L	1T	4
CA25**	ELECTIVE-II	CA	3L	1T	4
CA25**	ELECTIVE-III	CA	3L	1T	4
CA2561	UML LAB	CA	-	3P	1.5
CA2562	ANALYSIS AND DESIGN OF ALGORITHMS LAB	CA	-	3P	1.5
CA2571	MINI PROJECT	CA	-	3P	2
CA2581	INDUSTRIAL TRAINING/COURSEWORK	CA	-	3P	1
Total Credits					**26**

III RD. YEAR-VI TH. SEMESTER					
Subject Code	**Subject Name**	**Teaching Department**	**Duration**		**Credit Points**
CA2675	Major Project	CA/External Agencies	16 weeks		16
Total Credits					**16**

Prof.(Dr.) R. Pradhan
CA, SMIT, Member (BOS)

Prof.(Dr.) K. Sharma
CSE, SMIT, Member (BOS)

Dr. S. Borah
Associate Professor,
CA, SMIT (BOS)

Mr. Pijush Barthakur
Associate Professor,
CA, SMIT (BOS)

Prof.(Dr.) M.K. Ghose
CSE, SMIT,
Chairman, (BOS)

Prof. (Dr.) R.K. Samanta
Professor, NBU
(External Member)

Prof. (Dr.) Nabarun Bhattacharya
Associate Director, C-DAC, Kolkata
(External Member)

DEPARTMENT OF COMPUTER APPLICATIONS

LIST OF ELECTIVES FOR MCA (IV SEMESTER)			
1	CA2431	BIOINFORMATICS	4
2	CA2432	DIGITAL IMAGE PROCESSING	4
3	CA2433	DATA WAREHOUSING AND DATA MINING	4
4	CA2434	MANAGEMENT INFORMATION SYSTEMS	4
5	CA2435	UNIX/LINUX INTERNALS	4
6	CA2436	ENTERPRISE RESOURCE PLANNING	4
7	CA2438	OPTIMIZATION TECHNIQUE	4
8	CA2439	MOBILE APPLICATION DEVELOPMENT	4
LIST OF ELECTIVES FOR MCA (V SEMESTER)			
1	CA2531	ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS	4
2	CA2532	GRAPH THEORY	4
3	CA2533	PRINCIPLES OF PROGRAMMING LANGUAGE	4
4	CA2534	SYSTEM SIMULATION AND MODELLING	4
5	CA2535	AD-HOC WIRELESS NETWORKS	4
6	CA2536	ADVANCED COMPUTER ARCHITECTURE	4
7	CA2537	COMPILER DESIGN	4
8	CA2538	CLOUD COMPUTING	4
9	CA2539	CRYPTOGRAPHY AND NETWORK SECURITY	4
10	CA2540	DISTRIBUTED DATABASE SYSTEM	4
11	CA2541	AUTONOMOUS MOBILE ROBOTICS AND COMPUTATIONAL INTELLIGENCE	4
12	CA2542	SOFT COMPUTING	4
13	CA2543	WIRELESS SENSOR NETWORKS	4
14	CA2544	WEB ENGINEERING	4

Prof.(Dr.) R. Pradhan
CA, SMIT, Member (BOS)

Prof.(Dr.) K. Sharma
CSE, SMIT, Member (BOS)

Dr. S. Borah
Associate Professor,
CA, SMIT (BOS)

Mr. Pijush Barthakur
Associate Professor,
CA, SMIT (BOS)

Prof.(Dr.) M.K. Ghose
CSE, SMIT,
Chairman, (BOS)

Prof. (Dr.) R.K. Samanta
Professor, NBU
(External Member)

Prof. (Dr.) Nabarun Bhattacharya
Associate Director, C-DAC, Kolkata
(External Member)

DEPARTMENT OF COMPUTER APPLICATIONS

Minimum No. of credits to be earned for promotion		
MCA Course:		
From	To	Min. credits to be earned
I Year	II Year	30/49
II Year	III Year	60/98
Final	Final	140/140
MCA Lateral Entry Course:		
From	To	Min. credits to be earned
II	III	30/49
Final	Final	91/91
Credits:		4 each
Theory paper		1.5 each
Lab paper		
Promotion criteria are applicable for year-wise promotion.		

DEPARTMENT OF COMPUTER APPLICATIONS

MA 2108 **(3L + 1 T hrs./week)**

DISCRETE MATHEMATICS

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: Boolean algebra generally deals with design of h/w circuits forms a basis of the computer scientists, since computers can understand only machine level language which is of zeros and one so understanding of Boolean algebra is important. More over Boolean algebra also deals with minimization of the logic design which has considerably reduced the size of hardware so according to me each and every computer scientist should have a basic understanding of Boolean algebra Some applications of logic: architecture (logic gates) software engineering (specification and verification) programming languages (semantics, logic programming) databases (relational algebra) artificial intelligence (automatic theorem proving)theory of computation (general notion of complexity)

Pre- requisite: Basic idea of set theory, relation, functions.

UNIT- I

Modern Algebra [20 Hrs.]

Sets, subsets, set operation, Cartesian product, relation: properties, equivalence relation, partition, Transitive closure and Warshell's algorithm. Function: Injective, Surjective and Bijective function, composite function. Principle of inclusion and exclusions (statement only), Simple problems, Generating functions and recurrence relation.

Binary operation, semi groups, product and quotients, groups product and quotient, subgroups and normal subgroups. Burnside theorem (statement only and simple problems) coding & decoding. Rings, Integral domain and Fields.

UNIT- II

Boolean Algebra [10 Hrs.]

Partial order, algebraic structures and POSET, Lattices, distributive and complemented lattices, Boolean Lattice, Uniqueness of Boolean Lattices, Boolean expression & function.

Mathematical Logic [10 Hrs.]

Statement and notations, connectives, normal forms, well-formed formulas, implication, Tautology, Rules of inference, Predicate calculus, Causal form, Resolution and refutation and answer extraction. Simple examples-Language and Grammar.

Text Books:

1. C.L. Liu, Elements of Discrete Mathematics, McGraw-Hill Education.
2. J. P. Tremblay and R. Manohar, Discrete Mathematical Structure with Application to Computer Science, Tata McGraw Hill Education.
3. N. J. Nilsson, Principles of Artificial Intelligence, Springer.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2102 **(3L + 1 T hrs./week)**

WEB PAGE DESIGN

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: To familiarize the student with the structure and use of Internet application programming languages and with the elements of user-interface design. The focus is on client-side scripting using DHTML, CSS, JavaScript, and an introduction to server-side scripting using PHP. The emphasis of this course will be on syntax and debugging, web form processes and data validation using common programming structures, dynamic content using JavaScript and DHTML, and working with objects and cookies.

Pre-requisites: Programming concepts.

Learning outcomes: By the end of this course, the students will learn

1. How to make rich, visually stunning websites
2. How to understand and write HTML
3. How HTML and CSS (Cascading Style Sheets) work together to create a web page

UNIT- I

Applets [6 hrs.]
Introduction, Local and remote applets, How applets differ from applications, Preparing to write applets, Building applet code, Applet life cycle, Creating an executable applet, Designing a web page, Applet tag, Adding applet to html files, Running the applet.

AWT Controls and Programming [4 hrs.]
More about applet tag, passing parameters, AWT controls, frames, Windows, panels, Simulation programs, Layout.

Introduction to HTML [5 hrs.]
The set of 150 tags, Block level and text level tags, Images and image maps, Client side server side maps, Anchor elements, Tables, Frames, Forms.

Introduction to DHTML [5 hrs.]
Cascading style sheets, DOM, Scripting, ActiveX pad with embedded active controls.

UNIT- II

Introduction to VB Script [6 hrs.]
Example programs and some client side validation.

Introduction to Java Script [6 hrs.]
Example programs and some client side validation.

Introduction to XML [4 hrs.]

What's wrong with HTML, The SML solution, Displaying XML content, XML editors, XML tag format, XML declaration, Elements, Element attributes, Empty elements, Comments, Character entities, CDATA blocks

Processing forms [4 hrs.]

CGI submit, get ( ) and post ( ) methods

Text Books:

1. Herbert Schildt, Complete reference Java 2, Tata McGraw Hill.
2. Thomas A. Powell, HTML The Complete Reference, Tata McGraw Hill.

Reference Books:

1. Achyat. S. Godbole and Atul Kahate, Web Technologies, Tata McGraw Hill.
2. Jamie Jaworski, Java 1.2 Unleashed, Sams Publishing.
3. Steven Holzner, Java 2 Swing, servlets, JDBC and Java Beans Programming Black Book, Dreamtech Press.
4. Robert Niles et.al., CGI by Examples, Que.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2103 **(3L + 1 T hrs./week)**

DIGITAL LOGIC

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course provides a detailed knowledge of Boolean algebra and its application in digital design. It provides an in-depth coverage of combinational logic circuit analysis, minimization and design techniques. It also covers the basic concepts of sequential circuits including the use of state diagrams, timing diagram and state tables to represent the behaviour of interfacing circuits, sequential circuits and memory devices.

Pre-requisites: No Departmental pre-requisites, Knowledge of basic Electronics and Logic gates is desirable but not necessary.

Learning outcomes: Upon completion of this course, students should be able to

1. Convert values from one number system to another number system or to a coding system and perform arithmetic operations to any number system
2. Write Boolean equations from truth tables and implement Boolean equations with discrete logic gates
3. Design and understand the function of basic combinational logic circuits
4. Construct truth table and draw waveform diagram of latches and flip-flops
5. Design and analyse sequential logic circuits such as latches and flip-flops
6. Understand the concept of memory structures

UNIT – I

Binary System [6 hrs.]
Binary numbers and number base conversion, complements- Binary Codes, Binary storage, registers, binary logic and integrated circuits.

Boolean Algebra and Logic gates [4 hrs.]
Axiomatic definitions of Boolean algebra – Basic theorem and properties, Boolean functions, canonical and standard forms, other logical operations, Digital logic gates and IC digital logic families.

Simplification of Boolean Functions [6 hrs.]
The map method: Maps up to four variables, product of sums, sum of product, Simplification, NAND and NOR implementations, don’t care conditions, Tabulation method: determination and selection of prime implicate.

Combinational Logic [6 hrs.]
Adders, Subtractor, Code Converter etc., Multilevel NAND and NOR circuits, EX-OR and Equivalence functions, combinational logics.

UNIT – II

Synchronous Sequential Logic [6 hrs.]
Different types of Flip-flops and their triggering (D-flip flops, RS-Flip flops, JK-Flip flops, Master – Slave Flip flops)

Counters and Counter Applications [6 hrs.]
Clocking, Serial-Parallel data transfer, shift registers, Mod-n-counters, Ripple counter, Ring counter, ring counter application.

Memory Devices: [6 hrs.]
RAM, ROM, Word, Capacity, Address, Access time, Memory Hierarchy. Magnetic Core Memory, Semiconductor memory (SRAM and DRAM), Magnetic tape storage, magnetic disk storage, Winchester disk and floppy disk. Linear selection and coincident selection.

Text Books:

1. Morris Mano, Digital Logic and Computer Design, PHI.
2. Thomos L. Floyd, Digital Fundamentals, Universal Book Stall.

Reference Books:

1. Bartee, Digital Computer Fundamentals, Tata McGraw Hill.
2. Taub and Schilling, Digital Integrated Electronics, Tata McGraw Hill International Edition.
3. Tocci and Widmer, Digital Systems, Prentice Hall.
4. William I. Fletcher, An Engineering Approach to Digital Design, PHI

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2104 **(3L + 1 T hrs./week)**

PROGRAMMING CONCEPTS AND ‘C’ LANGUAGE

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course intends to provide thorough knowledge of programming and C language. It covers C programming topics from elementary to file handling with proper programming examples.

Pre-requisites: Fundamentals of computers and Basics of Computer programming concepts.

Learning outcomes: Having successfully completed the module, students will be able to demonstrate knowledge and understanding of:

1. The principles of designing structured programs
2. When and how to use the appropriate statements available in the C language
3. Explain the principles of structured program design
4. Write, Compile and Debug basic C programs using an IDE and using a workstation
5. Write basic C programs using Selection statements, Repetitive statements, Functions, Pointers, Arrays, Strings

UNIT- I

Programming Concepts [2 hrs.]

Programming Languages, Types of Programming Language, Structured vs. Unstructured Programming Language, Procedural vs. Object Oriented Programming Language

‘C’ Fundamentals [3 hrs.]

Importance of C, Basic Structure of C program, character set, token, Identifiers and keywords, Constants, variable, Data Types, Variable Declarations.

Operators and Expressions [5 hrs.]

Arithmetic operators, Relational and logical operators, Increment and decrement operators, Conditional Operators, Bit-wise operators, Special Operators, Assignment operators. Arithmetic expressions and evaluation, Precedence and associativity of operators, Type conversions.

Input and Output [4 hrs.]

Standard input and output, formatted input and output.

Flow of Control [6 hrs.]

Branching statements- if, if-else, else-if ladder, nested if, switch. Looping Statements- while do while, and for. Jumps in loop- break, continue, and goto.

UNIT – II

Arrays and Pointers [6 hrs.]

Arrays– One Dimensional and multidimensional arrays- Declaration, Initialization. Strings- Initialization, Display, String manipulation functions, array of strings. Pointers- declaration and initialization. Pointers and Arrays, Pointer and strings, array of pointers.

User defined functions [4 hrs.]

Basics of functions-function types, need for functions, elements and categories of functions, passing parameter to a function, passing array and string to a function, Recursion, Scope rules, function pointer.

Structures, Unions and Dynamic Memory Allocation[6 hrs.]

Structures- definition, declaration and initialization, accessing structure member, operations on structure member and variables. Arrays of structures, array within structure. Structure within structure, Pointers to structures, Self-referential structures, Unions, Dynamic Memory Allocation and Linked List.

File management in C [4 hrs.]

Introduction, Defining and opening a file, Closing a file, I/O operations on files, Error handling during I/O operations, Random access to files, command line arguments.

Text books

1. E. Balagurusamy, Programming in ANSI C, Tata McGraw Hill.
2. Byron S. Gottfried, Programming with C, Schaums' Outline Series.

Reference Books:

1. Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language, Prentice Hall.
2. K.R. Venugopal, S.R. Prasad, Mastering C, Tata McGraw Hill.
3. Yashavant Kanetkar, Let us C, BPB Publications.
4. Yashavant Kanetkar, Pointer in C, BPB Publications.

DEPARTMENT OF COMPUTER APPLICATIONS

BA 2110 **(3L + 1 T hrs./week)**

ACCOUNTING AND MANAGERIAL ECONOMICS

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective

Accounting : The objective of this segment is to enable the students of computer application to understand the basic concepts in book-keeping and accounting and utilize the same in planning, controlling and decision making process of a business firm.

Managerial Economics : The objective of this segment is to enable the students to understand relevance of the basic concepts and techniques in micro-economics which are of utmost significance in day-to-day business system.

UNIT-I

Introduction to Bool-keeping & Accounting-[3hrs]

Introduction to Book-keeping, Objectives and Needs of Book-keeping, Introduction to Accounting, Objectives, Significance, Accounting Process, Branches of Accounting, Difference between Book-keeping and Accounting, Users of Financial Statements.

Double Entry System and Single Entry System-[2hrs]

Double Entry System and its characteristics, advantages and disadvantages, Single Entry System and its characteristics, advantages and disadvantages. Difference between Double Entry System and Single Entry System.

Accounts & Trail Balance- [9hrs]

Account, Types of Account, Rules of debiting and crediting, Books of Accounts, Journal and its features and advantages, Ledger and its features and advantages, Types of Journal and Ledger, Difference between Journal and Ledger, Preparation of Journal and Ledger. Trail Balance and its characteristics, advantages and disadvantages, Preparation of Trial Balance.

Financial Statements-[2hrs]

Trading and Profit and Loss Account and its significance, Balance Sheet and its significance, Preparation of Trading and Profit and Loss Account and Balance Sheet.

UNIT-II

Introduction to Economics-[2hrs]

Nature and Scope of Managerial Economics - Business Decisions and Economic Analysis

Theory of Consumer Behaviour-[5hrs]

Theory of Demand -- Analysis of Consumer Behavior - Cardinal Utility Approach, Law of Demand, Exceptions; Determinants of Market Demand, Demand Function.

Ordinal Utility Approach-[3hrs]

Ordinal Utility Approach - Meaning of Indifference Curve - Marginal Rate of Substitution - Consumer Equilibrium - Income Effect, Substitution Effect, Price Distinctions

Elasticity of Demand-[2hrs]

Price Elasticity of Demand, Income Elasticity - Cross Elasticity - Practical Importance of Elasticity of Demand,

Demand Forecasting [1hr]

Demand Forecasting – Importance, Methods of Forecasting - Direct and Indirect Methods.

Theory of Production-[2hrs]

Theory of Production- Production Function - Laws of Production - Internal & External, Economies of Scale.

Theory of Cost and Break-Even Analysis-[2hrs]

Theory of Cost and Break-Even Analysis - Cost Concepts - Cost Output Relation, Importance of Break-Even Analysis in business decision making

Market Structure-[3hrs]

Market Structure and Pricing Theory.

Text Book

1. R. L. Gupta and V. K. Gupta, Principles and Practices of Accountancy, Sultan Chand & Sons
2. S. N. Maheshwari & S. K. Maheshwari, Financial Accounting, Vikas Publishing House
3. Yogesh Maheshwari, Managerial Economics, PHI Learning
4. H.L. Ahuja, Managerial Economics, S. Chand Publishing
5. M.L. Seth, Principles of Economics, Lakshmi Narain Agarwal, Educational Publishers

Reference Book

1. Ashok Banerjee, Financial Accounting, Excel Books India
2. P. C. Tulsian, Financial Accounting, Pearson Education
3. Sarma and Gupta, Financial Management, Kalyani Publishers
4. Peterson & Lewis, Managerial Economics, Pearson Education

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2161 **(3 hrs./week)**

WEB PAGE DESIGN LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA2102 – Web Page Design and associated prerequisites.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2162 **(3 hrs./week)**

C PROGRAMMING LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA2104 - Programming concepts and C language and associated prerequisites.

CA 2163 **(3 hrs./week)**

DIGITAL LOGIC LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA2103 - Digital Logic and associated prerequisites.

DEPARTMENT OF COMPUTER APPLICATIONS

MA2208 **(3L + 1 T hrs./week)**

QUANTITATIVE ANALYSIS FOR COMPUTER APPLICATIONS

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective - : The student of MCA needs powerful analytical tools to analyse the algorithms and computer systems. Before analysing the algorithm or system, various probability distribution have to be specified. For computer system serving a large number of users, several types of random phenomena needs to be considered, such as large population of diverse users, the arrival patterns of jobs. The theory of stochastic process is required to evaluate various measures of system effectiveness, response time, reliability and availability. To study different engineering subjects like Information Transmission and Coding Theory, Image Processing and Artificial Neural Network the help of Probability & Statistics is needed.

Pre-requisite: For this subject - Set theory, calculus, permutation and combination.

UNIT- I

Probability & Statistics
Axioms and Models of Probability, Conditional Probability, Bayes Rule – Examples. Random Variables – Simple Examples, Discrete and Continuous random variable, Jointly distributed Random Variables, Distribution of random variables – examples.

Expectation – Moments, Variance, Transform Methods, MTTF, Inequalities and Limit theorems – Examples, Conditional Distribution and Conditional Expectation.

Elements of Statistics, Frequency Distribution, Measure of Central tendency, Covariance, Correlation Coefficient, Regression, Least Square Curve Fitting.
Random Sampling, Sampling Distribution, Parameter Estimation and testing of Hypothesis.

UNIT- II

Stochastic Process & Queuing Theory
Stochastic Process – Bernoulli, Poisson and Renewal Process, Availability Analysis.
Markov Chain – Continuous parameter and discrete parameter Markov Chain. Concept of queue.
M/G/I queuing system, Open and closed queuing networks – examples.

Text Books:

1. Erwin Kreyszig, Advanced Engineering Mathematics, Wiley India.

Reference Books:

1. K.S. Trivedi, Probability, Statistics and Reliability, Queuing and Computer Science Applications, Prentice Hall of India.
2. P.L. Meyer, Introductory Probability Theory and Statistical Applications, Oxford * IBH Publishers.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2202 **(3L + 1 T hrs./week)**

COMPUTER GRAPHICS

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course highlights the overview of display devices and peripherals, software and techniques used in computer graphics. Study of the principles of interactive computer graphics; systems organization and device technologies for raster and vector displays; region filling techniques; 2-D and 3-D viewing, clipping, segmentation and interaction handling; 3-D geometrical transformations, projections and hierarchical data structures for graphics modelling including hidden lines and surfaces, lighting, texturing, shading and colour models.

Pre-requisites: Programming concepts and Basic Mathematics

Learning objectives: On successful completion of course, the learner will be able to

1. Demonstrate an understanding of contemporary graphics hardware.
2. Create interactive graphics applications
3. Write program functions to implement graphics primitives and geometrical transformations.
4. Demonstrate an understanding of the use of object hierarchy in graphics applications.
5. Write program functions to implement visibility detection, computer graphics animation and 2D/3D transformation

UNIT- I

Display devices [5 hrs.]
Line & Point plotting systems, Raster, Vector, Pixel and Point plotters, Continual refresh and storage displays, Digital frame buffer, Plasma panel displays, Character generators, Color display techniques.

Interactive Graphics [5 hrs.]
Pointing and positioning devices, Cursor, Light pen, Tablet, Mouse, Joystick, Track ball, Digitizer, Interactive graphical techniques, Positioning, Zooming, Rubber-band techniques, Sketching dragging.

2-D Graphics [5 hrs.]
Line drawing techniques, Fast drawing circles and Ellipse algorithms, Transformations (Screen co-ordinates, user co-ordinates, view algorithm), Polygon filling.

Windowing & Clipping [6 hrs.]
Windowing concepts, Clipping algorithms, Line clipping, Area clipping, Text clipping, Blanking, Window to view port transformations.

UNIT- II

3-D Graphics [5 hrs.]
Transformations, Perspective and parallel projections.

Hidden line removal algorithms [5 hrs.]
Depth buffer algorithm, Scan line coherence algorithm, Area coherence algorithm, Back face algorithm.

Curves and Surfaces [5 hrs.]
Shape description and requirements, Bezier Methods, B-spline methods.

Graphics Standard PHIGS [4 hrs.]
Introduction, Coordinate systems, Transformations structures, Input devices.

Text Books:

1. Donald Hearn & M. Pauline Baker, Computer Graphics, Pearson Education.
2. William M. Newman Robert F. Spronill, Principles of interactive computer graphics, McGraw-Hill.

Reference Books:

1. Steven Harington, Interactive Computer Graphics, PHI.
2. Dabod G. Rfgers, Procedure elements for Computer Graphics, Tata McGraw Hill.
3. A. Plastick& Gordon Kalley, Computer Graphics- Schaum’s outline series, PHI.
4. Amarendra N Sinha and Arun D Udai, Computer Graphics, Tata McGraw Hill.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2203 **(3L + 1 T hrs./week)**

DATA STRUCTURES

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course emphasizes on the organization of information, the implementation of linear data structures such as linked lists, stacks, queues, and non-linear data structures such as trees, and graphs. This course also explores recursion principles, the close relationship between data structures and algorithms and file systems.

Pre-requisites: Programming concepts and 'C' language.

Learning outcomes: On successful completion of this module, students should be able to

1. Assess how the choice of data structures and algorithm design methods impacts the performance of programs.
2. Choose the appropriate data structure for a specified application.
3. Compare and analyse data structures and associated algorithms.

UNIT – I

Linear data structure [9 hrs.]
Arrays, Multidimensional arrays, Sparse matrix, Stacks, Queues and Circular queues, De-queues, Linked linear list, Doubly linked list, Circular linked list, Header linked list.

Non-Linear Data Structures [9 hrs.]
Binary trees, Complete binary trees, Extended binary trees, 2 trees, Traversing binary trees, Pre-order, In-order, Post-order, Infix, Prefix, Postfix notations, Header nodes, In-order threading, Threaded binary search trees- searching, Inserting and deleting, Heaps- inserting, Deleting, Huffman's algorithm, B Tree, B+ Tree, B* Tree, AVL Tree, Height balanced tree.

UNIT – II

Searching and Sorting [8 hrs.]
Linear search, Binary search, Insertion sort, Bubble sort, Quick sort, Heap sort, Bucket sort, Radix sort, Shell sort, Exchange sort, Selection sort, K-way merge sort.

Graphs and their representation [8 hrs.]
Introduction, Graph theory terminology, Directed graphs, Sequential representation of graphs, Adjacency matrix, Path matrix, shortest path algorithm, traversing a graph: Breadth-First search, Depth-First search, Linked representation of graph.

Files [6 hrs.]
Queries and sequential organization, Cylinder surface indexing, Hashing, Hashed indexed, Tree indexing, True indexing, Sequential file organizational, Random file organization, Hashed file organization, Inverted files, Cellular partitions.

Text Books:

1. Horowitz & Sahni, Fundamentals of Data Structures, Galgotia.
2. Seymour Lipschutz, Data Structures, Schaum Series

Reference Books:

1. Aaron M.Tanenbaum, Data Structures using C and C++, PHI
2. D. Samanta, Classic Data Structures, PHI
3. G A V Pai, Data Structure and Algorithms, Tata McGraw Hill.
4. Sanjay Pahuja, A Practical Approach to Data Structures and Algorithms, New Age International.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2204 **(3L + 1 T hrs./week)**

OBJECT ORIENTED PROGRAMMING WITH C++

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course introduces C++ as an object-oriented programming language, building on existing knowledge of C and Java. This also covers the C++ language with a focus on its object-oriented features, and how these can be implemented as part of program designs and implementation. You will also study and gain practical experience with the implementation issues related to object-oriented techniques, be able to build good quality software using object-oriented techniques, and understand the role of patterns in object-oriented design.

Pre-requisites: Computer Programming concepts

Learning outcomes: At the completion of this unit students will be able to

1. Understand object-oriented programming features in C++
2. Apply these features to program design and implementation
3. Understand object-oriented concepts and how they are supported by C++
4. Gain some practical experience of C++
5. Understand implementation issues related to object-oriented techniques
6. Build good quality software using object-oriented techniques
7. Understand the role of patterns in object-oriented design.

UNIT – I

The Basic concepts of OOP [4 hrs.]
Literal, Constant, Variables, Pointer types, String types, Constant qualifier, Reference types , The bool type, Enumeration types, Array types, typedef names, Volatile qualifier, Class types. Expressions, Operators, Conditional operator, Bitwise operator, Precedence, Type conversions. Statements
.

Procedural based programming [5 hrs.]
Overview, Function prototype, Argument passing, Returning a value, Recursion, Inline functions, Linkage directives: Extern C, Scope: Global objects and functions, Local objects, dynamically allocated objects.

Object based programming [6 hrs.]
Classes definition , Class object, Class member functions, Constructors and destructors, Friend function and classes, Static class members, Structures and unions, Bit field, Class scope, Nested classes, Structured programming, Object Oriented Programming paradigm.

Overloaded functions/Operators (Polymorphism) [5 hrs.]
Overloaded declarations, the three steps of overload resolution, Argument type conversions, Overloading operators: - operators like =, ( ), ->, <<, >>, + and --.

UNIT-II

Generic programming using template function & template classes [4 hrs.]
Definition of generic programming, Function template, Template arguments, Overloading function template.

Inheritance and IO-Stream library [4 hrs.]
Inheritance, Base class member access, Inheritance types, Protected mechanism, Virtual base classes.

Virtual functions (Run-Time Polymorphism concept) [4 hrs.]
Virtual functions, Pure Virtual functions, Early Vs Late Binding.

The IO-Stream library [4 hrs.]
Streams, Stream classes, Managing unformatted and formatted I/O operations, Manipulators.

Files/Exception handling [4 hrs.]
Classes for file stream operations, Accessing files, Sequential I/O operations, Random access, Command-Line arguments, Exception handling (throw, try, and catch).

Text Books:

1. Herbert Schildt, The Complete Reference C++, Tata Mcgraw Hill
2. E. Balagurusamy, Object Oriented Programming with C++, Tata Mcgraw Hill

Reference Books:

1. Bjarne Stroustrup, C++ programming language, Addison Wesley
2. Stanley B Lippman and Lajoie, C++ Primer, Addison Wesley
3. SauravSahay, OOP with C++, Oxford
4. B.L. Juneja and Anita Seth, Programming with C++, New Age International

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2205 **(3L + 1 T hrs./week)**

COMPUTER ORGANIZATION AND ARCHITECTURE

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: The main objectives of this course are to develop an understanding of the functional blocks of a computer and the inter-relation between them. Covered in-sync with the course on Microprocessors, these two courses enables the learner to co-relate and build a complete picture of the machine and its working under microprocessor and multiprocessor systems..

Pre-requisites: Digital Circuits & Logic Design, PC hardware & peripherals.

Learning outcomes: On successful completion of course, the learner will be able to

1. Understand the computer organization and architecture.
2. Understand computer hardware components. The CPU, memory, I/O devices.
3. Write and debug simple programs written in assembly language.
4. Interpreted data expressed in binary, decimal, and hexadecimal
5. Program input/output, arithmetic operations, decisions, and repetitions in assembly
6. Use assembly language procedures and macros
7. Understand 16-bit addressing
8. Use bit operations and perform array processing

UNIT-I

Basic structure of computers [9 hrs.]
Functional units, Basic operational concepts, Bus structures, Memory locations, Encoding of information, Main memory operations, Addressing modes, Program sequencing, Basic Input-Output operations, Memory reference, Register reference, Input-Output reference instructions, Register stack, Memory stack, Subroutines.

.

The Processing Unit [9 hrs.]
Fetching a word from memory, Storing a word in memory, Register transfers, Performing an arithmetic or logic operation, Instruction format–Three, Two, One and Zero address instruction, Execution of a complete instruction, Control Unit-hardwired control, Micro programmed control.

UNIT –II

Arithmetic [7 hrs.]
Number representation, Fixed point addition and subtraction, Design of fast adders, Multiplication of fixed point numbers, Booth's multiplication, Integer division, Floating point numbers and operations.

Memory organization [7 hrs.]
A review of random and serial access memories, Basic concept of main memory-static and dynamic memory, Memory hierarchies-cache memory, Virtual memory, Logical to physical memory mapping.

Input and output organization [8 hrs.]

Accessing I/O devices interrupt, Enabling and disabling interrupts, Handling multiple devices, Controlling device requests, Direct memory access, I/O hardware- processor bus, Interface circuits, Standard I/O interfaces.

Text Books:

1. V. C. Hamacher, Zaky, Vranesic, Computer Organization, Tata McGraw Hill.
2. Morris Mano, Computer System Architecture, PHI.

Reference Books:

1. J. P. Hayes, Computer Architecture & Organization, Tata McGraw Hill.
2. P .Pal Choudhury, Computer Designing & Architecture, PHI.
3. William Stallings, Computer organization & Architecture – Designing for Performance, Pearson Education.
4. D. A. Patterson and J. L. Hennessy, Computer Organization and Design-The Hardware/ Software Interface, Morgan Kaufmann.

CA 2261 **(3 hrs./week)**

COMPUTER GRAPHICS LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA2202 - Computer Graphics and associated prerequisites.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2262 **(3 hrs./week)**

DATA STRUCTURES LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA2203 – Data Structures and associated prerequisites.

CA 2263 **(3 hrs./week)**

C++ LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA2204 - Object Oriented Programming with C++ and associated prerequisites.

DEPARTMENT OF COMPUTER APPLICATIONS

MA 2308 **(3L + 1 T hrs./week)**

NUMERICAL ANALYSIS – A SOFTWARE APPROACH

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: The student will learn to formulate ordinary differential equations (ODEs) and seek understanding of their solutions, obtained either exactly or approximately by analytic methods. Students will have the concept of a solution to an initial value problem, and the guarantee of its existence and uniqueness under specific conditions. The student will recognize basic types of differential equations, which are solvable, and will understand the features of linear equations in particular. Fourier series is helpful for solving ordinary and partial differential equation. The purpose of Laplace transforms in simple words. Laplace transform converts an arbitrary signal into exponentials while Fourier transform converts a given signal into sinusoids.

Pre-requisite: Basic concept of calculus, matrix algebra, ordinary differential equations, linear algebra.

UNIT- I

Interpolation [5 Hrs.]
Concept of Finite Difference, Finite difference operators and their use to derive interpolation formulae. Newtonian polynomial interpolation formulae (Forward and Backward interpolation algorithms and examples.), Newton’s divided difference (algorithm and example), Lagrange's interpolation formula (algorithm and example), Inverse interpolation. Numerical differentiation.

Numerical Integration [5 Hrs.]
Derivation of Newton’s Cote’s Quadrature formula. Trapezoidal rule, Simpson's one third and three eighth rule (Derivation and examples and algorithms).

Solution of Transcendental Equation [5 Hrs.]
Numerical solution of algebraic and transcendental equations using method of ordinary iteration. Regula Falsi and Newton Rapson’s method (Simple example and algorithms in all the cases).

Solution of System of Linear Equations [5 Hrs.]
i) Direct Method: Gauss Elimination and LU decomposition method,
(ii) Iterative Method: Jacobi, Gauss –Seidal method. Method for Matrix Inversion. (Examples and algorithms for all the cases).

UNIT -II

Solution of Differential Equations [10 Hrs.]
Numerical solution of initial value problems in ordinary differential equations by Taylor series method, Picard's method, Euler's method, Modified Euler's method, Runge Kutta fourth order method, Millne's Predictor and Corrector method. Finite difference method for the solution of boundary value problems in linear ordinary differential equations using central differences. (Examples and algorithms for all the cases).

Integral Equations [5 Hrs.]

Problem definition, quadrature method for solving linear Fredholm integral equation. (Examples and algorithms for all the cases).

Eigenvalue Problems [5 Hrs.]

Problem definition, Fadeev Leverrier method for characteristic equation, similar matrices, bounds on eigenvalues, power method. (Examples and algorithms for all the cases).

Text Book:

1. Jain, Iyenger and Jain, Numerical Methods for Scientists and Engineers, New Age International.
2. Johnston, Numerical Analysis- A software Approach, Wiley.
3. S. S. Sastry, Introductory methods of numerical analysis, PHI.

Reference Books:

1. V. Rajaraman, Computer Oriented numerical methods, PHI.
2. Sen and Krishnamurthy, Numerical Algorithms, Oscar.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2302 **(3L + 1 T hrs./week)**

SYSTEM PROGRAMMING AND MICROPROCESSOR

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: In this course, the emphasis is given on resource management and performance of the systems. Study of programming language translation: functions and general organization of compiler design and interpreters are discussed. It also highlights the basics of microprocessors.

Pre-requisites: Data Structures, Digital logic, Programming Language concept and Computer Organization and Architecture.

Learning outcomes: On successful completion of this module the learners will be able to understand

1. The machine structure of a microprocessor (IBM3622)
2. Design of assembler, linker, loader and compiler
3. Macro language to implement macro processor
4. The architecture and operation of a microprocessor system (Intel 8085)
5. Interfacing

UNIT-I

Introduction [6 hrs.]
Machine structure, evolution of the components of a programming system, evolution of Operating Systems, Operating System user viewpoint functions, Operating System user viewpoint functions, Operating System user view point batch control language, Operating System user viewpoint facilities.

General Machine structure and machine Language [5 hrs.]
General Machine Structure, Machine language, Assembly language (IBM-360).

Assemblers [5 hrs.]
Assembler basics, General Design Procedure, Design of assembler.

Macros [5 hrs.]
Macros language and Macro processor: Macro Instruction Arguments, conditional macro expansion, Macro calls with in macros, Macro Instructions defining macros, A two pass macro processing algorithm.

UNIT-II

Loaders [6 hrs.]

Loaders, loader Schemes, Compile and go loaders, General loader Scheme-absolute loaders, Subroutine linkages, Relocating loader, Loader schemes binders, Linking loaders, Overlays, Dynamic binders, Design of an absolute loader.

Compilers [5 hrs.]

Introduction to compilers, Recognizing basic units, Recognizing synthetic units and interpreting meaning, Intermediate form generation: Arithmetic statements, Non-arithmetic statements, Non-executable statements, Storage allocation, Code generation.

Microprocessor [4 hrs.]

Introduction to 8085 CPU architecture – register, Organization, 8085, Instruction set, Addressing modes, Instruction cycle, Machine cycle, Timing diagrams, Programming using 8085 instruction set.

Hardware Interfacing [4 hrs.]

Interfacing memory; Interfacing I/O-memory mapped I/O, and I/O mapped I/O. Interrupts, introduction to DMA(with reference to 8085 CPU).

Text Books:

1. John Donovan, System Programming, Tata McGraw Hill.
2. Ramesh S. Gaonkar, Microprocessor architecture, programming and applications with 8085/8080A, Penram International Publishing.

Reference Books:

1. D.M. Dhamdhere, Systems Programming and Operating Systems, Tata McGraw Hill.
2. Aho, Ulmann, Sethi, Compiler Design, Pearson.
3. Srimanta Pal, Systems Programming, Oxford.
4. Leland L Beck and D. Manjula, System Software, Pearson.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2303 **(3L + 1 T hrs./week)**

DATABASE MANAGEMENT SYSTEM

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course provides the basic information about relational Database Management System and their development. The major objectives of the course is to provide an introduction of DBMS and their use, be familiar with the basic DBMS architecture, components, and interfaces, have experience using at least one modern Database Management System, understand and use database models in database and application design

Pre-requisites: Programming Concepts.

Learning outcomes: At the end of this subject, learners should be able to

1. Describe fundamental elements of a relational database management system
2. Explain the basic concepts of relational data model, Entity-Relationship model, relational database design, relational algebra and database language SQL
3. Design Entity-Relationship diagrams to represent simple database application scenarios
4. Map Entity-Relationship diagrams into relational Schema, populate a relational database and formulate SQL queries on the data

UNIT –I

Introduction [7 hrs.]
Introduction to Database Management System, Characteristics, Advantages, Implications of the database approach, Database system concepts and architecture, Database models, Schemas and Instances, Database languages and interfaces, Classification of database management systems.

ER models, Relational Model and Relational algebra [8 hrs.]
Entity-Relationship model, Models for database design, Entity types, Entity sets, Attributes and keys, Relationships, Relationship types, Roles and structural constraints , Dealing with constraint violations. Relational data model, Relational algebra, Basic relational algebra operations, Relational database design using ER-to- Relational Mapping.

Database Design [8 hrs.]
Functional dependencies and Normalization for relational databases, General definitions of third normal forms, Boyce-Codd normal form relational database design algorithms and further dependencies, Algorithm for relational database schema design, Multivalued dependencies and forth normal form.

UNIT –II

SQL the relational database standard [7hrs.]

Data definition, Constraints, and schema changes in SQL 2 , Basic queries in SQL- Insert , Update and Delete statements in SQL, Specifying general constraints as assertion, Additional features of SQL.

Transaction Processing [5 hrs.]

Introduction to transaction processing and system concepts, Desirable properties of transactions, Schedules and Recoverability, Serial-ability, Transaction support in SQL.

Concurrency Control Techniques [5 hrs.]

Locking techniques for concurrency control, Concurrency control based on time stamp ordering, Multi version concurrency control techniques, Validation(optimistic) Concurrency for concurrency control in indexes, Some other concurrency control issues, Recovery concepts, Recovery techniques based on deferred update, Recovery technique base on immediate update, shadow paging.

Text Books:

1. Elmasri and Navathe, Fundamentals of database systems, Addison Wesley
2. Silberschatz, Korth, Sudarshan, Database System Concepts, McGraw-Hill Higher Education.

Reference Books:

1. Thomas Connolly, Carolyn Begg, Database Systems – A Practical Approach to Design, Implementation, and Management, Pearson Education.
2. Jefrey D.Ulman, Jenifer Widom, A First Course in Database Systems, Pearson Education.
3. Bipin C Desai, An Introduction to Database Systems, Galgotia.
4. AtulKahate, Introduction to Database Management systems, Pearson

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2304 **(3L + 1 T hrs./week)**

OPERATING SYSTEMS

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: The principles and concepts that govern the design of modern computer operating systems are studied. Managing computing resources such as the memory, the processor and the Input/output devices are covered. Algorithms for CPU scheduling, memory and general resource allocation; process coordination and management; deadlocks and memory management techniques; case studies of several operating systems are also covered.

Pre-requisites: Programming language concepts.

Learning outcomes: On successful completion of course, the learner will be able to

1. Explain the evolution, objectives and functions of modern operating systems
2. Analyse the trade-offs inherent in operating system design
3. Discuss networked, client-server, distributed operating systems and how they differ from single user operating systems
4. Describe the influences of open source software, the internet on the operating system design.
5. Identify potential threats to operating systems and the security features to counter the threats.

UNIT – I

Introduction [4 hrs.]
Simple batch systems, Multi-programmed batch system, Time sharing, Parallel, Distributed system, Real time system.

Processes and CPU scheduling [6 hrs.]
Process concepts, Scheduling, Operation on process, Cooperating processes, Threads, Scheduling criteria, Scheduling algorithms.

Process Synchronization [5 hrs.]
Critical section problems, Synchronization Hardware, Semaphores, Classical problems of synchronization.

Deadlock [5 hrs.]
Deadlock characterization, Deadlock prevention, Avoidance, Detection and recovery.

UNIT – II

Memory Management [5 hrs.]
Logical versus physical address space, Swapping, Contiguous allocation, Paging, Segmentation.

Virtual Memory [6 hrs.]
Introduction, Demand paging, Page replacement, Page replacement algorithms.

File System Interface [5 hrs.]

File concepts, Access methods, File Protection.

The UNIX Operating System [4 hrs.]

Design principles, Programmer & user interface, Process management, Memory management, File system.

Text Books:

1. Silberschatz and Galvin, Operating System concepts, Wiley.
2. Achynt Godbole, Operating Systems, Tata McGraw-Hill.

Reference Books:

1. System Programing by J. J. Donovan, Tata McGraw Hill.
2. S. E. Madwick & J. J. Donovan, Operating Systems, Tata McGraw-Hill.
3. H. M. Diatel, An introduction to Operating System, Wiley
4. Chales Crowley, Operating Systems- A design- oriented approach, Tata McGraw-Hill.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2305 **(3L + 1 T hrs./week)**

JAVA PROGRAMMING

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: The Java Programming Language course provides students with a solid foundation for programming with JAVA. It also highlights the creation of graphical user interfaces (GUIs), exceptions, file input/output (I/O), and threads; and network programming.

Pre-requisites: Object Oriented Programming.

Learning outcomes: On successful completion of course, the learner will be able to

1. Develop object oriented programs using Java
2. Develop quality software's using Java
3. Develop network programs using Java

UNIT-I

Introduction to Java [4 hrs.]
Evolution and features of java, Overview of java, Two control statements, Lexical issues, Data types, Variables and arrays, Literals, Variables, Type conversion and casting, Type promotion in expression, Arrays, Operators, Bitwise operators, Relational operators, Boolean and logical operators, Assignment Operators, The '?' operator, Operator precedence, JAVA statements.

Introducing classes [4 hrs.]
Class fundamentals, Declaring objects, Assigning object, Reference Variables, Introducing methods, Constructors, 'this' keyword, Garbage collection, The finalize() method, stack class.

Methods and classes [4 hrs.]
Overloading methods and constructors, Using object as parameters, Argument passing, Returning objects, Recursion, Access control, Static methods, Nested and inner classes, Command line argument.

Strings handling [4 hrs.]
String constructors, String length, Special string operators, Character extraction, String comparison, String searching, String modification, Changing case of characters within a string, Compression and String buffer, String builder.

Inheritance, Interfaces and Packages [4 hrs.]
Basics of inheritance, Types of inheritance, Using super keyword, Method overriding, Dynamic method dispatch, Abstract class, Using final with inheritance, The object class, Defining and implementing interfaces, Extending interfaces, Nested interfaces, Applying interfaces, Defining and creating packages, Access protection, Importing packages.

UNIT – II

Exception Handling [3 hrs.]

Fundamentals, Exception types, Uncaught exceptions, Using try and catch, Multiple catch clauses, Nested try statements, Throws, Finally, Java’s built in exceptions, Creating own exception classes.

Input/Output and file handling [4 hrs.]

Java I/O classes and interfaces, The stream classes, Byte streams, The character streams, The console class, File class, Byte-stream class, Random access files

.

Multithreaded programming [4 hrs.]

Thread basics, Java’s thread model, Thread priorities, Synchronization, Messaging, Thread class and runnable interface. The main thread, creating a thread, Creating multiple threads, Inter-thread communication, Suspending/resuming and stopping threads

.

Network programming [5 hrs.]

Networking basics, The networking classes and interfaces, The Inet Address class, Inet4Address, TCP socket, URL, URL Connection, HTTP/URL Connection, TCP/IP server sockets, Datagram Socket and Datagram Packet.

Event Based Programming [4 hrs.]

The applet class, Repaint(), The HTML applet tag, Passing parameter to applet, Event handling, Using delegation event model, Abstract Window Toolkit, AWT classes, Windows fundamentals, Working with frame windows, Creating a windowed program, Displaying information within a window, AWT controls,

Text Books:

1. E. Balagurusamy, Programming With JAVA, Tata McGraw Hill.
2. Herbert Schildt, Java: The Complete Reference, Tata McGraw Hill.

Reference Books:

1. K. Arnold and J. Gosling,The Java Programming Language, Sams Publishing.
2. Allamaraju, Professional java Server Programming, Wrox Press.
3. Patrick Naughton and Herbert Schildt, JAVA2: The Complete Reference, Tata McGraw Hill.
4. R. Krishnamoorthy and S. Prabhu, Internet & Java Program, New Age International.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2361 **(3 hrs./week)**

OPERATING SYSTEMS LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA2304 - Operating System and associated prerequisites.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2362 **(3 hrs./week)**

JAVA PROGRAMMING LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA2305 - Java Programming and associated prerequisites.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2363 **(3 hrs./week)**

DATABASE MANAGEMENT SYSTEM LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA2303 - Database Management System and associated prerequisites.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2306 **(3L + 1T hrs/week)**

STATISTICAL METHODS AND SIMULATION

No. of questions to be set: 4 each from UNIT I and UNIT II.
No. of questions to be answered: Any Five selecting at least TWO from each UNIT.

Objective: The courses aims at teaching the basics simulation system and to make them familiar with the mathematical concepts that are used in simulation. This course helps the students to understand the basic concepts of computer-based modeling and simulation applicable to various domains of computer science.

Pre-requisite: Basic programming concepts.

UNIT I

System Models [4 Hrs.]
The concepts of systems, system environment, Continuous and discrete system, system modeling, different types of models, principle used in modeling.

Introduction to simulation [6 Hrs.]
What is simulation, need of simulation, when to use simulation, Advantages and Disadvantages of simulation, areas of applications, and steps in simulation study.

Probability in simulation [10 Hrs.]
Basic concepts and definition of probability axioms, Laws of probability(based on set theory concepts), Events and their Probabilities distribution, Conditional probability Bayes' theorem, Normal Distribution, Poisson Distribution, Problems and applications.

UNIT-II

Statistics in Simulation [8 Hrs.]
Mean, standard deviation, coefficient of variation, skewness & kurtosis, Carl Pearson Correlation, Rank correlation and illustrated examples.

Random variable and Expectation [7 Hrs.]
Discrete and continuous random variables, expectation of random variables, the Poisson random variable use of random numbers in simulation.

Sampling distribution [5]
Introduction, sampling plan and simulation, statistics and sampling distribution.

Text books:

1. J. Banks, J. S. Carson II and B. L. Nelson, Discrete-Event System Simulation, Prentice Hall.
2. Sheldon Ross, A First course in Probability, Pearson.

References:

1. Banks J, Handbook of Simulation, Wiley.
2. Geoffrey Gardon, System Simulation, PHI.
3. Mendenhall Beaver, Probability and Statistics, Thomson.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2401 **(3L + 1 T hrs./week)**

SOFTWARE ENGINEERING

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course will offer a wide perspective on software development covering the full life cycle of software development. This would be inclusive of requirements analysis, technical design, estimating, programming style, testing, quality measures and management issues. Current software engineering theory and practices with special emphasizes on the methodologies, techniques and tools are also taught.

Pre-requisites: Algorithm concept and Database Management System.

Learning outcomes: At the end of the course, the student will be able to

1. Describe principles, concepts and practice of software engineering and apply its techniques and tools.
2. Author a software requirements and specifications for a software system.
3. Understand the distributed system architectures and application architectures.
4. Understand the differences between real-time and non-real time systems.
5. Demonstrate proficiency in rapid software development techniques and cost estimation.
6. Author a software testing plan.
7. Manage a project including planning, scheduling and risk assessment/ management.

UNIT – I

Introduction [2 hrs.]
The software engineering discipline-evaluation and impact, Programs vs. Software products, Why study of software engineering?, Emergence of software engineering, Notable changes in software development practice, Computer system engineering.

Software life cycle [6 hrs.]
Life Cycle Models: Classical waterfall model, Iterative waterfall model, Prototype model, Evolutionary model, Spiral model, Comparison of different life cycle models.

Software project management [4 hrs.]
Responsibilities of project manager, Project planning, Metrics for project size estimation, Project estimation techniques, Empirical estimation techniques, COCOMO: A heuristic estimation techniques, Halstead's software science: An analytical technique, Staffing level estimation, Scheduling, Organization and team structure, Staffing, Risk management, Software configuration management, Miscellaneous plans.

Requirements analysis and specification [2 hrs.]
Requirements gathering and analysis, Software requirement specification (SRS), Formal system development techniques, Axiomatic specification, Algebraic specification.

Software design [4 hrs.]
Cohesion and coupling, Neat arrangement, Software design approaches, Object oriented vs. function oriented design.

UNIT – II

Function-oriented software design [4 hrs.]
Overview of SA/SD methodology, Structured analysis, Data flow diagram, Extending DFD technique to real life systems, Structured design, detailed design, Design review.

User interface design [4 hrs.]
Characteristics of a good user interface, Basics concepts, Types of user interfaces, Component-based GUI development, User interface design methodology, GUI design methodology, Task and object modelling, Interaction design and rough layout, User interface inspection.

Coding and testing [4 hrs.]
Coding, Code review, Testing, Testing in the large vs. testing in the small, Unit testing, Black-box testing, Debugging, Program analysis tools, Integration testing, System testing, Some general issues associated with testing.

Software Reliability and Quality Management [4 hrs.]
Software reliability, Statistical testing, Software quality management system, ISO 9000, SEI capability maturity model, Personal software process (PSP), Six sigma.

Computer Aided Software Engineering [2 hrs.]
Case and its scope, Case environment, Case support in software life cycle, Other characteristics of case tools, Towards second generation case tool, Architecture of a case environment.

Software maintenance [2 hrs.]
Characteristics of software maintenance, Software reverse engineering, Software maintenance processes model, Estimation maintenance cost.

Software reuse [2 hrs.]
Basics issues in any reuse program, A reuse approach, Reuse at organization level.

Text Book:
1. Rajib Mall, Fundamentals of Software Engineering, PHI.
2. Richard Fairley, Software Engineering Concepts, Tata McGraw Hill.

Reference Books:
1. Jalote Pankaj, An integrated approach to Software Engineering, Narosa.
2. Pressman R, Software Engineering- Practioner Approach, Tata McGraw Hill.
3. Somerville, Software Engineering, Pearson.
4. Budgen, Software Design, Pearson.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 24** **(3L + 1 T hrs./week)**

ELECTIVE - I

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives and Pre-requisites are given under Elective course details

TO BE CHOSEN/OFFERED FROM THE LIST OF ELECTIVES FOR FOURTH SEMESTER.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2403 **(3L + 1 T hrs./week)**

ASP .NET APPLICATION

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: It covers the technique of creating ASP.NET applications that delivers dynamic content to the web. In the course students will learn how to create a web form with server controls, separate page code from content with code-behind pages, page controls, and components, display dynamic data etc.

Pre-requisites: Object oriented programming and Web Page Design.

Learning outcomes: At the end of the course the students will be able to

1. Define the structure and create a new ASP.NET web page
2. Utilize master pages and CSS
3. Build client rich web pages using server controls
4. Manage view, session, and application state data of ASP.NET webpages
5. Utilize ADO and ADO controls
6. Incorporate web site navigation and application security

UNIT- I

The .NET Framework [4 hrs.]
Introduction and features of .NET Platform, Components of .NET architecture, Assemblies overview, Common Language Runtime, .NET class library.

Learning the .NET Languages [4 hrs.]
Basic difference between C# and VB.NET, Data Types, Initializers, Arrays, Enumerations, Scope and accessibility, Variable operations, Conditional structures, Loop structures, Functions and Subroutines.

Types, Objects and Namespaces [4 hrs.]
Shared members of a class, Adding properties, A basic method, Event, Constructors, Value Types, Reference Types, Inheritance, Shared members & Casting, importing namespaces, Assemblies.

ASP .NET Applications [4 hrs.]
ASP .NET File types, Bin directory, Web form inheritance explained, Compiled code behind files, Compiling multiple files into one assembly, Importing namespaces, Global.aspx code-behind, Web. Config. files.

WEB form Fundamentals [3 hrs.]
Problem with response. Write, Different server controls, View State, HTML control classes, Page classes.

Validation and Rich Controls [3 hrs.]

Validation controls, Validation process, Validator classes, Regular expressions.

UNIT-II

State Management [5 hrs.]

The Problem of state, Viewstate, Query String, Cookie, Session state configuration, Application state.

Error Handling [4 hrs.]

Common errors, Handling and creating exceptions, Throwing exception.

Data Access using ADO.NET [5 hrs.]

Characteristics of ADO.NET, Comparing ADO and ADO.NET, Creating a connection, Select command, Using a command with a data reader, Updating data, Accessing, modifying, updating disconnected data, Selecting multiple tables.

Files, Streams and Email [4 hrs.]

Getting file system information in ASP.NET, Reading, writing files with streams, File Uploading, Sending mail inASP.NET.

Text Books:

1. Mathew Mac Donald, ASP.NET: The complete reference, Tata McGraw Hill.
2. Walther Stephen, Teach Yourself ASP. Net 2.0 unleashed, Sams Publishing.

Reference Books:

1. C. Ullman, Beginning ASP NET, Wiley.
2. Evjen Bill, Professional ASP.NET 2.0, Wiley.
3. Buczek Greg, Asp.Net: Web Developer's guide, Tata McGraw Hill.
4. Paul Wilton, Iman Spaanjaars, Asp.Net 2.0 Instant Results, Wiley.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2404 **(3L + 1 T hrs./week)**

COMPUTER NETWORK

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course emphasizes on several computer network concepts, applications and protocols in OSI as well as TCP/IP layered architecture. It also covers the various protocols of various layers, their operations and applications. Further it discusses the concept of network security, challenges and their counter measures.

Pre-requisites: Communication Technique and Data Communication.

Learning objectives: At the end of the course, students will be able to understand

1. The concept of data communication protocols and standards.
2. The concept of open Systems, giving an overview of Transport and Application Support Protocols.
3. The key concepts of protocols and algorithms in networking
4. The complexity of networks, their structure and utility
5. The networking applications, network infrastructure and the network management.

UNIT- I

Introduction [5 hrs.]
Introduction of Computer Network, Network Topologies, Network Reference models –ISO-OSI model, TCP/IP model.
Physical layer-Transmission media, Multiplexing: FDM, TDM, WDM. Switching techniques.

Data link layer and **MAC [8 hrs.]**
Error detection and correction Elementary data link protocol-Stop-and-wait ARQ, Sliding window, Go- back-n, Selective Repeat ARQ. Multiple Access protocol- ALOHA, Slotted ALOHA, CSMA Protocols, Introduction to MAC Protocols - 802.3, 802.4, 802.5, 802.11.

Network layer [7 hrs.]
Network design issue, Routing algorithm-introduction, Optimality principle, Shortest path, Flooding, Distance vector routing, Link State Routing. General principle of congestion control-leaky bucket algorithm, Token bucket algorithm, Internet routing protocols- RIP, OSPF, BGP, ICP.

UNIT-II

TCP/IP [5 hrs.]
TCP/IP Architecture, Internet Protocol (IP), ARP, RARP, DHCP and mobile IP.

Transport layer [6 hrs.]
Transport services, Element of transport protocols, TCP- connection management, transmission policy, congestion control, Timer management. UDP- RPC, RTTP.

Application layer [6 hrs.]

DNS, Email- SMTP, POP3, World Wide Web (WWW)- HTTP, mobile Web, FTP, TELNET, P2P network.

Introduction to Network Security [3 Hrs.]

Cryptography- Cypher, Cypher Modes, Encryption-Decryption, Types of cryptography. Digital Signature, Antivirus and Firewalls.

Text Books:

1. B. Forouzan, Data Communication and Networking, Tata McGraw Hill
2. Tanenbaum A S, Computer Networks, PHI.

Reference Books:

1. Leon-Garcia and Didjaja, Communication Network, Tata McGraw Hill
2. Black Vymos, Computer Networks, PHI.
3. Peterson & others, Computer Networks, New Age International.
4. Choudhury D R., Networks & Systems, New Age International.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2405 **(3L + 1 T hrs./week)**

IT LAW AND PRACTICES

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course covers the ethical legal practice skills and critical understanding of legal practice approaches. It also covers various issues and concepts of ‘reflective practice' (duties to client, confidentiality, conflict of interest) legal and non-legal problems - implications for practice/service provision of an appreciation of the relevant services and the meshing of non-legal issues interviewing routine - difference between legal information and legal advice legal practice approaches - ‘reactive', ‘proactive', ‘preventative', ‘activist' tailoring service models to client needs.

Pre-requisites: Fundamentals of IT and E-commerce.

Learning outcomes: On completion of the course the learners should
1. Have the knowledge of IT security and its related issues.
2. Acknowledge various security threats and perform counter actions.
3. Have the knowledge of Cyber security and practices and their counter measures.
4. Be aware of computer network vulnerabilities.
5. Be able to apply various tools and techniques to assess network security

UNIT – I

Introduction to Information Security [7 hrs.]
History of Information Security Characteristics of Information, NSTISSC Security model, Components of Information System, Security Systems Development life cycle, Security professional. The Information Technology Act, 2000 (No. 21 of 2000). A Methodology for Developing Trusted Information Systems, A national information infrastructure model for information warfare defence. Intellectual Property, Understand how copyrights, patents, trademarks and other forms of IP differ.

Need for Security [7 hrs.]
Business needs, Threats, Attacks. Information Security Risk Management: An Overview of Risk Management, Risk Identification Identifying assets, Threats and vulnerabilities Risk Analysis and Assessment Understanding risk analysis, planning for risk analysis, Performing risk analysis and assessment, Risk Control Strategies and selection.

Cyber Security and practices [6 hrs.]
Security problem in computing: definition of secure, Attacks, the meaning of computer security, computer criminals, methods of defence. Administering security policies, Physical security the economics of cyber security: Making a business case, Quantifying security, Modelling cyber security, current research and future directions Privacy in computing: privacy concepts, privacy principles and policies.

UNIT –II

Planning for Security [8 hrs.]

Information security policy, Standards and practices, the information security blueprint, security education, and Training and awareness program. Implementing Information Security: Project Management for Information Security, Technical topics, Nontechnical aspects. Disaster Recovery and Risk Monitoring: Disaster recovery planning, Disaster recovery plan , Risk monitoring Application-Level Risks the Fundamental hacking concept, Network service vulnerabilities and attacks memory manipulation attacks, Mitigation process manipulation risks.

Network Security Assessment and Tools Required [6 hrs.]

The business benefits, Classifying internet-based attackers, Assessment service definitions, Network security methodology, The cyclic assessment approach, Free and commercial network scanning tools, Protocol- Dependent assessment tools.

Computer Network Vulnerabilities [6 hrs.]

Sources of vulnerabilities, vulnerability assessment, vulnerability identification and assessment. Security Evaluation of Computer Products Product security evaluation process, Computer products evaluation standards, Major evaluation criteria.

Text Books:

1. Michael E.Whitman and Herbert J. Mattord, Principles of information security, Cengage Learning.
2. Chris McNab Network Security Assessment, O'Reilly.

Reference Books:

1. John W. Rittinghouse and William N. Hancock, Cyber Security Operations handbook, Digital Press.
2. The Information Technology 2000(No. 21 of 2000) -Bare Act
3. Lech J. Andrew M. Colarik, Cyber Warfare and Cyber Terrorism, IGI Global.
4. Charles P.Pfleegar and Shari Lawrence Plfeeger Security in Computing, PHI.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2461 **(3 hrs./week)**

SOFTWARE ENGINEERING LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA2401 - Software Engineering and associated prerequisites.

CA 2462 **(3 hrs./week)**

COMPUTER NETWORK LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA2404 - Computer Network and associated prerequisites.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2463 **(3 hrs./week)**

ASP .NET APPLICATION LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA2403 - ASP. NET Application and associated prerequisites.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2504 **(3L + 1 T hrs./week)**

FORMAL LANGUAGES AND AUTOMATA THEORY

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: The central objective of the course is to provide learners with a detailed understanding of the mathematical models of the machines and their evolution through requirement generation and advancement in languages.

Pre-requisites: Computer Programming concepts and Discrete Structures for Computer Science.

Learning outcomes: By the completion of the course, the students will be able to

1. Define a system and recognize the behaviour of a system. They will be able to minimize a system and compare different systems.
2. Convert Finite Automata to regular expression.
3. Check equivalence between regular linear grammar and FA.
4. Minimize context free grammar and check equivalence of CFL and PDA.
5. Design Turing Machine.

UNIT-I

Introduction [4 Hrs.]
Mathematical preliminaries: Sets, Logic, Functions, Relations, Languages.
Definitions: Language, Grammar, Automata, Relation between language, Grammar and automata, Importance of automata theory.

Finite automata [5 Hrs.]
Informal introduction: Drawing examples from everyday life to bring out the essence of finite automata, Finiteness and its importance in automata theory.
Deterministic finite automata: Definition, Processing strings, Transition functions, Language of a DFA, Nondeterministic finite automata: Non-determinism, Definition, Extended transition functions, Language of a NFA, Equivalence of DFA and NFA, Kleene's theorem, Epsilon transitions, Applications of Finite automata in text search.

Regular expressions and regular languages [4 Hrs.]
Memory required to recognize a language, Regular expressions, Regular expression to finite automata, Finite automata to regular expression, Algebraic laws for regular expressions, Applications of regular expressions, Criterion for regularity, Regular languages.

Properties of regular languages [3 Hrs.]
Pigeonhole principle, Pumping lemma for regular languages, Closure properties, Testing membership of regular languages, Equivalence of automata.

Context Free Grammars and Languages [4 Hrs.]

Definition, Leftmost and rightmost grammars, Parse trees, Ambiguity: Ambiguous grammar, Removing ambiguity. Normal forms, Applications of context free grammars: Parsers.

UNIT-II

Pushdown automata and context free languages [7 Hrs.]

Definition of pushdown automata, Representing pushdown automata, Acceptance by pushdown automata: By final state, By empty stack, Deterministic pushdown automata, Equivalence of pushdown automata and context free grammars, Pumping lemma for context free languages, Closure properties of context free languages, Testing membership of context free, Decision problems for context free languages.

Turing machines [6 Hrs.]

Definition, Language of a Turing machine, Programming Turing machines, The Church-Turing thesis, A simple programming language, Extensions of the basic Turing machine.

Recursively enumerable languages [2 Hrs.]

Definition, Enumeration, Chomsky hierarchy.

Undecidability [3 Hrs.]

The halting problem, The post correspondence problem, Time and space complexity of Turing machines, Complexity classes.

Language learning [2 Hrs.]

Learning framework, Inductive inference, Grammar induction.

Text Books:

1. John. E. Hopcroft, Rajeev Motwani, Jeffry. Ullman, Introduction to Automata Theory, Languages and Computation, Pearson Education.
2. John Martin, Introduction to Languages and the Theory of Computation, Tata McGraw Hill.

Reference Books:

1. Peter Linz, An Introduction to Formal Languages and Automata, Narosa.
2. James. L. Hein, Discrete Structures, Logic and Computability, Narosa.
3. Partha Niyogi, The Computational Nature of Language Learning and Evolution, PHI.
4. Zvi Kohavi and Niraj K. Jha, Switching and Finite Automata theory, Tata McGraw Hill.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2502 **(3L + 1 T hrs./week)**

ANALYSIS AND DESIGN OF ALGORITHMS

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course builds upon preliminary knowledge delivered in Data Structures. The main objectives of the course are to provide thorough knowledge and understanding of different algorithm analysis techniques, design strategies and their applications. Special purpose machines, some critical problems and innovative techniques are used in solving them.

Pre-requisites: Data Structures and Programming concepts

Learning objectives: On successful completion of course, the learners will be able to

1. Describe, apply and analyse the complexity of certain divide and conquer, greedy, and dynamic programming algorithms.
2. Identify and analyse criteria and specifications appropriate to new problems, and choose the appropriate algorithmic design technique for their solution.
3. Describe the classes P, NP, and NP-Complete and be able to prove that a certain problem is NP- Complete.
4. Explain and apply backtracking and branch and bound techniques to deal with some hard problems.

UNIT-I

Introduction [3 hrs.]
Algorithms, Insertion Sort, Pseudo code conventions, Analysing algorithms, Analysis of insertion sort, Worst case and average case analysis, Order of growth, Divide-and-Conquer approach, Analysis of merge sort, Analysing Divide-and-Conquer algorithms.

Growth of Functions [4 hrs.]
Asymptotic notation: Θ-notation, O-notation, Ω-notation, Asymptotic notation in equations. Standard notation and common functions: Monotonicity, Floors and ceilings polynomials, Exponentials, Logarithms, Factorials.

Heapsort [3 hrs.]
Heaps, Maintaining the heap property, Building a heap, The heap sort algorithm.

Quicksort [5 hrs.]
Description of Quicksort, Performance of Quicksort, Worst case, best case and balanced partitioning. Intuition for the average case. Analysis of quicksort: Worst case analysis, Average case analysis: Analysis of partitioning, A recurrence for the average case.

Sorting in linear time [4 hrs.]
Counting sort, Radix sort, and Bucket sort algorithms.

Graph Representation [4 hrs.]

Properties of graphs and graph traversal algorithms, BFS and DFS, Connected components and spanning trees, Bi-connected components and DFS.

UNIT-II

Binary Search Trees [4 hrs.]

Introduction, Querying a binary search tree, Insertion and deletion, Optimal binary search trees, Dynamic programming approach.

Dynamic Programming [5 hrs.]

Introduction, Matrix chain multiplication, Counting the number of parenthesization, An optimal parenthesization, Recursive solution, Computing the optimal cost, Constructing an optimal solution.

Greedy Algorithm [4 hrs.]

Introduction, Greedy vs dynamic programming. Huffman codes: Prefix codes, Constructing a Huffman code, Knapsack problem, Optimal merge patterns.

NP Completeness [4 hrs.]

Basic Concepts, Cook's theorem statement.

Text Book:

1. Coreman, Liesorson, Rivest, Design and Analysis of Algorithms, PHI.
2. Horowitz and Sahini, Fundamental of Computer Algorithms, Galgotia.

Reference Books:

1. Hopcroft and Ullman, Design and Analysis of Algorithms, Wesley.
2. Sedgewick, Analysis of algorithms, Pearson.
3. A.V Aho, Design and analysis of computer algorithms, Wesley.
4. Horowitz, Fundamentals of computer algorithms, Galgotia.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2503 **(3L + 1 T hrs./week)**

OBJECT ORIENTED DESIGN AND ANALYSIS USING UML

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course delves into the processes of both object-oriented analysis and object-oriented design using UML as the notation language to provide a common, standard notation for recording both analysis models and design artifacts. Facets of the Unified Process approach to designing and building a software system are also covered.

Pre-requisites: Object oriented Design concepts, Design & Analysis of Algorithms and Software engineering.

Learning outcomes: At the end of the course, participants will be able to

1. be familiar with standard Unified Modelling Language (UML) notation
2. Model requirements with Use Cases
3. Describe the dynamic behaviour and structure of the design.
4. Describe Object Oriented Analysis and Design concepts and apply them to solve problems
5. Prepare Object Oriented Analysis and Design documents for a given problem using UML

UNIT – I

Introduction to Object Oriented Paradigm [4 hrs.]
Objects, Functional method and Object-Oriented method, the key concepts of OOP, Relationship among objects.

Object Orientation [6 hrs.]
Object orientation, Fundamentals of Object Oriented approach, various concepts, Object Oriented Analysis &Object Oriented Design.

Object Oriented methodology [8 hrs.]
Major issues in OOP, Typical Object-Oriented development process, Rumbaugh et. al.'s object model, Booch Model, Jacobson et. al. methodology, Coad and Yourdon Object Oriented analysis model, Fusion method, Pattern& frameworks.

UML Basics [2 hrs.]
UML Background, Its evolution and various variants, UML building blocks and modelling views.

UNIT – II

Requirements eliciting, its analysis and specification [4 hrs.]
Requirements engineering using various UML tools such as Use Case Modelling, Conceptual modelling, Behavioural modelling.

Software Architecture [8 hrs.]

System Architecture using various UML tools such as collaboration diagrams, Component diagrams, Packages, Frameworks and patterns

.

Design [8 hrs.]

Design of class diagrams, Activity diagrams, Sequence diagrams, State chart diagrams, Design patterns.

Text Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Addison- Wesley.
2. Ali Bahrami, Object Oriented Systems Development, Tata McGraw Hill.

Reference Books:

1. Michael R Blaha, James R Rumbaugh, Object-Oriented Modeling and Design with UML, Pearson.
2. Grady Booch, Object Oriented Analysis and Design, Wesley.
3. Priestley, Practical object oriented design with UML, Tata Mcgraw Hill
4. Srimathi, Object oriented analysis and design using UML, Scitech.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 25** **(3L + 1 T hrs./week)**

ELECTIVE - II

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives and Pre-requisites are given under Electives details.

TO BE CHOSEN/OFFERED FROM LIST OF ELECTIVES FOR FIFTH SEMESTER.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 25** **(3L + 1 T hrs./week)**

ELECTIVE – III

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives and Pre-requisites are given under Electives details.

TO BE CHOSEN/OFFERED FROM LIST OF ELECTIVES FOR FIFTH SEMESTER.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2561 **(3 hrs./week)**

UML LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper CA 2503 - Object Oriented Design and UML and associated prerequisites.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2562 **(3 hrs./week)**

ANALYSIS AND DESIGN OF ALGORITHMS LAB

Objective: At least 10 experiments covering the entire syllabus of the corresponding theory paper to be carried out using the theory studied /programming skill of the subject concerned to get insight into the practical applications of the theoretical studies. The outcome of the lab classes must lead to a skilled and self-sustained program developer.

Pre-requisites: Corresponding theory paper MCA 2502 - Analysis and Design of Algorithms and associated prerequisites.

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2571

MINI PROJECT

Sl No	Subject	Objectives	Total Credits
1.	Mini Project	The students are required to undertake innovative and research oriented project under the direct supervision of a faculty member of the department. The mini project should not only to reflect their knowledge gained in the previous Semesters but also to acquire additional knowledge and skill of their own effort. The mini projects are assigned at the end of IV semester and the final evaluation and grades are awarded at the end of V semester.	
2.	Mini Project Reviews	The progress is being evaluated in phases through interim seminars/ presentations to make the department aware of his/her project.	1
3.	Awarded by Project Guide	The faculty-guide assesses the work of the group(s) working under his/ her guidance on day to day basis and evaluate.	1
		Total Credits	2

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2581

INDUSTRIAL TRAINING/COURSE WORK

Sl No	Subject	Objectives	Total Credits
1.	Industrial training/ Coursework	The students are required either to undergo 45days training in Industries or to attend summer training course on courses beyond the scope of normal curriculum organized by the Department by calling experts from outside.	1
		Total Credits	1

DEPARTMENT OF COMPUTER APPLICATIONS

CA 2675 **Duration of Project: 16 weeks**

MAJOR PROJECT

Sl No	Subject	Objectives	Total Credits
1.	Major Project	The students are required to undertake innovative and research oriented project, not only to reflect their knowledge gained in the previous seven semesters but also to acquire additional knowledge and skill of their own effort. During their major project, the students are required to submit progress of their work in phase's manner to make the department aware of his/her project. At the end of 16 weeks, students have to report to the internal guides/faculty members for final refinement and documentation. It is mandatory to follow the software engineering methodologies in carrying out the project work. The project is evaluated through internal presentation before the panel of faculty members followed by the evaluation by external examiner appointed by the university.	16
		Total Credits	16

DEPARTMENT OF COMPUTER APPLICATIONS

CA2431 **(3L + 1 T hrs./week)**

BIOINFORMATICS

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: The course is designed to introduce the basic concepts, methods, and tools used in Bioinformatics. The objective is to help the students to reach rapidly the frontier of bioinformatics and working knowledge of a variety of publicly available data and the bioinformatics tools in handling flood of biological data and to solve the problems on their own research. The course also highlights the programming paradigm of Bioinformatics.

Pre-requisites: Concept of Algorithms, Probability and Statistics and Knowledge of Biology.

Learning outcomes: By the end of this module, students should be able to

1. Set a solid background in bioinformatics
2. Choose and modify suitable computational models to solve the problem for a given biological sequence analysis problem
3. Develop computer programs for bioinformatics data analysis.
4. Interpret the advantages and/or disadvantages of these approaches identify and pursue research topics in bioinformatics

UNIT- I

Bioinformatics an Overview [7 Hrs.]
Definition and history, Information networks, Internet in bioinformatics, EMBnet , Commercial databases and software, Intranet and internet packages, Bioinformatics glossary.

Bioinformatics Programming using Perl [10 Hrs.]
Basics of programming PERL, Basics, String handling, subroutines.HTML basics, Tags, Text handling, Image handling, Links, Frames and tables. XML, Basics, Data binding and record sets.

UNIT- II

Protein Information Resources [10 Hrs.]
Biological databases, Primary sequence databases, Composite protein sequence databases, -Secondary databases, Prosite, Prints, Blocks profiles and identity.

Genome Information Resources [8 Hrs.]
DNA sequence databases, EMBL DDBJ, Genbank GSDB (Genome, Sequence database), UniGene.

Evolution of Bioinformatics [5 Hrs.]
Scope, Potentials of bioinformatics, Human genome project, Bioinformatics in India, Future of bioinformatics.

Text Books:

1. Arthur M. Lesk, Introduction to Bioinformatics, Oxford University Press.
2. T.K. Attwood and D.J. Parry-Smith, Introduction to Bioinformatics, Pearson Education.

Reference Books:

1. D. Higgins and W. Taylor, Bioinformatics- Sequence, structure and databanks, Oxford University Press.
2. Baxevanis and B.F. Ouellette. Bioinformatics: A practical Guide to the Analysis of Genes and Proteins, Wiley.
3. S. R. Swindell, R.R.Miller and G.S.A.Myers (Eds.), Internet for the Molecular Biologist, Horizon Scientific Press.
4. Andrea Cabibbo, Richard Grant and Manuela Helmer-Citterich (Eds.), The Internet for Cell and Molecular Biologists, Horizon scientific Press.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2432 **(3L + 1 T hrs./week)**

DIGITAL IMAGE PROCESSING

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: The objective of this course is to provide the students a general understanding of the fundamentals of digital image processing. It also introduces analytical tools which are currently used in digital image processing. By the end of the course student will be able to develop any software/programs that uses image restoration, enhancement and compression etc.

Pre-requisites: Computer Graphics, Engineering Mathematics and Probability & Statistics

Learning outcomes: By the end of this course, the students should have

1. A deep understanding of the methods presented in the course.
2. Ability to apply the methods presented or combinations of them or modifications of them in a real life image processing problem.
3. The experience to decide which method is appropriate to tackle a real life problem
4. Ability to do further independent work on the subject including topics which are not covered in the class

UNIT- I

Introduction [5 Hrs.]
Digital image representation, Fundamental steps in Image Processing, Elements of DIP systems.

Digital Image Fundamentals [5 Hrs.]
Elements of Visual Perception, Sampling and Quantization, Relationships between pixels, Linear and Nonlinear operations.

Image Enhancement in Spatial domain [5 Hrs.]
Enhancement by Point Processing, Histogram Processing, Spatial Filtering.

Image Enhancement in Frequency Domain [5 Hrs.]
Introduction to the Fourier Transform, The discrete Fourier Transform, Properties of the two-dimensional Fourier Transform, Smoothing Frequency-domain filters, Sharpening Frequency domain filters.

UNIT- II

Image Compression [5 Hrs.]
Fundamentals, Image Compression Models, Error Free Compression, Lossy Compression.

Image Segmentation [5 Hrs.]
Threshold Techniques: Global, Adaptive and Optimum thresholding, Edge detection, Region Growing.

Representation and Description [5 Hrs.]

Representation Schemes, Boundary Descriptors, Regional Descriptors.

Object Recognition [5 Hrs.]

Patterns and Pattern Classes, Recognition based on Decision-theoretic methods, structural methods.

Text Books:

1. Rafael C Gonzalez, Richard E Woods, Digital Image Processing, Pearson Education.
2. Rajjan Shinghal, Pattern Recognition, Oxford.

Reference Books:

1. Chanda and Majumder, Digital Image Processing and Analysis, Prentice Hall.
2. Rafael C Gonzalez, Richard E Woods, Digital Image Processing with Matlab, Pearson Education.
3. S. Sridhar, Digital Image Processing, Oxford University Press.
4. Jayaraman, Digital Image Processing, Tata McGraw Hill.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2433 **(3L + 1 T hrs./week)**

DATA WAREHOUSING AND DATA MINING

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: This course is indent for understanding the techniques behind the recent development in data warehousing and data mining. The data warehousing part of the module aims to give students a good overview of the ideas and the techniques, which are behind recent developments in the data warehousing and On Line Analytical Processing (OLAP) fields, in terms of data models, query languages, conceptual design methodologies, and storage techniques. The data mining part discusses various tools and techniques used to find out the interesting patterns from data.

Pre-requisites: Data structures, Concepts of Algorithms, Fundamentals of probability & statistics and programming skills

Learning outcomes: On completion of this module, the student should be able to

1. Understand the techniques behind the recent development in data warehousing and data mining.
2. Understand query languages and conceptual design methodologies.
3. Practice on different tools of data warehousing and data mining.
4. Design small projects with data mining and data warehousing.

UNIT-I

Data Warehousing [5 Hrs.]
Introduction, Definition, Multidimensional data transformation, OLAP operations, Warehouse schema, Warehouse Server, Meta data, OLAP engine.

Data Mining [5 Hrs.]
Introduction, Definition, KDD vs. DM, DBMS vs. DM, DM techniques, Issues and challenges in DM, DM applications.

Association Rules [5 Hrs.]
A prior algorithm, Partition, Incremental, Dynamic item set counting, FP-tree growth, Incremental and border algorithms.

Classification: Parametric and Non-Parametric Technology [6 Hrs.]
Bayesian classification, two class and generalized class classification, Classification error, Decision boundary, Discriminant functions, Non-parametric methods for classification.

UNIT-II

Clustering [7 Hrs.]

Clustering analysis, Types of data in cluster analysis, Partitioning algorithms, Hierarchical, Density based, Grid based, Model based algorithms, High dimensional & Categorical data clustering.

Decision Trees [7 Hrs.]

Decision tree induction, Tree pruning, Extracting classification rules from decision trees, Decision tree construction algorithms, Decision tree construction with pre-sorting.

Unstructured Data Mining [5 Hrs.]

Text mining, Web mining, spatial data mining, Multimedia data mining.

Text Books:

1. Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann.
2. A K Pujari, Data Mining Concepts, University Press.

Reference Books:

1. Ian H. Witten, Eibe Frank, Data Mining Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann Publishers.
2. Alex Berson, Stephen J. Smith, Data Warehousing, Data Mining and OLAP, Tata McGrawHill.
3. Richard O. Duda, Peter E. Hart, Pattern Recognition and Scene Analysis, Wiley.
4. VikramPudi, P. Radha Krishna, Data Mining, Oxford University Press.

CA2434 **(3L + 1 T hrs./week)**

MANAGEMENT INFORMATION SYSTEM

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: The main objective of this course is to describe the use and function of Management Information Systems, describe and evaluate Information Systems Development processes and techniques. It also identifies and evaluates hardware and software requirements for Information Systems, Data Management technologies and the security risks associated with Management Information Systems.

Pre-requisites: Programming skills.

Learning Outcomes: On completion of the course the students will be able to

1. Apply fundamental business analysis methods to understand the organizational and economic characteristics of organizations.
2. Work ethically and effectively with others to implement new processes and systems in an organization.
3. Apply the fundamental concepts of information systems and technology to organizational and societal priorities.
4. Model real-world situations for information systems support.
5. Apply best practices to manage information systems projects and programs.
6. Reflect thoughtfully on personal goals within a dynamic and evolving profession.

UNIT – I

Organization and Information Systems [6 hrs.]

Changing Environment and its impact on Business - The IT/IS and its influence - The Organization: Structure, Managers and activities - Data, information and its attributes - The level of people and their information needs - Types of Decisions and information - Information System, categorization of information on the basis of nature and characteristics.

Kinds of Information Systems [6 hrs.]

Transaction Processing System (TPS) - Office Automation System (OAS) - Management Information System (MIS) - Decision Support System (DSS) and Group Decision Support System (GDSS) - Expert System (ES) - Executive Support System (EIS or ESS).

System Analysis and Development and Models [8 hrs.]

Need for System Analysis - Stages in System Analysis - Structured SAD and tools like DFD, Context Diagram Decision Table and Structured Diagram. System Development Models: Water Flow, Prototype, Spiral– Roles and responsibilities of System Analyst, Database Administrator and Database Designer.

UNIT – II

Enterprise System [7 hrs.]
Enterprise Resources Planning (ERP): Features, selection criteria, merits, issues and challenges in Implementation- Supply Chain Management (SCM): Features, Modules in SCM - Customer Relationship Management (CRM)and its Phases.

Knowledge Management and E-Governance [6 hrs.]
Choice of IT Nature of IT decision - Strategic decision - configuration design and evaluation Information technology implementation plan.

Security and Ethical Challenges [7 hrs.]
Ethical responsibilities of Business Professionals – Business, technology. Computer crime – Hacking, cyber theft, unauthorized use at work. Piracy – software and intellectual property. Privacy – Issues and the Internet Privacy. Challenges – working condition, individuals. Health and Social Issues

Text Books:

1. Kenneth J Laudon, Jane P. Laudon, Management Information Systems, Pearson.
2. W. S. Jawadekar, Management Information Systems, Tata McGraw Hill.

Reference Books:

1. James A O' Brien, Introduction to Information System, Tata McGraw Hill.
2. S. Sadagopan, Management Information Systems, PHI.
3. Effy Oz, Management Information Systems, Thomson Course Technology.
4. Lynda M Apple Gate, Robert D Austin Corporate Information Strategy and Management, Tata McGraw Hill.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2435 **(3L + 1 T hrs./week)**

UNIX/ LINUX INTERNALS

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course focuses on the principles and techniques of Unix Operating Systems concepts and terminologies, including file system programming, shell programming including advanced Unix commands and utilities, process management, signal management, inter process communication (IPC) issues and techniques in Unix programming.

Pre-requisites: Data Structures, Operating Systems and Programming language concepts.

Learning outcomes: The students who succeeded in this course will be able to

1. Express the historical development of UNIX/Linux operating systems.
2. Have an introductory knowledge about Unix/Linux internals and utilities.
3. Use the bash shell and the basic commands in this shell.
4. Write scripts to be run with bash in Unix/Linux operating system.
5. Have the basic abilities on the administration issue of Unix/Linux operating systems.
6. Explain the concept of open source software development.

UNIT – I

General overview of the system

History, System structure, User perspective, Operating system services, Assumptions about hardware, Introduction to the kernel, Architecture of the UNIX Operating system, Introduction to System concepts, Kernel data structures, System administration, Summary and preview.

Buffer cache

Buffer headers, Structure of the Buffer Pool, Buffer Retrieval, Reading and Writing Disk Blocks Advantages and disadvantages of the buffer cache. Internal representation of files, inodes, Structure of a regular file, Directories, Conversion of a path name to an inode, Super block, inode Assignment, Allocation of Disk Blocks and other file types.

System calls for file system

Open, Read, , Write, File and record locking, Adjusting the position of file I/O, LSEEK , Close, File creation, Special files creation, Change Directory and Change Root-Change Owner and Change Mode, Stat-Fstat Pipes, Dup, Mounting and unmounting file systems, Link, Unlink, File System Abstraction, Maintenance.

UNIT – II

The System Representation of Processes

Process, States and Transitions, Layout of System Memory, Context of a process, Saving the context of a process, Manipulation of a Process Address Space, Sleep, Process control, Process creation, Signals, Process termination, Awaiting process termination, Invoking other programs, The Shell, System Boot and the INIT process.

Process scheduling and memory management policies

Process scheduling, Memory management policies, Swapping, A Hybrid system with swapping and demand paging, The I/O subsystem, Driver interfaces, Disk drivers, Terminal drivers.

I/O Subsystems and Interprocess Communications

Driver interfaces, Disk Drivers, Terminal Drivers, Streams, Process tracing, System V IPC and Network Communications.

Text Books:

1. Bach, M.J., The Design of the Unix Operating System, PHI.
2. Karee Christian, The Unix Operating System, John Wiley & Sons.

Reference Books:

1. Vahalia, Unix Internals: The New Frontiers, Pearson Education.
2. UreshVahalia, UNIX Internals: The New Frontiers, PHI.
3. M. Beck et al, Linux Kernel Programming, Pearson Education.
4. Sumitabha Das, UNIX Concepts and Applications, Tata McGraw Hill.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2436 **(3L + 1 T hrs./week)**

ENTERPRISE RESOURCE PLANNING

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: This course intends to familiarize the students about the connection between all the business disciplines in the real world and how ERP systems support the planning and management of business processes. It also gives students hands-on exposure to how enterprise-wide information systems support the planning and management of business processes.

Pre-requisites: No departmental pre-requisite. Knowledge of System Analysis and Design and Management
Information System is desirable.

Learning outcomes: After completing this course, students will have the ability to

1. Understand the structure and the customization of an ERP system.
2. Plan the implementation and customization of an ERP system using the appropriate modeling methods.
3. Understand and implement data handling techniques in an ERP system.
4. Understand software design issues, integration and supply chain and customer relationship applications.
5. Realize the importance of project management in an ERP implementation project.
6. Understand what to expect, and not to expect, from a consultant implementing an ERP system.
7. Understand the importance of IT governance in long-term relationships with a software vendor, such as SAP.

UNIT- I

ERP Introduction [4 hrs.]
Business Functions and Business Processes, A Brief History of ERP, The advantages of ERP, Common ERP Myths, How does ERP create Value.

Risks and benefits of ERP [6 hrs.]
Justifying ERP investments, Quantifiable benefits from the ERP System, The Intangible benefits of ERP, Risk factors of ERP implementation, People Issues, Process Risks, Technological Risks, Implementation Issues, Operation and Maintenance Issues, Managing Risk on ERP projects, Benefits of ERP.

ERP and Related Technologies [5 hrs.]
Business Process Reengineering, Data Warehousing, Data Mining, OLAP, PLM, SCM, CRM, GIS, Intranets and Extranets, Middleware, Computer Crimes, Security and ERP, Computer Security, Crime and Security.

ERP Functional Module [4 hrs.]
Functional Modules of ERP Software, Integration of ERP, Supply Chain and customer relationship applications.

UNIT-II

ERP Implementation Life Cycle [5 hrs.]
Objectives and different phases of ERP Implementation, Why many ERP implementations fails.

ERP transition strategies [5 hrs.]
Transition strategies, Big Bang Strategy, Phased implementation, Parallel implementation, Process line transition strategy, Hybrid Transition strategy, Choosing a strategy.

ERP implementation process [6 hrs.]
Implementation methodologies, Organizations of the ERP project team, Implementation strategy, ERP implementation plan, Risk assessment, Budget, Cost, Performance measurement, Problem resolution, System Issues, ERP Implementation-The Hidden Costs, Success and Failure factors of an ERP Implementation.

ERP Operation and Maintenance [5 hrs.]
On-going implementation efforts, Upgrading versus new software, Operation and maintenance of the ERP System, ERP maintenance phase.

Text Books
1. Alexis Leon, Enterprise Resource Planning, Tata McGraw-Hill.
2. Vinod Kumar Garg and N.K.Venkita Krishnan, Enterprise Resource Planning – Concepts and Practice, PHI.

Reference Books
1. Rahul V. Altekar, Enterprise wide Resource Planning, Tata McGraw Hill,
2. Joseph A Brady, Ellen F Monk, Bret Wagner, Concepts in Enterprise Resource Planning, Thompson Course Technology.
3. Alexis Leon, ERP Demystified, Tata McGraw Hill.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2438 **(3L + 1 T hrs./week)**

OPTIMIZATION TECHNIQUE

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: The objective of this course is to understand the need and origin of the optimization methods and to get a broad picture of the various applications of optimization methods used in engineering. This course is indent for designing and controlling complex systems, solving hard problems of efficiently allocating scarce resources using in complete information, and developing sustainable strategies to master situations of conflict and co-operation using scientific methods and information technology.

Pre-requisites: Quantitative Analysis using C/C++, Design and Analysis of Algorithms and Probability & Statistics.

Learning outcomes: On completion of the course it is expected to endow the students with skills to

1. Demonstrate knowledge and understanding of the basic ideas underlying optimization techniques.
2. Obtain solutions for continuous linear optimization problems using the simplex method and duality theory.
3. Analyse the robustness of continuous linear optimization problems solutions using sensitivity analysis.
4. Formulate and large-scale linear and integer programming problems using computer.
5. Build solutions for mixed integer and binary optimization problems using tree-search algorithms.
6. Use decomposition techniques to solve hard optimization problems
7. Appreciate the power of using the mathematical approach to optimization problems relevant to engineering.

UNIT – I

Introduction to Operations Research [8 Hrs.]
Introduction to OR modeling approach and various real life situations, Linear programming problems and applications, Solving Linear Programming problem using simultaneous equations and Graphical Method, Simplex Method and extensions, Sensitivity analysis - Duality theory. Transportation model, Transhipment problems and Assignment problems.

Dynamic Programming [6 Hrs.]
Bellman’s principle of optimality, Examples on the application on routing problem, Inventory problem, Simplex problem, Marketing problem.

Network Analysis [6 Hrs.]
PERT and CPM, Probability of achieving completion data, Cost analysis, Graph reduction theory, Updating, Resource allocation, Resource smoothing.

UNIT – II

Inventory Method [7 Hrs.]

Inventory problem, Variables in an inventory problem, Inventory models with penalty, Storage and Quantity discount, Safety stock, Inventory models with probability Demand, Multi-item deterministic model. Simulation, Types of simulation models, Applications of simulation for Inventory problems.

Queuing Theory [7 Hrs.]

Poison arrivals and Exponential service times, Waiting time and Idle time cost, Single channel and Multi-channel problem. Applications of simulation for Queuing problems. Monte Carlo technique applied to queuing problems, Poisson arrivals and service time.

Theory of Games [6 Hrs.]

Introduction – Minimax (maximin) – Criterion and optimal strategy – Solution of games with saddle points –Rectangular games without saddle points – 2X2 games, Examples on the application of theory of games. –Dominance principle – mX2 & 2Xn games - Graphical method and Linear programming method for different problems, Decision trees.

Text Books:

1. Hamdy A. Taha, Operations Research, Macmillan.
2. Kumar Gupta, Prem and Hira, D.S., Operations Research, S Chand.
3. Swarup, Kanti, Gupta, P.K. and Manmohan, Operations Research, Sultan Chand & Sons.

Reference Books:

1. Richard Bronson, Operations Research – Schaum outline series, Tata McGraw Hill.
2. V.K. Kapoor-- Operations Research, S Chand & Sons.
3. Hiller F. and Leibermann G. J., Operation Research, Tata McGraw Hill.
4. Srinath L.S., PERT & CPM Principles and Applications, Affiliate East West Press(P).

DEPARTMENT OF COMPUTER APPLICATIONS

CA2439 **(3L + 1 T hrs./week)**

MOBILE APPLICATION DEVELOPMENT

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objective: The Java Programming Language course provides students with a solid foundation for programming with JAVA. It also highlights the creation of graphical user interfaces (GUIs), exceptions, file input/output (I/O), and threads; and network programming.

Pre-requisites: Basic Programming Concept, Object Oriented Programming.

Learning outcomes: On successful completion of course, the learner will be able to

1. Describe the requirements for mobile applications
2. Explain the challenges in mobile application design and development
3. Develop and design mobile applications for specific requirements
4. Implement the design using Android SDK
5. Implement the design using Objective C and iOS
6. Deploy mobile applications in Android and iPhone marketplace for distribution

UNIT- I

Introduction
Introduction to mobile applications, Embedded systems, Market and business drivers for mobile applications, Publishing and delivery of mobile applications, Requirements gathering and validation for mobile applications

Basic Design
Introduction to basics of embedded systems design, Embedded OS, Design constraints for mobile applications, Architecting mobile applications, User interfaces for mobile applications, touch events and gestures, Achieving quality constraints, performance, usability, security, availability and modifiability.

Advanced Design
Designing applications with multimedia and web access capabilities, Integration with GPS and social media networking applications, Accessing applications hosted in a cloud computing environment, Design patterns for mobile applications.

UNIT -II

Android Development Environment
Introduction to Android Environment, Establishing the development environment, Android architecture, Activities and views, Interacting with UI, Persisting data using SQLite, Packaging and deployment, Interaction with server side applications, Using Google Maps, GPS and Wifi – Integration with social media applications.

IOS Development environment

Introduction to Objective C, iOS features, UI implementation, Touch frameworks, Data persistence using Core Data and SQLite, Location aware applications using Core Location and Map Kit, Integrating calendar and address book with social media application, Using Wifi - iPhone marketplace.

Text Books:

1. Jeff McWherter and Scott Gowell, Professional Mobile Application Development, Wrox Publishers.
2. Charlie Collins, Michael Galpin and Matthias Kappler, Android in Practice, DreamTech,

Reference Books:

1. James Dovey and Ash Furrow, Beginning Objective C, Apress.
2. David Mark, Jack Nutting, Jeff LaMarche and Frederic Olsson, Beginning iOS 6 Development: Exploring the iOS SDK, Apress.
3. DT Editorial Services, Mobile Application Development Black Book, Dreamtech Press.
4. Carmen Delessio, Lauren Darcey, Shane Conder, Sams Teach Yourself Android Application Development in 24 Hours, Sams Publishing.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2531 **(3L + 1 T hrs./week)**

ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: The objective of the course is to learn various areas of Artificial Intelligence, such as symbolic representation, knowledge representation, cognitive models, goal-based systems, heuristic search, games as environments, deductive systems and expert systems, vision; audition, language understanding etc. Rather than present AI as a loose collection of ideas and techniques, this course will strive to emphasize important unifying themes that occur throughout many areas of AI research.

Pre-requisites: Data Structures, Analysis and Design of Algorithms and Statistics

Learning outcomes: On completion of this course, the students will be able to

1. Develop an understanding of artificial intelligence, problem solving techniques and game playing strategies
2. Understand knowledge representation tools
3. Draw inferences using the concepts of knowledge and reasoning
4. Understand natural language processing
5. Understand and apply the concepts of expert system

UNIT- I

Artificial Intelligence [4 hrs.]
Introduction, Intelligent agents: Agents and environments, the concept of rationality, Nature of environments, Structure of environments

Problem Solving [7 hrs.]
Solving Problems by Searching: Problem solving agents, Uninformed Search strategies, Avoiding repeated states, Informed Searches and exploration, Informed search strategies, Heuristic functions, Local search algorithms and optimization problems.

Adversarial searches [5 hrs.]
Games, Optimal decision in games, Alpha beta pruning, Imperfect real-time decisions

UNIT- II

Knowledge and Reasoning [7 hrs.]
Logical Agents: Knowledge based agents, Propositional logic, Resolution, Effective propositional inference, Agents based on propositional logic.

First-Order Logic [3 hrs.]
Syntax and semantics of First order logic, Using FOPL.

Inference in First-Order Logic [4 hrs.]

Unification, Forward chaining, Backward chaining, Resolution.

Learning [5 hrs.]

Learning from Observations: Form of learning, Inductive learning, Learning decision tree, Ensemble learning, Computational learning theory.

Expert Systems [5 hrs.]

Introduction (characteristic features of expert systems, Background History, Applications, importance of expert systems); Rule based system architectures (the knowledge base, the inference process, explaining how or why, building a knowledge base, the I/O interface).

Text Books:

1. S. Russel, & P Norvig, Artificial Intelligence A Modern Approach, Pearson Education.
2. Nills J Nillson, Artificial Intelligence A new Synthesis, Morgan Kauffman

Reference Books:

1. Luger, Artificial Intelligence, Pearson education.
2. Charnials Engine, Introduction to artificial intelligence, Wesley.
3. Petterson, Introduction to artificial intelligence and expert systems, PHI.
4. Padhy, N.P., Artificial intelligence and intelligence system, Oxford University Press.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2532 **(3L + 1 T hrs./week)**

GRAPH THEORY

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: This course provides a formal introduction to the theory of graphs including paths, circuits, trees, matrix representation of graphs and their applications. The goal of this course is to make students aware of how graphs are used to model different situations and processes with special emphases on computer science and engineering applications.

Pre-requisites: Data Structures, Concepts of algorithms, Engineering mathematics and Programming concepts.

Learning outcomes: Upon successful completion of this course, students should be able to

1. Understand the basic concepts of graphs, directed graphs, weighted graphs and bipartite graphs, Eulerian, Hamitonian and plane graphs.
2. Find the components of a graph and the strongly connected components of a digraph.
3. Apply coloring algorithms to select the registers for memory location or to find its chromatic polynomial.
4. Construct breadth/depth first search and minimum weight spanning tree of a connected graph.
5. Perform matrix representation of a graph.
6. Apply the max flow/min cut algorithm for finding a maximum (s; t) flow in a network.
7. Find a maximum matching and a maximum weight matching in a bipartite graph.
8. Find an Euler trail in a graph or digraph and for solving the Chinese Postman Problem.

UNIT- I

Introduction [2 Hrs.]
Definitions, applications of graph in Diagram tracing, Konigsberg bridge problem, Chinese postman problem, DNA fragment assembly, floor design, Knight's tour, Integer programming, Solution to the travelling salesman problem. Isomorphism, walks, paths, circuits, connected, disconnected graphs, operations in graphs-Euler & Hamilton graphs.

Tree [4 Hrs.]
Properties, distance & canters, binary trees, fundamental circuits, minimal spanning tree. Tree traversal. Applications of Tree in manipulating hierarchical data, Parse tree, File system, manipulate sorted lists of data, workflow for compositing digital images for visual effects and Routing algorithms.

Cut sets [4 Hrs.]
Properties, Fundamental circuits and cut sets, connectivity, separatability network flows, 1-2 isomorphism. Applications of cut sets to solve graph models for reliability analysis, identify potential single point failures in a modelled system, state equation for network and superposition network theorem.

Planar and dual graphs [4 Hrs.]
Combinational representation, planar graphs, Kuratowshi's graphs detection of planarity, dual graphs. Applications of planar graphs in image segmentation, shape matching, extended modelling capabilities, route planning.

Matrix representation of graph [5Hrs]
Incidence matrix, circuit matrix, cut set matrix, fundamental matrices, relationships among matrices, path matrix, and adjacency matrix. Applications of Matrix in computer programming language for the computation of path or circuit.

UNIT-II

Coloring, covering & partitioning [6 Hrs.]
Chromatic number, chromatic partitioning, matching, covering, four color problem. Coloring and covering concepts used in pattern matching, register allocation, frequency assignment in GSM, time tabling, scheduling and computer network security.

Directed graphs [6 Hrs.]
Different types, directed path, and connectedness, Euler digraphs, Trees, matrix representation, tournament. Directed graphs used in compiler construction, finite state machine, combinational circuit design and traffic flow problem.

Graph theoretic algorithms [4 Hrs.]
Computer representation of graphs, Input / output. Devising algorithms for connectedness, a spanning tree, fundamental circuits, cut vertices, directed circuits, shortest paths. Applications of Graph theoretic algorithm in worm propagation, workflow for compositing digital images for visual effects, routing algorithms.

Applications [3Hrs]
Graph in sequential switching networks, graph in coding theory, graph in signal flow graph, graph in markov process, and graphics in computer programming. Applications in code generation, sequential switching networks, graphics.

Text Books:

1. N.Deo, Graph Theory with applications to Engineering and Computer Science, PHI.
2. Jonathan L. Gross and Jay Yellen, Graph Theory and Its Applications, CRC Press.

Reference Books:

1. M. N. S. Swamy, K. Thulasiraman, Graphs, Networks & Algorithms, Wiley Interscience.
2. F. Harary, Graph Theory, Narosa.
3. Jonathan L. Gross and Jay Yellen, Handbook of Graph Theory, CRC Press.
4. Jonathan L. Gross and Thomas W. Tucker, Topological Graph Theory, Dover.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2533 **(3L + 1 T hrs./week)**

PRINCIPLES OF PROGRAMMING LANGUAGE

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: The objective of this course is to identify the conceptual building blocks from which languages are assembled and specify the semantics, including common type systems, of programming languages.

Pre-requisites: Programming languages & Concepts.

Learning outcomes: After the completion of the course students will be capable of

1. Visualizing the problems for software requirements and decomposing into various modules for clarity and efficiency.
2. Preparing initial documents like SRS etc. effectively.
3. Incorporating OOD approach for developing software.
4. Develop programs using special procedural language: SCHEME.
5. Creating effective design like DFD, ERD and Data dictionaries for any software.

UNIT- I

Programming Concepts [12 Hrs.]
Declarative, Imperative knowledge, Procedure, Process, Language, Basic set of data elements, basic set of operations, Introduction to Scheme, Compound expressions, Primitive procedures, Variables and Functions. Designing Programs with specifications, Functions, Predicates, Conditional expressions, scope and extent, Recursive and Iterative Processes and Procedures, Procedural and Data Abstraction, Types, Compound data, Introduction to data structures.

Structured Modeling and Implementation [8 Hrs.]
Data flow diagram, Data dictionary, Mini specification, ERD, Structure chart, Normalization, Testing Methodologies, Case study on modelling and implementation – Emphasis on applying SSAD principles to solve a case study.

UNIT- II

Object Oriented Programming [9 Hrs.]
Building abstractions, Controlling interactions, Objects and message passing, OO Concepts (Abstraction/Encapsulation/Inheritance/ Delegation).

Object Oriented Modelling and Implementation [1Hrs.]
Overview of UML, Model views and UML diagrams, applying UML, MVC architecture, Design patterns – GoF, Case study modeling and implementation - Emphasis on applying OOAD principles to solve a case study end to end.

Text Books:

1. Mathias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi, How to Design Programs, Prentice Hall.
2. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa.
3. Craig Larman, Applying UML and Patterns, Pearson Education.

Reference Books:

1. Steve McConnell, Code Complete, WP Publishers and Distributors.
2. Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich, Prabin K. Panigrahi, Modern Systems Analysis and Design, Pearson Education.
3. Meilir Page-Jones, The Practical Guide to Structured Systems Design, Prentice Hall.
4. Glenford J. Myers, The Art of Software Testing, Wiley.
5. Roger S. Pressman, Software Engineering, McGraw Hill.
6. Martin Fowler, UML Distilled, Wesley.

CA2534 **(3L + 1 T hrs./week)**

SYSTEM SIMULATION AND MODELING

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: This course envisages the fundamentals of discrete event simulation (DES), which includes discrete event simulation methodology, development of simulation models, verification and validation, and the design of simulation experiments.

Pre-requisites: Probability & Statistics and System Analysis concepts.

Learning outcomes: On completing this course students should be able to

1. Define basic concepts in modelling and simulation and demonstrate an understanding of system modelling through the competent use of computer simulation methods and mathematical modelling techniques
2. Determine the type of systems whose behaviour can be investigated using discrete event simulation and modelling as well as system dynamics-simulation modelling technique
3. Classify various simulation models and give practical examples for each category
4. Construct a model for a given set of data and motivate its validity
5. Generate and test random number variates and apply them to develop simulation models
6. Analyze output data produced by a model and test validity of the model

UNIT – I

Introduction to Simulation [5 Hrs.]
Components of a system, Model of a system, Types of models, Discrete and continuous systems, Components and organization of a discrete event simulation model, Continuous simulation, Combined discrete-continuous simulation, Monte Carlo simulation, Steps in simulation study, Advantages, Disadvantages, and pitfalls of simulation.

Simulation Examples [3 Hrs.]
Simulation of Queuing systems, Simulation of inventory systems, Simulation of reliability Systems.

General Principles and Modelling Complex Systems [5 Hrs.]
The event scheduling approach, the process interaction approach, the activity scanning approach, List processing in simulation: Approaches to storing lists in a computer, Time-shared computer model, Multi-teller Bank with Jockeying, Job-Shop Model, Efficient event-list manipulation.

Statistical Models in Simulation [2 Hrs.]
Review of terminology and concepts, Useful statistical models, discrete distributions, Continuous distributions.

Queuing Theory [5 Hrs.]

Characteristics of queuing systems, Queuing Notations, Transient and Steady-State behaviour of Queues, Long –Run measures of performance of Queuing Systems, Steady state behaviour of Infinite population Markovian models [M/G/1 only], Networks of Queues.

UNIT – II

Random Number Generation [4 Hrs.]

Properties of random numbers, Generation of pseudo random numbers, Random-Number Generators: Linear Congruential Generators, Combined Linear Congruential Generators, Feedback Shift Register Generators, Tests for Random Number Generators.

Building Valid, Credible, and Appropriately Detailed Simulation Models [4 Hrs.]

Guidelines for determining the level of Model Details, Verification of Simulation Computer Programs, Techniques for increasing Model Validity and Credibility, Statistical Procedures for Comparing Real-World Observations and Simulation Output Data.

Selecting Input Probability Distributions [5 Hrs.]

Identifying the distribution with data, Estimation of Parameters, Determining how representative the fitted distributions are, Selecting input Models in the absence of data, Models of arrival Processes(Poisson process, Non stationary Poisson process, batch arrivals).

Output Data Analysis for a Single System [4 Hrs.]

Transient and Steady State behaviour of a stochastic process, Types of simulations with respect to output analysis.

Experimental Design [3 Hrs.]

Common mistakes in experimentation, Types of Experimental Designs, 2k factorial Designs, 2kr factorial Designs.

Text Books:

1. Averill M Law, Simulation Modeling and Analysis, Tata McGraw Hill.
2. Banks, Carson, Nelson, Nicol, Discrete-Event System Simulation, Pearson Education.

Reference Books:

1. Raj Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, John Wiley & Sons.
2. Gordon, System Simulation, Prentice Hall.
3. Singh V.P, System Modelling and Simulation, New Age International.
4. Frank L. Severence, System Modeling and Simulation, Wiley.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2535 **(3L + 1 T hrs./week)**

AD-HOC WIRELESS NETWORKS

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: This course covers all aspects of Ad Hoc and Sensor networking from design through performance issues to application requirements. The course starts with the design issues and challenges that are associated with implementations of ad hoc network applications. This also includes the various protocols in different layers of the architecture with various security mechanisms and localization techniques.

Pre-requisites: Computer Networks

Learning outcomes: After attending this course students will be capable of

1. Understanding and explaining the concept of ad-hoc and sensor networks, their applications and typical node and network architectures.
2. Understanding and explaining protocol design issues (especially energy-efficiency) and protocol designs for wireless sensor networks
3. Designing and implementing sensor network protocols in the C/Tiny OS environment.
4. Setting up and evaluate measurements of protocol performance in wireless sensor networks.

UNIT – I

Wireless Network Basics and Mobile Ad Hoc Networks [6 Hrs.]
Introduction, Fundamental of wireless communication: Technical Issues, Design Goals, Difference between wired and wireless network. Basics of WLAN 802.11: Network Architecture and Component in typical IEEE 802.11Network, Services Offered by a Typical 208.11 Network. Applications of Ad Hoc wireless Networks, Characteristic of Ad Hoc wireless Network, Issues in Ad Hoc wireless Networks, Infrastructure Based versus Ad Hoc LANs, Cellular versus Ad Hoc Networks, Ad Hoc wireless Internet.

MAC Protocols for Ad Hoc Wireless Network [8 Hrs.]
Introduction, Issues in Designing a MAC Protocol for Ad Hoc wireless networks, Design Goals of a MAC Protocol for Ad Hoc Wireless networks, Classification of MAC Protocol, MAC protocols: A Media Access Protocol For Wireless LAN's (MACAW), Busy tone Multiple Access Protocol(BTMA).

Routing Protocol for Ad Hoc Wireless Network [8 Hrs.]
Introduction, Issues in Designing a Routing Protocol for Ad Hoc wireless networks, Characteristics of an ideal Routing Protocol for Ad Hoc Wireless networks, Classification of Routing Protocol. Proactive routing Protocols: DSDV, WRP, CGSR, STAR. Reactive Routing Protocol: DSR, AODV, TORA, LAR I and II, ABR, SSA, Hybrid routing Protocol.

UNIT –II

Transport Layers Protocol for Ad Hoc Wireless Networks [8 Hrs.]

Introduction, Issues in designing a Transport layer protocol for Ad Hoc Wireless Networks, Design Goals of a Transport layer Protocol for Ad Hoc wireless Networks, Classification of Transport Layer solution, TCP over Ad Hoc Wireless Networks, Feedback-Based TCP (TCP-F).

Security in Ad Hoc Wireless Networks [5 Hrs.]

Introduction, Network Security Requirements, Issues and Challenges in Security Provisioning, Network security Attacks: Network layer Attacks, Transport Layer attacks, Application Layer Attacks , Multi-Layer Attacks, Key management, Secure Routing in Ad Hoc Wireless Network.

Wireless Sensor Networks [5 Hrs.]

Introduction, Application of Sensor Network, and Comparison with Ad Hoc Wireless Network, Issues and Challenges in designing a sensor network, Sensor Network Architecture: Layered architecture, Clustered Architecture. Data Dissemination: Flooding, Gossiping, Rumour Routing, Sequential Assignment Routing (SAR), Direct Diffusion, sensor Protocol for information via Negotiation (SPIN). Location Discovery: Indoor Localization, Sensor Network localization.

Text Book:

1. C. Siva Ram Murty and B. S. Manoj, Ad Hoc Wireless Networks- Architectures and Protocols, Pearson.
2. Xiuzhen Cheng, Xiao H. Huang, Dingzhu Du, Ad Hoc wireless networking, Springer.

Reference Books:

1. Charles E. Perkins Ad Hoc Networking, Addison-Wesley, Pearson Education.
2. C. K. Toh Ad Hoc Mobile Wireless Networks-Protocols and System, Pearson.
3. Prashant Mohapatra and Srikanth Krishnamurty, Ad Hoc Networks-Technologies and Protocols, Springer.
4. Auerbach, Subir Kumar Sarkar, T. G. Basavaraju, C. Puttamadappa, Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Applications, CRC Press.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2536 **(3L + 1 T hrs./week)**

ADVANCED COMPUTER ARCHITECTURE

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: This course focuses on advanced computer architectures such as pipelined and parallel systems. It also emphasizes on the concept of a complete system consisting of asynchronous interactions between concurrently executing hardware components and device driver software in order to illustrate the behaviour of a computer system as a whole.

Pre-requisites: Computer Organization and architecture, operating systems and Data structures.

Learning outcomes: On successful completion of the module, the student should be able to:

1. Understand the advanced concepts of computer architecture.
2. Expose the major differentials of RISC and CISC architectural characteristics.
3. Investigate modern design structures of Pipelined and Multiprocessors systems.
4. Become acquainted with recent computer architectures and I/O devices, as well as the low-level language required to drive/manage these types of advanced hardware.

UNIT – I

Introduction to parallel processing [4 Hrs.]

Criteria for judging the architecture, Architectural classification schemes, Trends towards parallel processing, Parallelism in uniprocessor systems, Parallel computer structure, Applications of parallel processing.

Principles of pipelining [5 Hrs.]

Principles of linear and non-linear pipelining, Classification of pipeline processors, General pipelines and reservation tables, Interleaved memory organization.

Structures and algorithms for array processors [5 Hrs.]

SIMD array processors: SIMD computer organization, Masking and data routing mechanisms, SIMD interconnection networks: static v/s dynamic, Mesh connected ILLIAC network, Barrel shifter network, Shuffle- exchange and Omega network.

Multiprocessor architecture [5 Hrs.]

Functional structures, UMA and NUMA multiprocessors, Interconnection Networks: Time shared or common buses, Bus arbitration algorithm, Cross bar switch and multiport memories, Comparison of multiprocessor interconnection structure, multistage networks for multiprocessors.

UNIT – II

Algorithm analysis [2 Hrs.]

Tree – Data Structure and associated algorithms.

Algorithm design techniques [4 Hrs.]
Parallel processing terminology and Sieve of Eratosthenes, Speed up, Scaled speed up and parallelizability.

Elementary parallel algorithms [4 Hrs.]
Developing algorithms for processor arrays/MIMD computers, Process communication and synchronization on MIMD, Deadlock, Task scheduling on MIMD.

Matrix multiplication [4 Hrs.]
Sequential matrix multiplication, Algorithm for processor array-2D mesh SIMD model, Hypercube SIMD model, Shuffle exchange model, Algorithms for multiprocessors.

The fast Fourier Transform [3 Hrs.]
Hypercube multiprocessor fast Fourier transform.

Sorting [4 Hrs.]
Bitonic merge sort, Bitonic merge sort on-2D mesh network, Hypercube network, Shuffle exchange network.

Text Books:

1. Kaihwang and Faye A. Briggs, Computer Architecture and Parallel Processing, Tata McGraw Hill.
2. Michael J. Quinn, Parallel Computing: Theory and Practice, Tata McGraw Hill.

Reference Books:

1. Kaihwang, Advanced Computer Architecture – Parallelism, Scalability, Programmability, Tata McGraw Hill.
2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C, Pearson.
3. Michael J.Quinn, Parallel Computing Theory and Practice, Tata McGraw Hill.
4. Rajiv Chopra, Advanced Computer Architecture, S. Chand Group.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2537 **(3L + 1 T hrs./week)**

COMPILER DESIGN

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: The course is aimed at offering complete knowledge on compiler design and ends with the development of a working compiler in parts. Topics include compiler structure, symbol tables, regular expressions and languages, finite automata, lexical analysis, context-free languages, LL(1), recursive descent, LALR(1), and LR(1) parsing semantic analysis, and code generation

Pre-requisites: Discrete Structures for Computer Science, Formal Language and Automata Theory and Programming skills.

Learning outcomes: After completing the course, the students will

1. Understand the structure of compilers.
2. Understand the basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code generation.
3. Understand the basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines.
4. Design and implement a compiler using a software engineering approach.
5. Use generators (e.g. Lex and Yacc).
6. Design a recursive-descent parser for any given LL(1) grammar.

UNIT- I

Compiler structure [4 Hrs.]
Analysis-synthesis model of compilation, Various phases of a compiler, Tool based approach to compiler construction.

Lexical analysis [6 Hrs.]
Interface with input, Parser and symbol table, Token, Lexeme and patterns, Regular definition, Transition diagrams, LEX.

Syntax analysis [8 Hrs.]
CFG’s, Ambiguity, Associativity, Precedence, Top down parsing, Recursive descend parsing, Transformation on the grammars, Predictive parsing, Bottom up parsing, Operator precedence grammars, LR parsers (SLR, Canonical, LALR), YACC.

Syntax directed translation [2 Hrs.]
Inherited and synthesized attributes, Dependency graph, Evaluation order, Bottom up evaluation of S- attributed definitions L- attributed definitions and top down translation of attributes.

UNIT-II

Type checking [2 Hrs.]
Type system, Type expressions, Structural and name equivalence of types, Type conversion.

Run time environments [4 Hrs.]
Storage organization, Storage-allocation strategies, Access to nonlocal names, Activation tree, Activation record, Parameter passing, Symbol table and dynamic storage allocation.

Intermediate code generation [8 Hrs.]
Intermediate representations, Translation of declarations, Assignments, Control flow, Boolean expressions and procedure calls.

Code generation [6 Hrs.]
Issues in the design of a code generator, Basic blocks and flow graphs, Next use information, Register allocation, Code generation algorithm, Dag representation of programs, Code generation from dags, Peephole optimization and code generator generators.

Text Books:

1. A.V. Aho, R. Sethi, J.D. Ullman, Compilers: Principles, Techniques and Tools, Wesley.
2. Steven S. Muchnick, Advanced Compiler Design and Implementation, Elsevier.

Reference Books:

1. W. Appel, Modern Compiler Implementation in C: Basic design, Cambridge Press.
2. Fraser and Hanson, A Retargetable C Compiler: Design and Implementation, Wesley.
3. Dhamdhere, Compiler Construction, McMillan.
4. A. V. Aho and J. D. Ullman, Theory of Parsing, Translation and Compiling, PHI.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2538 **(3L + 1 T hrs./week)**

CLOUD COMPUTING

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: This course gives an introduction to cloud computing and its techniques - issues, ecosystem.

Pre-requisites: Operating Systems and Computer Network.

Learning outcomes: On successful completion of this module, learners will be able to

1. Analyse the Service Oriented Architecture and Cloud Computing paradigms.
2. Analyse the enterprise models in cloud computing
3. Evaluate Software as a Service (SaaS) application.
4. Analyse the Security issues associated with the Cloud Computing paradigm.

UNIT–I

Introduction to Cloud Computing [6 Hrs.]
Introduction- Definition, Characteristics, Components, Applications, Pros and cons, Limitations. Need for cloud computing. History/ Evolution of cloud and related technologies- Multi-processing, Distributed computing, Parallel computing to ubiquitous computing. What cloud computing really is and what really isn’t? Importance of cloud computing in current era- why cloud computing matters? Who should use cloud computing and who shouldn't use it? Types of cloud computing. Major players in cloud computing.

Cloud Computing Platforms and Technologies [9 Hrs.]
Migrating into the cloud platform- Issues and deployment considerations. Cloud Reference Model, NIST Cloud model, Exploring cloud service models- IaaS, PaaS, SaaS, IDaaS, STaaS, CaaS, and other service models. Deployment models- Private, Public, Community and Hybrid clouds, Cloud-cube model. Cloud computing platforms- Microsoft Azure, Hadoop, Map-reduce, Amazone Web services (AWS)

Cloud Computing Architecture [5 Hrs.]
Service Level Agreement (SLA), Understanding cloud architecture, Service Oriented Architecture (SOA), Cloud abstraction and virtualization, Cloud federation, Cloud meshup, Cloud life cycle.

UNIT-II

Virtualization [6 Hrs.]
Virtualization- definition, Implementation levels of virtualization, Virtualization architecture and software. Virtual clustering, Virtual Infrastructure, Virtualization of CPU, Memory, I/O Devices, Application virtualization, Virtualization for data-centre automation. Virtualization and cloud

computing, Migrating virtual machines, Pros and cons of virtualization. Para-virtualization, Full-virtualization. Examples- Xen, Microsoft Hyper-V, VMware.

Developing Cloud Services [6 Hrs.]
Web-based application, Pros and cons of cloud service development, Types of cloud service development, cloud development life cycle. Discovering cloud services, Development services and tools, Amazon EC2, Amazon Web services (AWS), Google app engine, IBM blue cloud, Microsoft Azure, Hadoop, Map-reduce, Force.com.

Cloud Security and Privacy [8 Hrs.]
Infrastructure security, Data security, Security management in cloud, Privacy, Security in the cloud, Cloud Information security, Cloud security services, Design principles, Secure Cloud Software Requirements, Cloud Computing Security Challenges. Future of cloud computing.

Text Books:

1. Barrie Sosinsky, Cloud Computing Bible, Wiley.
2. Rajkumar Buyya, James Broberg, AndrzejGoscinski, Cloud Computing– Principles and paradigms, Wiley.
3. Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, Cloud Computing: A Practical Approach, Tata McGraw Hill.
4. Kai Hwang, Geoffrey C Fox, Jack J. Dongarra, Distributed and Cloud Computing: From parallel processing to the Internet of Things, Elsevier.

Reference Books:

1. George Reese, Cloud Application Architectures, O'Reilly.
2. Michael Miller, Cloud computing: Web based applications that change the way you work and collaborate online, Pearson.
3. Lee Gillam, Cloud Computing: Principles, Systems and Applications, Springer.
4. Brian J. S. Chee, Curtis Franklin, Jr., Cloud Computing: Technologies and Strategies of the Ubiquitous Data Center, CRC Press.
5. Haley Beard, Cloud computing best practices for managing and measuring processes for on demand computing, Applications and data centres in the cloud with SLAs, Emereo.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2539 **(3L + 1 T hrs./week)**

CRYPTOGRAPHY AND NETWORK SECURITY

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: This course provides an overview of computer security principles ranging from cryptography to network security. The course will help to learn the principles and practices of computer security in various computing environments. The goal of the course is to provide students with the necessary foundations to apply crypto analysis techniques in new and emerging fields.

Pre-requisites: Computer Networks and Number theory.

Learning outcomes: On successful completion of this module the learner will be able to

1. Identify and evaluate the major types of threats to information security and the associated attacks.
2. Understand the role of cryptography, the techniques for access control and intrusion detection and write code to encrypt and decrypt information using some of the common cryptographic algorithms.
3. Install, configure, and evaluate firewalls.
4. Configure and appraise Intrusion Detection and Response Systems
5. Understand and implement authentication protocols and processes.
6. Understand how digital signatures are performed and the role of digital certificates.
7. Understand how authentication is implemented in wireless systems.

UNIT- I

Cryptography – I [10 Hrs.]

Introduction: Attacks, Services and mechanisms, Security attacks, Security services, A model of internetwork conventional encryption model, Steganography, Classical encryption technique, Simplified DES, Block cipher principles, The data encryption standard, The strength of DES, Differential and linear cryptanalysis algorithms, Triple DES, International data encryption algorithms, Blowfish, Confidentiality using conventional encryption, Placement of encryption function, Traffic confidentiality, Key distribution, Random number generation.

Cryptography-II [10 Hrs.]

Public key cryptography: Principles of Public key cryptography, The RSA algorithm, Key management, Hellman key exchange, Number theory: Prime and relatively prime numbers, Modular arithmetic, Fermat's and Euler's theorems, Testing for primality, Message authentication and Hash functions, Authentication requirements, Authentication functions, Message authentication codes, MD5 message digest algorithm, Digital signature and authentication protocols, Digital signatures, Authentication protocols, Digital signature standard.

UNIT- II

Network Security- I [10 Hrs.]

Authentication applications, Kerberos, X.509 directory authentication service, Electronic mail security, Pretty good privacy, S/MIME, IP security, IP security overview, IP security architecture, Authentication header, Encapsulation security, Payload, Combining security associations, Key management.

Network Security – II [10 Hrs.]

Web security, Web security requirements, Secure socket and transport layer security, Secure electronic transaction, Intruders, Viruses and worms, Intruders, Viruses, and related threats, Firewalls, Firewall design principles, Trusted systems.

Text Books:

1. William Stallings, Cryptography and Network Security, Pearson.
2. Behrouz A Frouzan, Cryptography and Network Security, Tata McGraw Hill.

Reference Books:

1. Richard E. Smith, Internet Cryptography, Pearson.
2. D. Chapman and E. Zwicky, Building Internet Firewalls, O’Reilly.
3. Derek Atkins et al., Internet Security, Professional Reference, Tech media.
4. AtulKahate, Cryptography and Network Security, Tata McGraw Hill.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2540 **(3L + 1 T hrs./week)**

DISTRIBUTED DATABASE SYSTEM

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: This course is designed to present the concepts of heterogeneous multi-computer systems and distributed operating systems. Communication in a client/server model using RPC, Message oriented communications, remote object invocation, and distributed processes and software agents are discussed. Other distributed systems concepts such as clock synchronization, data consistency and replication, fault tolerance, security and distributed component and file systems are also covered.

Pre-requisites: Operating Systems, Data Communication and Computer Networks.

Learning outcomes: A student successfully completing this unit will have

1. Understanding of the complexities of distributed system development
2. Understanding of the goals and architectures of distributed systems
3. Knowledge of important issues in distributed systems, including time, inter-process communication, state management, distributed computing paradigms, middleware and naming
4. Understanding of the middleware technologies that support distributed applications such as RPC, RMI and object-based middleware
5. Ability to apply their knowledge in analysing and designing distributed systems.

UNIT- I

Introduction [4 Hrs.]
Distributed data processing, what is Distributed Database System (DDBS), resembling setups which are not DDBS, examples of DDBS's, Promises of DDBs, case studies such as airline/railway reservation system, banking sector, e-commerce etc.

Distributed Database Architecture [5 Hrs.]
DBMS Standardization, DDBS reference model, Architectural Model – Client Server Architecture and its variants, Multi-database model (MDBMs) Architecture.

Distributed Database Design [5 Hrs.]
Design Strategies, Distribution Design Issues, Fragmentation – Primary Horizontal fragmentation (PHF), Derived Horizontal Fragmentation (DHF), Vertical Fragmentation, Mixed/Hybrid Fragmentation, Access Frequencies and patterns, Replication and Allocation – concepts and strategies.

Overview of Query Processing [5 Hrs.]
Query processing Problem, Objectives, Complexities of Relation Algebra Operators, Distributed query processing operators, Characterization of Query Processor, and Layers of Query Processing.

Transaction Management [3 Hrs.]

Definition of Transaction, Properties, Types (Flat Tx's, Nested Tx's, and Workflows), Architectures.

UNIT- II

Distributed Concurrency Control [5 Hrs.]

Serializability Theory, Taxonomy, Locking Based Algorithm, Timestamp Based Algorithm, Optimistic Concurrency Control algorithm, Deadlock Management.

Distributed DBMS Reliability [5 Hrs.]

Reliability Concepts, Failures and Fault Tolerance Concepts, Failures in DDBMs, Local reliability Protocol, Distributed Reliability Protocols, Dealing with site failures, Network Partitioning.

Object Database Management Systems [5 Hrs.]

Fundamental Object Concepts, Object Distribution Design, Object Management.

Current Issues [5 Hrs.]

Push-Based technology, Mobile Databases, soft computing applications in distributed database system design.

Text Books:

1. M. Tamer Oezsu, Patrick Valduriez Principles of Distributed Database Systems, Prentice Hall.
2. Bell and J. Grimson, Distributed Database Systems, Wesley.

Reference Books:

1. Coulouris, Dollimore and Kindberg, Distributed Systems: Concepts and Designs, Wesley.
2. Ceri, Pellagati, Distributed Database Systems, Tata McGraw Hill.
3. M. Stonebraker, Readings in Database Systems, San Mateo, California: Morgan Kaufmann.
4. Ray, Distributed Database Systems, Pearson.

CA2541 **(3L + 1 T hrs./week)**

AUTONOMOUS MOBILE ROBOTICS AND COMPUTATIONAL INTELLIGENCE

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: The course will cover basic principles of design and practice of intelligent robotics systems including the algorithms for the analysis of the data obtained by vision and range sensors, basic principles of modelling kinematics and dynamics and design of basic control strategies. Emphasize will also be given on formulating interesting robotics tasks and show how they can be accomplished by individual robot or cooperative robot teams (such as flocking, foraging as well as robotic soccer).

Pre-requisites: Design and Analysis of Algorithms, Basic Mathematics (linear algebra, calculus and probability), Soft computing.

Learning Outcomes: After completion of this course the students will be able to

1. Understand concepts of designing robots.
2. Map human intelligence and behaviour to develop intelligent robot.
3. Apply computational intelligence algorithms for programming robot’s behaviour.
4. Explore current trends in robotics research.

UNIT-I

Introduction [4 Hrs.]
Definition, Applications of mobile robotics, History of mobile robotics.

Design of system and navigation architecture [7 Hrs.]
Reference control scheme of a mobile robotics environment, Temporal decomposition of architecture, Control decomposition, Hybrid architecture, Mobile architecture, Perception, Representation and the mapping process.

Locomotion [6 Hrs.]
Issues for locomotion, Legged mobile robots, Wheeled mobile robots.

Kinematics [6 Hrs.]
Kinematics introduction, Forward and reverse kinematics, Wheeled kinematics and its constraints, Mobile system locomotion, Human biped locomotion as a rolling polygon, Representation of robot position through the reference frame.

UNIT-II

Perception [4 Hrs.]
Sensors for mobile robots, Sensor classification, Characterization and sensor performance, Wheeled motor sensor, Ground bases beacon, Active ranging, Motion/Speed sensor, Vision based sensors.

Navigation [3 Hrs.]
Localization overview, Path planning.

Computational intelligence [5 Hrs.]
Swarm intelligence, Evolutionary computation, Artificial immune system, Ant algorithm.

Mobile robot programming [7 Hrs.]
This chapter is included to provide hands on introduction to the field of mobile robotics and various issues in designing and planning of robot work environment. It includes construction and programming of robotic agents using robotic kits and microcontrollers applying concepts of locomotion, perception, navigation and computational intelligent algorithms.

Text Books:

1. Ronald Siegwart, Illah R. Nourbakhsh, Introduction to Autonomous Mobile Robots, MIT Press.
2. Andries P. Engelbrecht , Computational Intelligence, Wiley.

Reference Books:

1. Ronald C. Arkin , Intelligent Robotics and Autonomous Agents, MIT Press.
2. Ulrich Nehmzow, Mobile Robotics: A practical Introduction, Springer-Verlag London.
3. Robin R. Murphy, Introduction to AI Robotics, MIT Press.
4. Leandro N. de Castro and Jonathan Timmis, Artificial Immune system: A new Computational Intelligence Approach, Springer-Verlag.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2542 **(3L + 1 T hrs./week)**

SOFT COMPUTING

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: It comprises of computational techniques like Genetic/ Evolutionary algorithms, Artificial Neural Networks, Fuzzy Systems, Machine learning and probabilistic reasoning etc. This course thoroughly discusses Genetic Algorithms, Artificial Neural Networks (major topologies and learning algorithms) and Fuzzy Logic. At the end of the course, the students will be able to solve a variety of problems in their area of interest ranging from Optimization problems to Pattern recognition and Control tasks, by using soft computing tools.

Pre-requisites: Design and analysis of algorithms, Programming concepts.

Learning outcomes: After completing the course, the students:

1. Acquire knowledge of soft computing theories fundamentals and so they will be able to design program systems using approaches of these theories for solving various real-world problems.
2. Awake the importance of tolerance of imprecision and uncertainty for design of robust and low-cost intelligent machines.

UNIT-I

Introduction [4 hrs.]
Introduction of soft computing, Aspects of soft computing, Dealing with vagueness: Fuzzy systems, Rough sets, Modelling the brain-human cognition, Artificial neural networks, Modelling nature’s optimization process: Natural evolution.

Fuzzy set theory [6 hrs.]
Review of crisp set theory: Sets and subsets, Definitions & notations - Universal set, Null set, Empty set, Subsets, Power set, Venn diagram, Operations on sets: Union, Intersection, Complementation, Difference, Symmetric difference, Cartesian product, Properties of sets: Commutative, Associative, Distributive, De Morgan’s Law, Fuzzy sets: Fuzziness, Vagueness.

Fuzzy membership function [6 hrs.]
Crisp membership, Fuzzy membership, Membership profiles, Fuzzy sets: Definition, Notation, Features, Normality, Height, Support, Core cardinality, Transformation: Normalization, Dilation, Concentration, Contrast intensification, Fuzzification.

Fuzzy set operations [6 hrs.]
Operators – Union, Intersection, Complementation, Equality, Inclusion, Product, Difference, Disjunctive sum, Properties – Commutative, Associative, Distributive, Idempotency, De Morgan’s law, Law Boundary conditions, Law of involution, Transitive law.

UNIT-II

Fuzzy logic [4 hrs.]

Propositional logic: Propositions, Propositional logic well-formed formulae, Properties of wffs, Interpretation of logical expression, Logical equivalence, Tautology, Contradiction, Consistency, Validity of an argument.

Artificial neural networks [4 hrs.]

Basic concepts: The biological neuron, The artificial neuron, Characteristics of the brain, The McCulloch-Pitts neural model, The perceptron neural network architectures: Single layer feed forward ANNs, Multi layer feed forward ANNs.

Back propagation [5 hrs.]

Multilayer feed forward net- structure, Notations, Activation function, Generalized delta rule, The Back propagation Algorithm: Learning, Parameter choice, Initialization, Stopping criteria, Training set, Data representation, Hidden layers.

Advanced search algorithms [5 hrs.]

Genetic algorithms : Natural evolution, Chromosomes, Natural selection , Cross-over, Mutation, Basic GA, Encoding a solution as chromosome , decoding it, Fitness function, Population, GA operators- Selection, Tournament, Roulette wheel, Cross-over, Mutation, GA parameters , Convergence.

Text Books:

1. J.S.R. Jang, C.T. Sun and E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, PHI.
2. Melanie Mitchell, An Introduction to Genetic Algorithms, PHI.

Reference Books:

1. Simon Haykin, Neural Networks – A Comprehensive Foundation, PHI.
2. Jerry M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, PHI.
3. Timothy J. Ross, Fuzzy Logic with Engineering Applications, Wiley.
4. Laurene Fausett, Fundamentals of Neural Networks – Architecture, Algorithms and Applications, Pearson.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2543 **(3L + 1 T hrs./week)**

WIRELESS SENSOR NETWORKS

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: This course provides an introduction to the area of wireless sensor networks. A detailed study on related technologies and standards ranging from networking, OS support and algorithms, to security will constitute the syllabus. Its primary concern will be on protocol design, communication and computational challenges posed by Wireless Sensor based networking systems.

Pre-requisites: Data Communication, Computer Networks, Knowledge of Ad Hoc Wireless Networks desirable.

Learning outcomes: After completing the course the students shall be able to

1. Explain in detail the principles of wireless communication as applied in sensor networks.
2. be familiar with architectures, functions and performance of wireless sensor networks systems and platforms
3. Describe and analyse the specific requirements for applications in wireless sensor networks regarding energy supply, memory, processing and transmission capacity
4. Discuss new applications and particular problems that distinguish wireless sensor networks from mainstream embedded or networked computers.
5. Apply appropriate theory, practices and tools to the development of wireless sensor network applications.
6. Discuss the design principles and implementation of a variety of key sensor networking protocols and algorithms and critically evaluate their effectiveness in a range of practical applications.

UNIT-I

Introduction to Wireless Sensor Networks [6 Hrs.]
Definition, Requirement and Evolution of Wireless Sensor Networks (WSN), Examples of WSNs, Difference between WSNs and Ad Hoc Wireless Networks, IEEE 1451 and Smart Sensors, Transducers and Physical Transduction Principles, Sensors for Smart Environments, Commercially Available Wireless Sensor Systems, Self-Organization and Localization.

Infrastructure Control & Communication Networks [6 Hrs.]
Network Topology, Communication Protocols and Routing, Topology Control, Clustering, Time Synchronization, Power Management, Network Structure and Hierarchical Networks, Historical Development and Standards.

Canonical Problem Localization and Tracking [6 Hrs.]
A Tracking Scenario, Problem Formulation, Distributed Representation and Inference of States, Tracking Multiple Objects, Sensor Models and Performance Comparison and Metrics, Localization and Localization Services.

Signal Processing and Decision Making [2 Hrs.]

Signal Conditioning, Digital Signal Processing, Decision Making and User Interface.

UNIT-II

Networking Sensors and Routing [4 Hrs.]

Introduction, Key Assumptions, Medium Access Control, Issues and Challenges for Routing in WSN, Routing Based on Network Structure, Routing Based on Protocol Operation.

Security for Sensor Networks [8 Hrs.]

Requirement, Issues and Challenges for different security protocols for WSNs, Security Parameters, Sensor Network Limitations, Requirements for Bootstrapping Security in Sensor Networks, Evaluation Metrics, Single Network-Wide Key, Using Asymmetric Cryptography, Pairwise-shared Keys, Bootstrapping Security off a Trusted Base Station.

Sensor Network Databases [6 Hrs.]

Sensor Database Challenges, Query Interfaces, High Level Database Organization, In Network Aggregation, Data Centric Storages, Distributed Hierarchal Aggregation, Temporal Data.

Applications and Future Directions [2 Hrs.]

Emerging Applications, Future Research Directions.

Text Books:

1. Feng Zhao & Leonidas Guibas, Wireless Sensor Networks: An Information Processing Approach, Elsevier.
2. Jun Zheng& Abbas Jamalipour, Wireless Sensor Networks: A Networking Perspective, Wiley.

Reference Books:

1. Waltenegus Dargie & Christian Poellabauer, Fundamentals of Wireless Sensor Networks: Theory and Practice, Wiley.
2. Holger Karl & Andreas Willig, Protocols and Architectures for Wireless Sensor Networks, Wiley.
3. Robert Faludi, Building Wireless Sensor Networks, O’Reilly.
4. Shahin Farahani, ZigBee Wireless Networks and Transceivers, Elsevier.

DEPARTMENT OF COMPUTER APPLICATIONS

CA2544 **(3L + 1 T hrs./week)**

WEB ENGINEERING

No. of questions to be set: 4 each from Unit – I and Unit – II
No. of questions to be answered: Any Five selecting at least TWO from each UNIT

Objectives: Web Engineering course introduces structured methodologies utilized in software engineering to Web development projects. It addresses the concepts, methods, technologies, and techniques of developing Web sites that collect, organize and expose information resources.

Pre-requisites: Web Page Design, Data communication and Network and Software engineering.

UNIT- I

Introduction to Web Engineering [4]
The Web, Web Applications, What Is Web Engineering? Agile, WebE Framework?, Principles to Follow for Adapting the Framework, The Components of Web Engineering, Importance of WebE Process Agility.

Web Engineering Process [5]
Defining the Framework, Incremental Process Flow, Framework Activities, Generic Actions and Tasks for the WebE Framework, Tasks Are Required to Develop an Increment Plan, Modeling and elements of Design Model, Umbrella Activities, Managing the Quality of an Increment, managing Risk.

Communication [5]
The Communication Activity, Formulation, Techniques Use for Communication, Encouraging Collaboration, Elicitation, What Happens Before an Elicitation Session?, Tasks Performed During an Elicitation Session, User Categories for the WebApp, How Are Content and Functional Requirements Identified?, Isolating Constraints and Performance Issues, Use Cases, How Is a Use Case Created?, Identifying WebApp Increments, Negotiation

Planning [6]
Understanding Scope, Relevant Communication Work Products, Refining Framework Activities, Way to Assess Quality, Managing Change, Managing Risk (Identify & Evaluate Risks, Develop Contingency Plans and a schedule), Macroscopic Scheduling, Increment Scheduling, Estimate Effort and Time, Representing Task Interdependencies, Managing Quality, Quality Assurance Mechanisms, Mechanics of a Pair Walkthrough, Mechanics of a Team Walkthrough, Tracking the Project

UNIT- II

The Modelling Activity [5]
Modelling as a Concept, Modelling Frameworks, Modelling Languages, Existing Modelling Approaches, Understanding Analysis in the Context of WebE, Outputs from Analysis, Analysis Tasks to Carry Out, The Content Model, The Structural Elements of the Content Model, Information Exchange, Defining Content Objects, Ways to Depict Content Relationships and Content Hierarchy, Selecting and Representing Analysis Classes for WebApps, The Interaction

Model, The Functional Model, The Configuration Model, Relationship-Navigation Analysis, Establishing Relationships Between Content Objects and Functionality.

WebApp Design [5 Hrs.]

Design for WebApps, Logical Design, Physical Design, Consequence of Design, Design Goals, Design and WebApp Quality, Quality Framework, Design Process, Elements of WebApp Design, Characteristics of the Design Process, Initial Design of the Conceptual Architecture, Initial Design of the Technical Architecture

Interaction Design [5 Hrs.]

Interface Design Principles and Guidelines, Principles Do We Apply to Design Effective Interfaces, Interface Design Workflow, Interface Design Preliminaries, of WebApp Users, Elaborate the Content Objects That Are Identified, Elaborate the Tasks That Are Identified, Design for Different Users with Different Roles, Content Integrated into the Interface Description, Interface Design, Design of Navigation Mechanisms for the Interface, Aesthetic Design, Pleasing Layout.

Information Design [5 Hrs.]

Information Architecture, Elements of an Information Architecture, Characteristics of a Good Information Architecture, Developing an Information Architecture, Organizing Content Structuring the Information Space, Navigation Design: Creating the Detailed Structure How Have Information Design and Navigation Design

Text Books:

1. Roger S.Pressman, David Lowe, Web Engineering, Tata Mcgraw Hill.
2. W. Richard Stevens, TCP/IP Illustrated Volume, I The Protocols, Pearson Addison Wesley.

Reference Books:

1. Jaiswal. S, TCP/IP Principles, Architecture, Protocols & Implementation, Galgotia.
2. Achyut S Godbole and Atul Kahate, Web Technologies, Tata McGraw Hill.
3. Gopalan N P , Akilandeswari Web Technology: A Developers Perspective, PHI.
4. C. Xavier, Web Technology & Design, Tata McGraw Hill.