COURSE SPECIFICATION

Choice based Credit System (CBCS) Scheme and course structure for

Masters in Information Technology 1st semester effective from academic session 2023 and onwards

Course Code	Name of the Subject	Paper Category	Hours/Week			Credits
			L	T	P	
IT23101CR	Database Systems	Core	3	0	2	4
IT23102CR	Open Source Technologies	Core	3	0	2	4
IT23103CR	Programming with C/C++	Core	3	0	2	4
IT23104CR	Fundamentals of Multimedia	Core	2	0	0	2
Discipline Centric Elective (8 credits)						
IT23105DCE	Advanced Computer Architecture	Elective (DCE)	3	1	0	4
IT23106DCE	Data Communications	Elective (DCE)	3	1	0	4
IT23107DCE	Numerical and Statistical Computing	Elective (DCE)	3	1	0	4
IT23001OE	Fundamentals of Information Technology	Elective (OE)	2	0	0	2
					Total Credits	**24**

Choice based Credit System (CBCS) Scheme and course structure for

Masters in Information Technology 2nd semester effective from academic session 2023 and onwards

Course Code	Name of the Subject	Paper Category	Hours/Week			Credits
			L	T	P	
IT23201CR	Java Programming	Core	3	0	2	4
IT23202CR	Data structure using C/C++	Core	3	0	2	4
IT23203CR	Operating Systems	Core	3	1	0	4
IT23204CR	Cloud Computing	Core	2	0	0	2
Discipline Centric Elective (8 credits)						
IT23205DCE	Software Engineering	Elective (DCE)	3	1	0	4
IT23206DCE	Cyber Security	Elective (DCE)	3	1	0	4
IT23207DCE	System Programming	Elective (DCE)	3	1	0	4
IT23002GE	Fundamentals of Network & Internet	Elective (GE)	2	0	0	2
					Total Credits	**24**

Choice based Credit System (CBCS) Scheme and course structure for

Masters in Information Technology 3rd semester effective from academic session 2023 and onwards

Course Code	Name of the Subject	Paper Category	Hours/Week			Credits
			L	T	P	
IT23301CR	Design and Analysis of Algorithms	Core	3	0	2	4
IT23302CR	Python Programming	Core	3	0	2	4
IT23303CR	Dot Net Technologies	Core	3	0	2	4
IT23304CR	Pervasive Computing	Core	2	0	0	2
Discipline Centric Elective (8 credits)						
IT23305DCE	Discrete Mathematics	Elective (DCE)	3	1	0	4
IT23306DCE	Object Oriented Analysis and Design	Elective (DCE)	3	1	0	4
IT23307DCE	Computer Graphics	Elective (DCE)	3	1	0	4
IT23003OE	Project Management	Elective (OE)	2	0	0	2
					Total Credits	**24**

Choice based Credit System (CBCS) Scheme and course structure for

Masters in Information Technology 4th semester effective from academic session 2023 and onwards

Course Code	Name of the Subject	Paper Category	Hours/Week			Credits
			L	T	P	
IT23401CR	Project	Core	0	0	16	8
IT23402CR	Software project/Demo/Dissertation	Core	0	4	0	4
IT23403CR	Internet of Things (IoT)	Core	2	0	0	2
Discipline Centric Elective (8 credits)						
IT23405DCE	Data Warehouse	Elective (DCE)	3	1	0	4
IT23406DCE	Finite Automata and Formal Languages	Elective (DCE)	3	1	0	4
IT23407DCE	Machine Learning	Elective (DCE)	3	1	0	4
IT2300O4GE	Management Information Systems (MIS)	Elective (GE)	2	0	0	2
					Total Credits	**24**

COURSE SPECIFICATION

Choice based Credit System (CBCS)
Scheme and course structure for

Masters in Information Technology 1st semester effective from academic session 2023 and onwards

Course Code	Name of the Subject	Paper Category	Hours/Week			Credits
			L	T	P	
IT23101CR	Database Systems	Core	3	0	2	4
IT23102CR	Open Source Technologies	Core	3	0	2	4
IT23103CR	Programming with C/C++	Core	3	0	2	4
IT23104CR	Fundamentals of Multimedia	Core	2	0	0	2
Discipline Centric Elective (8 credits)						
IT23105DCE	Advanced Computer Architecture	Elective (DCE)	3	1	0	4
IT23106DCE	Data Communications	Elective (DCE)	3	1	0	4
IT23107DCE	Numerical and Statistical Computing	Elective (DCE)	3	1	0	4
IT23001OE	Fundamentals of Information Technology	Elective (OE)	2	0	0	2
Total Credits						**24**

OBJECTIVE AND LEARNING OUTCOME

M.Sc.IT 1st Semester

Course Title: Database Management Systems

Course Code: IT23101CR **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The objectives of this course are to impart a comprehensive understanding of database systems, encompassing their fundamental purpose, architectural components, and diverse database models. It aims to equip students with practical skills in SQL, PLSQL, and database manipulation techniques, enabling them to proficiently handle data. Additionally, the course intends to delve into advanced topics such as data storage, indexing, query optimization, transaction management, and concurrency control, fostering a holistic knowledge of database system applications and development.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Understand the fundamental concepts of database systems and their role in managing data effectively.
2. Demonstrate proficiency in SQL, including data definition, manipulation, and transaction control operations.
3. Gain knowledge of PLSQL architecture, control structures, error handling, and subprograms for database development.
4. Analyse and apply various storage structures, indexing, and hashing techniques for efficient data retrieval.
5. Evaluate query processing and optimization strategies to improve database performance.
6. Comprehend transaction management concepts, including concurrency control and recovery mechanisms.

Course Title: Open Source Technologies

Course Code: IT23102CR **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The course objectives include providing students with a comprehensive understanding of Linux, covering its history, distinctions from Windows and Unix, and practical skills in file management and command-line operations. In subsequent units, the course aims to introduce PHP, focusing on syntax, variable usage, and web development integration. Students will also delve into advanced PHP topics such as input validation, string manipulation, and database interaction with MySQL. By the end of the course, students will be proficient in Linux administration and PHP web development.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Gain expertise in Linux administration, including file management, permissions, and essential command-line operations.
2. Develop proficiency in PHP programming, covering syntax, variables, and the integration of web development concepts.
3. Acquire advanced PHP skills, including input validation, string manipulation, and database interaction using MySQL.
4. Demonstrate the ability to design and manage databases, write SQL queries, and create dynamic web applications using PHP and MySQL.

Course Title: Programming with C/C++

Course Code: IT23103CR **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The overall objective of this course is to provide students with a comprehensive understanding of programming in C and C++, starting with fundamental concepts like data types, control structures, and arrays, and progressing to advanced topics such as functions, pointers, structures, object-oriented programming, and the use of templates and libraries. Through this curriculum, students will develop the skills necessary to design, develop, and maintain efficient and modular software solutions, equipping them with a strong foundation in programming and problem-solving for a variety of applications.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Understand and apply fundamental programming concepts, including data types, control structures, and arrays, to solve basic programming problems.
2. Develop proficiency in using functions, pointers, and structures to create efficient and organized C programs.
3. Gain the ability to work with files, implement macros, and grasp the basics of multi-file programming, enhancing code modularity and reusability.
4. Comprehend object-oriented programming principles, including classes, encapsulation, inheritance, and polymorphism, enabling the development of more complex and modular software solutions.
5. Acquire advanced C++ skills such as virtual functions, templates, exception handling, and the utilization of the Standard Template Library (STL) for efficient and robust software development.

Course Title: Fundamentals of Multimedia

Course Code: IT23104CR **Max. Marks 40**
Credits: 02 **Theory External: 40; Min Marks: 16**
Contact Hrs: 20 **Internal (Continuous Assessment): 10 Marks, Min Marks: 04**

Objectives

The objectives of this course in multimedia are to provide students with a foundational understanding of multimedia technology, encompassing its definition, components, and diverse applications. Additionally, the course aims to equip students with the knowledge and skills required to work with graphics and images effectively, including image properties, file formats, and compression techniques. Lastly, the course seeks to foster a deep comprehension of digital and analog data, as well as the intricacies of digital-to-analog conversion, thereby enabling students to engage with multimedia technology confidently and proficiently.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Introduce the fundamental concepts of multimedia, covering its definition, components, and applications, as well as the features and components of a multimedia system.
2. Explore key aspects of graphics and images, including properties such as resolution, bit depth, and color, and familiarize students with popular image file formats and compression techniques.
3. Foster an understanding of digital and analog data, as well as the processes involved in digital-to-analog conversion, including sampling and quantization, to provide a comprehensive foundation in multimedia technology.

Course Title: Advanced Computer Architecture

Course Code: IT23105DCE **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The objectives of this course in Computer Architecture & Organization are to provide students with a comprehensive understanding of fundamental concepts, including parallel processing architectures, taxonomy, and instruction set structures. Furthermore, the course aims to explore advanced topics such as pipelining, hazards, cache memory, virtual memory, I/O systems, and concurrency, enhancing students' knowledge and skills in computer system design. Lastly, the course delves into specific architectures, including SIMD structures, interconnected networks, data flow graphs, and message passing, enabling students to apply these principles to real-world computing challenges.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Provide students with a foundational knowledge of computer architecture and organization, covering parallel processing architectures and taxonomy, as well as the differences between various processing structures such as SISD, MISD, SIMD, and MIMD.

2. Explore the intricacies of instruction set architectures, including their structures and desirable attributes, while also contrasting complex instruction set computers (CISC) with reduced instruction set computers (RISC).

3. Foster an understanding of advanced concepts like pipelining, hazards, cache memory, virtual memory, I/O systems, and concurrency, emphasizing their significance in designing efficient computer systems.

4. Delve into specific architectural structures and models, including SIMD structures, interconnected networks, data flow graphs, and message passing, preparing students to apply these principles to the design and optimization of multiprocessor systems..

Course Title: Data Communications

Course Code: IT23106DCE **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The overarching objectives of this course in Data Communication are to provide students with a holistic understanding of the field. It starts by introducing fundamental concepts such as key components in data communication systems, communication modes, bandwidth, and channel capacity quantification. It then progresses to cover advanced topics, including local area networks (LANs), wide area networks (WANs), network technologies, transmission media, error detection, and mobile network infrastructure. By the end of the course, students should have a solid grasp of data transmission principles, network architectures, encoding techniques, and mobile communication technologies, enabling them to design, analyze, and troubleshoot data communication systems effectively.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Equip students with fundamental knowledge in data communication, introducing key components and communication modes, as well as concepts like bandwidth, channel capacity, and the quantification of channel capacity through Nyquist and Shannon's Laws.
2. Explore the principles of data transmission, analog and digital signals, and the electromagnetic spectrum, using examples from digital telephone systems to illustrate these concepts and the importance of sampling.
3. Familiarize students with local area network (LAN) and wide area network (WAN) concepts, connecting devices, WAN technologies, and the OSI & TCP-IP model, while also addressing factors influencing transmission media and introducing concepts of ARQ standard, multiplexing, and modulation techniques.
4. Delve into the specifics of guided and wireless transmission media, data encoding, error detection, cellular communication concepts, and mobile network infrastructure, including GSM and CDMA technologies, thereby providing a comprehensive understanding of data communication systems and technologies.

Course Title: Numerical and Statistical Computing

Course Code: IT23107DCE **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The primary objectives of this course in Numerical Methods are to equip students with a strong foundation in computational problem-solving. It begins by introducing fundamental concepts such as numerical representation, error analysis, and error propagation, emphasizing accuracy and precision in numerical computations. It then proceeds to cover a wide range of numerical techniques, including root-finding methods, linear algebraic equation solvers, interpolation, ordinary differential equation solving, and statistical analysis. Through practical implementation and theoretical understanding, the course aims to empower students with the skills necessary to address real-world numerical challenges in scientific, engineering, and data analysis contexts.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Develop a comprehensive understanding of numerical representation, error analysis, and error propagation, allowing for accurate and precise computations in numerical problem-solving.
2. Gain proficiency in implementing numerical methods for root finding, enabling the solution of mathematical equations with practical application.
3. Master various techniques for solving linear algebraic equations, providing the skills to tackle complex mathematical problems and systems of equations.
4. Acquire the ability to interpolate data using polynomial techniques and solve ordinary differential equations using numerical methods, facilitating mathematical modeling and simulation in diverse fields.
5. Learn statistical analysis, including standard deviation, correlation, regression analysis, and hypothesis testing, preparing students to analyze data and draw meaningful conclusions in research and scientific investigations.

Course Title: Fundamentals of Information Technology

Course Code: IT23001OE **Max. Marks 50**
Credits: 04 **Theory External: 40; Min Marks: 16**
Contact Hrs: 20 **Internal (Continuous Assessment): 10 Marks, Min Marks: 04**

Objectives

The course aims to introduce fundamental Information Technology concepts, including data processing, computer classification, hardware components, numerical systems, and operating systems, equipping students with essential knowledge and skills to understand, configure, and interact with computer systems effectively.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Develop a foundational understanding of Information Technology, enabling students to differentiate between data and information and classify various types of computers, from microcomputers to supercomputers.
2. Gain knowledge of computer hardware components, memory types, and input/output devices, facilitating the configuration and maintenance of computer systems.
3. Master number systems and data representation techniques, including binary arithmetic and character encoding, empowering students to work with digital data effectively.
4. Acquire familiarity with operating systems, data organization principles, and software translators, enabling students to navigate and interact with computer systems efficiently while understanding the evolution and significance of various operating systems.

Detailed Syllabus of

1st Semester

Course Code: IT23101CR
Course Title: Database Systems

UNIT I
Database System Applications, Purpose of Database Systems, View of Data, Database Languages, Relational Databases, Database Architecture, Data Mining and Information Retrieval, Database Models and Comparison, Relation Algebra, ER Model, CODDS Rules, Normalization..

UNIT II
Introduction to SQL, Data Types, Data Definition Language, Data Manipulation Language, Transaction Control Language, Integrity Constraints, SQL Functions, Set Operators and Joins, View, Synonym and Index, Sub Queries and Database Objects, Locks and SQL Formatting Commands.

UNIT III
Introduction to PLSQL, Architecture, Data Types, Control Structures, Concept of Error Handling, Cursors and Database Triggers, Subprograms and Packages.

UNIT IV
Data Storage and Querying using various storage structures, Indexing and Hashing, Query Processing and Optimization. Transaction Management Concepts, Concurrency Control and Recovery.

Reference Books:

- William Page,“Using Oracle 9i – Special Edition”, Que/PHI.
- Database System Concepts by A. Silbershatz, H.F. Korth and S. Sudarshan, 6th edition, 1997, McGraw-Hill, International Edition.
- Ivan Bayross, ”SQL & PL/SQL Using Oracle 8i & 9i with SQLJ”, BPB.
- Desai.B , "An introduction to Database Concepts", Galgotia Publications, N.Delhi
- Dates.C , " An introduction to Database Systems", Pearson Education, Asia

Course Code: IT23102CR
Course Title: Open Source Technologies

UNIT I

Introduction to Linux, History, Difference Between Linux and Windows., Difference Between Linux and Unix, Linux is Virus proof, Various Linux Distributions, Pros and Cons Understanding Files and Directories in Linux - File Structure and hierarchy, File Permissions, root, shell, Using VI editor and command associated with it.Basic Commands –mkdir, touch, ls, pwd, cd, chmod, df, du, dd, adduser, passwd, rm, rmdir, date.

UNIT II

Introduction to PHP- History of web programming; how PHP fits into the web environment; installation and configuration, syntax, variables, operators, flow control structures

More language basics; using GET and POST input, working with HTML, forms; built-in and user-defined functions; variable scope; using the PHP manual, getting help

UNIT III

Input validation, string manipulation and regular expression functions; date and time functions, code re-use, require(), include(), and the include_path; filesystem functions and file input and output; file uploads; error handling and logging; sending mail, HTTP headers and output control functions; HTTP cookies; maintaining, state with HTTP sessions; writing simple web clients

UNIT IV

Introducing MySQL; database design concepts; the Structured Query, Language (SQL); communicating with a MySQL backend via the PHP, MySQL API;

References books:

- N. B. Venkateshwarlu (Ed); Introduction to Linux: Installation and Programming, B S Publishers; 2005.
- Matt Welsh, Matthias Kalle Dalheimer, Terry Dawson, and Lar Kaufman, Running Linux, Fourth Edition, O'Reilly Publishers, 2002
- Programming PHP. Rasmus Lerdorf, Kevin Tatroe., (O'Reilly, ISBN 1565926102)
- Learning PHP 5. David Sklar, (O'Reilly, ISBN 0596005601)
- Core PHP Programming. Leon Atkinson, (Prentice Hall, ISBN 0130463469)

Course Code: IT23103CR
Course Title: Programming with C / C++

UNIT I
Data Types, Identifiers, Variables Constants and Literals. Arithmetic Relational Logical and Bitwise. Basic input/output statements, [4L]
Control structures: if-else statement, Nested if statement, Switch statement Loops: while loop, do while, for loop, Nested loops. [4L]
Arrays: Declaration, initialization, 2-dimensional and 3-dimensional array, passing array to function, strings and string functions, and character arrays [4L]

UNIT II
Functions; prototype, passing parameters, storage classes, identifier visibility, Recursive functions. [4L]
Pointers: variables, swapping data, swapping address v/s data, misuse of address operators, pointers and arrays, pointers to pointers, strings, pointer arithmetic, additional operators, portability, pointers to functions, using pointers with arrays , void pointers. [4L]
Structures and unions: syntax and use, members, structures as function arguments, structure pointers, array of structures as arguments, passing array of structure members [4L]

UNIT III
Multi-file programming, File processing in C, Introduction to Macros [2L]
Introduction to object oriented programming, Classes and objects, Abstraction, Encapsulation, Inheritance, Constructor; destructor; Operator overloading; Function overloading; function overriding; friend function; copy constructor; Inheritance: Single , Multiple, and Multilevel Inheritance [10L]

UNIT IV
Virtual function and Polymorphism: Dynamic binding, Static binding; Virtual functions; Pure virtual function; concrete implementation of virtual functions; Dynamic binding call mechanism; Implementation of polymorphism; [8L]
Templates: Function Templates, Class Templates, Member Function Template and Template Arguments, Exception Handling, Standard Template Library, namespaces [4L]

Reference Books:

- Foster and Foster “C by discovery” RRI penram.
- Yashwant Kanetkar “Let us C” PHI.
- E. Balaguruswami “Programming in ANSI C” Tata McGraw Hill.
- Bjarne Stroustrup “The C++ programming language” Pearson Education.
- Herbert Schild “C++ The complete Reference” Tata McGraw Hill.
- Robert Lafore “Object orientation with C++ Programming” Waite Group.

Course Code: IT23104CR
Course Title: Fundamentals of Multimedia

UNIT I

Introduction to Multimedia. Definition, Components of multimedia, Hypermedia, Users of multimedia, Applications of Multimedia; A Multimedia System. Features of a Multimedia System, Storage Devices Graphics Processor (Internal and external). . Digital and Analog Data, Digital to analog conversion (Sampling and Quantization).

UNIT II

Graphics and Images: Image Properties; Resolution, Bit Depth, Color, Aspect Ratio, Dpi , Pixels.

Popular Image File Formats: JPEG, GIF,PNG, TIFF. Basics of Image Capture. Image Compression techniques: lossless and lossy compression. Various compression techniques

References books:

- Fundamentals of Multimedia Ze-Nian Li, Mark S.Drew, Jiangchuan Liu, 2nd Edition (Pearson)
- Multimedia in Practice Judith Jeffcote (PHI).

Course Code: IT23105DCE
Course Title: Advanced Computer Architecture

UNIT I
Computer Architecture & Organization. Basic Parallel Processing Architecture, Taxonomy-SISD. MISD, SIMD, MIMD structures, Serial, Parallel & Concurrent Computation, CISC Vs RISC, Structure of Instruction of instruction sets and Desirable Attributes.

UNIT II
Basic Concepts of pipelining, Instruction Pipelining. Hazards, Reservation Tables, Collision, Latency, Dynamic pipeline, Vector processing & Vector processors. Cache Memory & Virtual Memory: Structure, Analysis & Design. I/O Systems: Design Issues, Performances Measures. Loosely Coupled & Tightly Coupled Systems, Concurrency & Synchronization, Scalability, Models of Consistency, Application of

UNIT III
SIMD Structure- Definition. Types of Interconnected Networks; Baselines, Shuffle-Exchange, Omega, Cuba, Comparison & Application. Mapping Algorithm to array structures, Systolic processors. Mapping design & Optimization, Wave Front Array processor.

UNIT IV
Data Flow Graphs, Petri nets, Static & Dynamic DFA. Different Models, Languages, Compilers, dependency Analysis. Message Passing, Program mapping to Multiprocessors, Synchronization

References:

- A Quantitative Approach by David A. Patterson, John L. Hennessy, David Goldberg
- 2nd Edition Hardcover, 760 pages Morgan Kaufman Publishers Publication date: January 1996
- A Hardware/Software Approach David Culler and J.P. Singh with Anoop Gupta
- Solutions to Selected Exercises in Computer Architecture a Quantitative Approach by Thomas E. Willis, Allan D. Knies, Paperback Published by Morgan Kaufman Publishers Publication date: October 1996 ISBN: 1558604065
- High-performance Computer Architecture (3rd edition), by Harold Stone ,Addison Wesley.
- Computer Architecture: Pipelined and Parallel Processor Design by Michael J Flynn. Jones and Bartlett Publishers, 1995..

Course Code: IT23106DCE
Course Title: Data Communications

UNIT I

Introduction to Data Communication, Key components in data communications systems. Simplified model, different modes of communication. Bandwidth and Channel Capacity. Quantifying Channel Capacity for noiseless channel (Nyquist Law) and noisy channel (Shannon's Law). Example of a digital telephone system to explain basic concepts of analog signals, digital signals, sampling. Nyquist Sampling. Data transmission concepts. Analog and Digital Signals. Electromagnetic Spectrum.

UNIT II

Local area network (LAN) concepts and characteristics. Network Connecting Devices, Wide area networks (WANs). WAN technologies (traditional packet and circuit switching, ISDN, Overview of the OSI & TCP-IP model. Transmission media – factors affecting distance and data rate. Concept of ARQ standard and its versions. Concept of Multiplexing. FDM. & TDM.

UNIT III

Guided transmission media: Twisted-Pair, Co-axial Cable. Optical Fiber Technology. Wireless Transmission concepts; antenna system, antenna gain and Wireless Propagation. Data encoding. NRZ-L, NRZ-I encoding. Multilevel Binary and Biphase Coding techniques and their implementations. ASK, FSK, PSK and QPSK. PCM concepts: sampling, quantization. Delta Modulation.

UNIT IV

Error detection: Parity-based, CRC-based. Concept of cells, Cell Geometry, sectorization, coverage area, co-channel interference, frequency-reuse handoffs. GSM Mobile Network Infrastructure: MS, BTS, BSC, MSC, VLR, HLR. CDMA-Technology, IS-95 CDMA System overview.

Reference Books:

- William Stallings, "Data and Computer Communications", Pearson Education
- Andrew Tanenbaum, "Computer Networks", Pearson Education 4/e.
- Ulysses Black, "Principles of Data Communications ", PHI.
- Morley, Gelber, "The Emerging Digital Future", Addison-Wesley.

Course Code: IT23107DCE
Course Title: Numerical and Statistical Computing

UNIT I

Introduction. Requirements for computer-oriented solutions to numerical problems. Different Number Systems, Floating Point Representation and Arithmetic, Normalized Floating Point Representation of Numbers. Approximations & Errors – Types of Programming Errors, Computer & Arithmetic Errors, Accuracy and Precision, Round off and Truncation Errors. Propagation of Error. [6L]

Algorithms to Compute Roots of Equation – Methods of Tabulation or Brute Force Method, Method of Bisection, Secant Method, Newton-Raphson Method, Method for False Position. Implementations of these methods.[6L]

UNIT II

Linear Equations, Types of Methods to find solutions to linear equations. Algorithms to Solve Linear Algebraic Equations: Gauss Elimination, Gauss Jordan, Gauss Seidel, L.U. Decomposition, Pivoting [12L]

UNIT III

Lagrange Interpolated Polynomial, Newton Divided Differences Interpolating Polynomial. Implementation of these methods. [4L]

Algorithms to solve Ordinary Differential Equations – Euler Method and Modification. The trapezoidal Rule, Simpson's Rule. R-K Method. Implementation of these methods. [8L]

UNIT IV

Standard Deviation, Correlation, Regression Analysis, Least Square Approximation of Functions: Linear Regression and Polynomial Regression, Concept of Hypothesis, Statistical Tests: Chi-Square Test, Student t-Test, f-Test. [12L]

Reference books:

- S.C.Chapra & R.P.Canale: “Numerical methods for Engineering”. Tata McGraw Hill.
- Krishenmurty and Sen : “Numerical Algorithms”
- V. Rajaraman “Computer oriented numerical methods.” Prentice Hall of India.
- McCalla, Thomas Richard: “Introduction to Numerical Methods and FORTRAN Programming”, John Wiley & Sons, Inc.
- Grewal, B. S.: “Higher Engineering Mathematics”, Hindustan Offset Problems Series.
- “SCHAUM’S Solved Problems Series”.
- Sharma, K. D.:“Programming in Fortran”.
- Jain, M. K., Iyengav, S. R. K., Jain, R. K.: “Numerical Methods for Scientific and Engineering Computation”+, Wiley Eastern Ltd, New Delhi

Course Code: IT23001OE
Course Title: Fundamentals of Information Technology

UNIT I:
Introduction to Information Technology: Basic concepts of IT, Data Processing: Data and Information. Introduction to Computers: Classification, History, Types of Computers; Mini Computers, Micro Computers, Mainframe Computers, Super Computers. Applications of Information Technology:
Block Diagram of The Computer System, Introduction to various units. Hardware: CPU, Memory, Input and Output devices, Auxiliary storage devices. Software: System and Application Software, Utility packages, Configuration of Computer System. Types of Memory RAM, ROM, PROM, EPROM. Secondary Storage Devices (FD, CD, HD, Pen drive). I/O Devices: Scanners Digitizers Plotters, LCD, Plasma Display etc.

UNIT II
CPU type and speed; memory: capacity, type, word size, speed. Hard drive: capacity, speed; fire wire, expansion slots, ports. Number System. Binary, octal and hexadecimal number systems; binary addition and subtraction. Integers (positive and negative): sign and magnitude,
BCD, two's complement; representation of characters, ASCII.
Operating System: History, evolution and structures, Introduction to MS-DOS/WINDOWS/LINUX/UNIX. Data Organization: Drives, Files, Directories. Translators: Assembler, Compiler and Interpreter.

Recomended Books:

- Raja Raman V., "Fundamental of Computers" (4th edition.), Prentice Hall of India, New Delhi.
- Trainer T., et al, "Computers", McGraw Hill.
- Norton, Peter, "Introduction to Computers, Mc-Graw-Hill.
- S.Jaiswal, "Fundamental of Computer & IT", Wiley dreamtech India.

Scheme and course structure for

Masters in Information Technology 2nd semester effective from academic session 2023 and onwards

Course Code	Name of the Subject	Paper Category	Hours/Week			Credits
			L	T	P	
IT23201CR	Java Programming	Core	3	0	2	4
IT23202CR	Data structure using C/C++	Core	3	0	2	4
IT23203CR	Operating Systems	Core	3	1	0	4
IT23204CR	Cloud Computing	Core	2	0	0	2
Discipline Centric Elective (8 credits)						
IT23205DCE	Software Engineering	Elective (DCE)	3	1	0	4
IT23206DCE	Cyber Security	Elective (DCE)	3	1	0	4
IT23207DCE	System Programming	Elective (DCE)	3	1	0	4
IT23002GE	Fundamentals of Network & Internet	Elective (GE)	2	0	0	2
Total Credits						**24**

2nd Semester
OBJECTIVE AND LEARNING OUTCOME

M.Sc.IT 2nd Semester

Course Title: Java Programming

Course Code: IT23201CR **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives:
The primary objective of this course is to provide students with a comprehensive understanding of Java programming, including its fundamental features, object-oriented programming concepts, and practical application in various domains. Students will gain proficiency in Java syntax, exception handling, multi-threading, networking, graphical user interface (GUI) development using Swing, and database connectivity through JDBC.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Java Proficiency: Develop a strong foundation in Java programming, covering data types, variables, operators, control statements, and object-oriented concepts like classes, methods, and interfaces.
2. Exception Handling and IO: Master exception handling mechanisms, including try-catch-finally, and understand IO operations with input and output streams.
3. Multithreading and Networking: Acquire knowledge of multithreading concepts, thread synchronization, and networking in Java, enabling the development of efficient, concurrent, and networked applications.
4. Graphical User Interface and Database Connectivity: Learn to create graphical user interfaces using Swing, handle events, and implement layout managers. Additionally, gain skills in Java Database Connectivity (JDBC) to interact with databases, execute SQL statements, and work with ResultSet, including storing and retrieving images from a database

---

Course Title: Data Structure Using C/C++

Course Code: IT23202CR **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The primary objective of this course is to provide students with a comprehensive understanding of data structures and algorithms. It begins with an introduction to problem-solving and primitive data structures, emphasizing efficient operations and recursion. Students will then explore advanced topics such as linked lists, stacks, queues, searching techniques, binary trees, sorting algorithms, hash tables, and file structures, with practical applications in text editing, lexical analysis, and data organization.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Data Structures Proficiency: Develop a strong foundation in data structures, including primitive data structures and their efficient operations, as well as advanced structures like linked lists, stacks, and queues, both in array-based and linked list-based implementations.

2. Algorithms and Searching: Acquire the ability to implement and apply various searching techniques, including sequential and binary search, binary trees, and binary search trees (BSTs). Understand the concepts of height balance and AVL trees for efficient data retrieval.

3. Sorting and Hashing: Master sorting techniques like insertion sort, selection sort, quick sort, and heap sort, along with external sorting strategies. Learn the fundamentals of hash functions, hash tables, and hashing with chaining for efficient data storage and retrieval.

4. File Structures: Gain knowledge of different file structures, including sequential files, indexed files, and direct files. Understand the k-Way merge strategy for external sorting, enabling efficient management and retrieval of data in various applications.

Course Title: Operating Systems

Course Code: IT23203CR **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The primary objectives of this course in Operating Systems are to provide students with a comprehensive understanding of the fundamental concepts and principles behind operating systems. It begins with an introduction to the definition, purpose, and evolution of operating systems, emphasizing their services, functions, and the concept of virtual memory. The course covers resource management, architecture, and the goals and structures of operating systems, as well as their interaction with hardware through system calls. Students will gain knowledge of process management, concurrency, deadlock prevention, memory management, virtual memory, and I/O management, enabling them to understand, design, and analyze operating systems effectively.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Operating System Fundamentals: Develop a deep understanding of operating system concepts, services, and functions, including virtual memory management and resource allocation, laying the foundation for effective operating system design and operation.
2. Process and Thread Management: Master the principles of CPU scheduling, including scheduling criteria and various scheduling algorithms. Understand the concepts of processes, threads, and real-time scheduling, enabling efficient resource utilization in multi-process environments.
3. Concurrency and Memory Management: Acquire knowledge of concurrency principles, mutual exclusion, and deadlock prevention, as well as memory management requirements, partitioning, allocation strategies, segmentation, and paging. Develop the skills to manage system resources efficiently and handle concurrency challenges.
4. I/O Management and Disk Scheduling: Learn about I/O devices, system organization, I/O buffering, and disk scheduling, enabling students to optimize I/O operations and effectively manage data storage and retrieval in operating systems.

Course Title: Cloud Computing

Course Code: IT23104CR **Max. Marks 40**
Credits: 02 **Theory External: 40; Min Marks: 16**
Contact Hrs: 20 **Internal (Continuous Assessment): 10 Marks, Min Marks: 04**

Objectives

The primary objectives of this course in Cloud Computing are to provide students with a comprehensive understanding of cloud computing fundamentals and services. It begins by introducing the history of cloud computing, cloud architecture, storage, and various cloud service models, emphasizing the advantages and disadvantages of cloud computing. The course covers major components of cloud computing, including public, private, and hybrid clouds. Additionally, it explores the need for web-based applications, cloud service development, development types, and associated tools. Students will gain insights into cloud economics, security issues, threat solutions, and disaster recovery, enabling them to evaluate, develop, and secure cloud-based solutions effectively.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Cloud Computing Fundamentals: Develop a strong understanding of the fundamentals of cloud computing, including its history, architecture, storage, and various service models. Identify the pros and cons of cloud computing and assess the benefits it offers.

2. Cloud Service Development: Acquire the skills needed for web-based application development in cloud environments, including an understanding of development types and tools. Learn about cloud economics, economies of scale, and how to choose the right cloud platform for specific needs.

3. Security and Disaster Recovery: Understand security issues in cloud computing and explore solutions to mitigate threats. Gain knowledge of disaster recovery strategies to ensure data and service availability in cloud-based systems.

4. Cloud Service Models: Comprehend the major components of cloud computing, including public, private, and hybrid clouds. Evaluate and select appropriate cloud service models for different applications, enabling efficient utilization of cloud resources..

Course Title: Software Engineering

Course Code: IT23205DCE **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The primary objectives of this course in Software Engineering are to provide students with a comprehensive understanding of software engineering principles, processes, and methodologies. It begins by introducing the concept of software engineering, its evolution, myths, and importance, while addressing the challenges such as scale, quality, productivity, consistency, and repeatability. The course covers software process management, including frameworks, process models, estimation, scheduling, staffing, and risk management. Additionally, it explores software requirement analysis, specification, design engineering principles, and Computer Aided Software Engineering (CASE) tools. Students will gain insights into various software engineering concepts, methodologies, and practices.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Software Engineering Fundamentals: Develop a solid understanding of software engineering fundamentals, including the evolving role of software, software characteristics, challenges, and standards, enabling students to appreciate the importance of software engineering in modern applications.

2. Software Process Management: Acquire knowledge of software process management, including phase frameworks, process models, estimation techniques, and risk management strategies. Understand the principles of process planning and staffing for successful software project management.

3. Software Requirements and Design: Learn the essentials of software requirement analysis and specification, encompassing problem analysis, requirement validation, and the use of methodologies such as data flow modeling and prototyping. Gain proficiency in software design principles, methodologies, and verification techniques, both function-oriented and object-oriented.

4. Computer Aided Software Engineering (CASE): Understand the concept and scope of CASE tools in the software development life cycle, including their role in documentation and project management. Explore how CASE tools support software engineering practices and enhance the efficiency and quality of software development projects.

Course Title: Cyber Security

Course Code: IT23206DCE **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The primary objectives of this course in Cybersecurity are to provide students with a comprehensive understanding of the field, encompassing cyber, network, and information security. It begins with an introduction to cybersecurity, its terminology, and critical concepts such as the CIA Triad. The course covers a wide range of topics, including malware, threats, attacks, attack prevention, detection, and mitigation. It also explores security testing, penetration testing, digital forensics, and the development of cybersecurity policies and management strategies. Students will gain practical knowledge of cybersecurity tools, techniques, and case studies.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Cybersecurity Fundamentals: Develop a solid understanding of cybersecurity, network security, and information security, including the CIA Triad. Explore key terminologies such as threats, risks, vulnerabilities, exploits, and various types of attacks.

2. Malware, Threats, and Attacks: Acquire knowledge of malware types, threat scenarios, and attack techniques. Understand how to detect and mitigate threats such as spam, website defacement, identity theft, and DDoS attacks. Analyze case studies like STUXNET and Salami Attacks.

3. Security Tools and Techniques: Learn about cybersecurity tools and techniques for prevention, detection, and mitigation, including packet filtering, firewalls, VPNs, and intrusion detection systems like Snort. Gain practical insights into security testing, penetration testing, and digital forensics.

4. Cybersecurity Policy and Risk Management: Understand the importance of cybersecurity policies, management plans, and risk management in protecting critical IT infrastructure. Explore national cybersecurity policies and strategies, preparing students to address cybersecurity challenges effectively.

Course Title: System Programming

Course Code: IT23207DCE **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The primary objectives of this course in Systems Programming and Compiler Construction are to provide students with a comprehensive understanding of the design and operation of computer systems, programming languages, and compiler construction. It begins with an introduction to machine structures, the evolution of programming system components, and general machine structure concepts. The course covers assembly language, assemblers, loaders, linkers, macros, formal systems, and programming languages. Students will gain practical knowledge of the design and implementation of system components and compilers.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Machine Structure and Language: Develop a strong understanding of computer machine structures, including their evolution and the components of programming systems. Acquire proficiency in machine and assembly languages, enabling low-level programming and system interaction.

2. Assembler and Loader Design: Learn the principles and design of assemblers, including single-pass and multi-pass algorithms. Understand loader schemes, subroutine linkages, and the design of various types of loaders, including absolute and direct linking loaders.

3. Formal Systems and Programming Languages: Gain knowledge of formal systems, language hierarchies, BNF notation, and canonic systems. Explore the phases of compiler construction, including lexical analysis, syntax parsing, interpretation, optimization, storage allocation, code generation, and assembly.

4. Compiler Construction: Understand the intricacies of compiler construction, including the statement of problems and the organization of compiler phases. Learn about the passes of a compiler and how each phase contributes to the generation of executable code from high-level programming languages.

Course Title: Fundamentals of Network & Internet

Course Code: IT23002GE **Max. Marks 50**
Credits: 04 **Theory External: 40; Min Marks: 16**
Contact Hrs: 20 **Internal (Continuous Assessment): 10 Marks, Min Marks: 04**

Objectives

The primary objectives of this course in Data Communication and Networking are to provide students with a comprehensive understanding of data communication principles and network technologies. It begins with the definition of data communication, network communication modes, and types of networks, including local area networks (LANs), wide area networks (WANs), and metropolitan area networks (MANs). The course covers networking devices, topologies, addressing, protocols, transmission media, and network models. Students will gain practical knowledge of networking technologies, including wired and wireless media, the client/server model, Internet concepts, and information security basics.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Data Communication Fundamentals: Develop a solid understanding of data communication and network principles, including communication modes (simplex, duplex, half-duplex) and network types (LAN, WAN, MAN). Explore the data communication model and examples to comprehend real-world applications.

2. Networking Devices and Technologies: Acquire knowledge of networking devices such as repeaters, hubs, switches, bridges, routers, and gateways. Understand network topologies, including bus, star, ring, mesh, tree, and hybrid configurations.

3. Transmission Media and Internet Concepts: Learn about various transmission media, both wired (coaxial, fiber-optic, twisted pair) and wireless (satellite, microwave, infrared). Explore wireless network technologies like Bluetooth and Wi-Fi, and grasp concepts such as Internet, Intranet, and bandwidth.

4. Client/Server Model and Internet Services: Gain proficiency in the client/server model, protocols, and Internet services, including mail servers, chat servers, HTTP, FTP, World Wide Web, web browsing, websites, and various online platforms like social networking sites, e-commerce, e-learning, pod-casting, and VoIP. Understand the basics of information security.

Detailed Syllabus of

2nd Semester

Course Code: IT23201CR
Course Title: Java Programming

UNIT I

Features of Java, OOPs concepts, Java virtual machine, Byte code interpretation, JDK installation, PATH variable, Data types, variable, operators in Java, control statements in java, Arrays: Single, multi-dimensional and jagged arrays, Objects and classes, Methods in Java, Abstract classes, Inner classes, Wrapper classes , Interfaces, Keywords like super, this and final. [12L]

UNIT II

Method overloading and overriding, Constructor overloading and chaining. Strings: Mutable and immutable, StringBuffer and String Tokenizer, Operations on Strings, Packages: Creating and importing packages, Exception handling - Exception as objects , Exception hierarchy, Try catch finally, Throw, throws , Chained Exceptions [10L]
IO package - Input streams, Output streams, Sample programs on IO files [2L]

UNIT III

Multi-threading: Thread Life cycle, Multi-threading advantages and issues, creating multiple threads, thread priorities, Daemon threads, Inter-Process communication (IPC), Thread synchronization [8L]
Networking in Java: TCP/IP Server Sockets in Java, Simple networking applications for two-way communication between client and server. [4L]

UNIT IV

Graphics Programming with Swing, creating GUI for basic applications, event delegation model, handling events, Layout managers, using a layout inside another layout [8L]
JDBC architecture, establishing connectivity and working with connection interface, working with statements, Creating and executing SQL statements, Working with ResultSet, Storing and retrieving images from database. [4L]

References books:

1. Programming with Java A Primer, E.Balaguruswamy Tata McGraw Hill Companies
2. H. Schildt, Java: The Complete Reference, Tata McGraw Hill
3. K. Sierra and B. Bates, Head First Java , O'Reilly, 3rd Edition
4. Big Java, Cay Horstmann 2nd edition, Wiley India Edition
5. Core Java: An Integrated Approach, R. Nageswara Rao, 2016

Course Code: IT23202CR
Course Title: Data Structures using C/C++

UNIT I

Introduction: Structure and problem solving, Primitive Data Structures: Operation On primitive Data Structures, Recursion Function and its examples.
String Manipulation, String Matching Techniques &Applications (examples: Text Editing, Lexical Analysis).

UNIT II

Concept of Stack and Queue.Singly and Doubly-Linked Lists, Circular Linked List, their implementation and comparison. Array based and Linked List based Implementation of stack and Queues and their applications.

UNIT III

Searching: Sequential and Binary Search on Array-based ordered lists. Binary Trees, their implementation and traversal. Binary Search Trees: Searching, Insertion and Deletion of nodes. Height Balance and Concept of AVL Trees. Concept and purpose of B-Trees.

UNIT IV

Concept of Hash Functions, Hash-tables and Hashing with Chaining. Sorting Techniques: Insertion Sort, Selection Sort, Quick Sort, Heap Sort. External Sorting: k-Way Merge Strategy. File Structure: Sequential Files, Indexed Files, Direct Files.

Text books:

- Trebley and Sorenson: “An Introduction to Data Structures with Applications” McGraw Hill,
- Kongakusha 1976.
- Horowitz and Sahni: “Data Structures” SBCS Publication, 1980.
- Michael J.Folk et al “File Structure an Object Oriented Approach with C++.
- Data Structures Using C – Aaron Tenenbaum

Reference books:

- Fundamentals of data structures – Ellis Horowitz and SartajSahni
- Data Structures Files and Algorithms – Abhay K. Abhyankar
- Data Structures and Algorithms – Alfred V. Aho, John E. Hopcroft,Jeffrey D. Ullman (Pearson Education)

Course Code: IT23203CR
Course Title: Operating Systems

UNIT I

Introduction: Definition and purpose of operating systems, Types of operating systems, Evolution of operating systems. Operating System services and functions, system programs, virtual memory. Overview Of An Operating System, Resource Management, Operating System Interface, Architecture, Goals & Structures of O.S, Basic functions,Interaction of O. S. & hardware architecture, System calls.

UNIT II

Process Management: CPU Scheduling: Concepts, CPU Scheduler, Scheduling Criteria, Scheduling Algorithms, FCFS, SJFS, Priority Scheduling, Round-Robin Scheduling, Multi-processor Scheduling, Real-Time Scheduling, Algorithm Evaluation, Process Scheduling Models. Threads: Thread Scheduling, Real Time Scheduling.

UNIT 3

Concurrency: Principles of Concurrency, Mutual Exclusion, Semaphores, pipes, Message Passing, signals, Monitors. Deadlock: Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection

Memory Management: Memory Management requirements, Memory partitioning: Fixed and Variable Partitioning, Memory Allocation: Allocation Strategies (First Fit, Best Fit, and Worst Fit), Fragmentation, Swapping, and Paging. Segmentation, Demand paging

UNIT 4

Virtual Memory: Concepts, management of VM, Page Replacement Policies (FIFO, LRU, Optimal, & Other Strategies), Thrashing.

I/O management & Disk scheduling: I/O Devices, Organization of I/O functions, Operating SystemDesign issues, I/O Buffering, Disk Scheduling.

Reference books:

- Dietel, H.M. “An introduction to operating system” Pearson Education, 2/e.
- Milenkovic.M. “An Operating System – Concepts &Design”.McGraw Hill International
- Education Computer science series 1992.
- Peterson. J.L.AbharamSilberschatz. “Operating System Concepts”. John wiley ,1989.
- Tananbum, A. S. “Modern Operating System”, Pearson Education.
- Karnetkar, “Unix Shell Programming”, BPB.

Course Code: IT23204CR
Course Title: Cloud Computing

UNIT I

Fundamentals of Cloud Computing

Fundamentals – Short history of cloud computing – Cloud Architecture – Cloud Storage – Cloud Service – Pros and Cons of cloud computing – Benefits from cloud computing. Major components of cloud computing, Public, private and hybrid cloud

UNIT II

Cloud Services

Need for Web-Based Application – The cloud Service Development – Cloud Service, Development Types – Cloud Service development tools.

Cloud Economics: Economics of private cloud, Economies of scale, Economics of choosing a cloud platform,

Security issues, solution to threats, disaster recovery

Text books

- Joshy Joseph & Craig Fellenstein, "Grid Computing", Pearson Education, 2004.
- Michael Miller, "Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online", Que, 2008.

References books:

- Fran Berman, Geoffrey Fox, J.G. Anthony Hey, "Grid Computing: Making the Global Infrastructure a reality", John Wiley & sons, 2003.
- Hmar Abbas, "Grid Computing: A Practical Guide to technology and Application Charles River media, 2003.

Course Code: IT23205DCE
Course Title: Software Engineering

Unit I

Introduction: Software engineering, Evolving role of software, Concept of software, Changing nature of software, Software Myths, Software Importance, Characteristics, Software Components, Software crises, Software Engineering Challenges (Scale, Quality Productivity, Consistency and Repeatability, Change), Software standard, Software Engineering approach.

Unit II

Software Process Management: Software process, phase's framework, capability maturity model integration (CMMI), Process patterns, process assessment, personal and team process models (PSP, TSP) process technology, characteristics of software process

Introduction to software process models waterfall, incremental process models, Evolutionary process model. Process Planning, Estimation, COCOMO Model, Project Scheduling and staffing Risk management (concepts, Risk assessment, and Risk control)

Unit III

Introduction to Software Requirement Analysis and Specification: software requirement, (need for SRS requirement process), problem analysis (informal approach, data flow modeling, object –Oriented modeling, prototyping), requirement specification (characteristics, components), Concept of Use Cases, Concept of validation.

Unit IV

Design Engineering: Function oriented design, Design principles, Coupling and Cohesion, Design Notations & Specifications, Structured Design Methodology; Object-Oriented Design, OO Concepts, Design Concepts, Design Methodology (Dynamic & Functional Modeling), Design Verification.

CASE (Computer Aided Software Engineering): Concept, scope, CASE Support in Software Life Cycle, Documentation, Project management.

Reference books:

- ROGER S. PRESSMAN - Software Engineering - A Practitioner's Approach, Sixth edition,
- PankajJalote - An Integrated approach to Software Engineering, 3rd edition, Narosa Publication.
- Sommmerville - Software Engineering. Pearson , 7/e , 2006. SCHAUM'S Outlines, TMH.
- JAMES F. PETERS Software

Course Code: IT23206DCE
Course Title: Cyber Security

UNIT I

Introduction to Cyber security, Network Security and Information Security, CIA Triad Cyber security terminologies - threat, risk, vulnerability, exploit, Attack, attack surface, attack tree, hacking and its types. Non-state actors, Cyber-terrorism, Critical IT and National Critical Infrastructure, Cyber warfare, Cyber Crime, Cyber Law, Social media- data privacy and security issues. Overview of IT Act-2000. Case Studies.

UNIT II

Understanding Malware: Brain, Morris Worm, Trojan Horse, Malware Detection. Understanding Threats: Spam, website defacement, Impersonation on Social Networking Sites, Identity Theft, Data Theft, Information Extortion, Understanding Attacks: Malicious Code, Back Doors, Password Crack, Brute Force, Dictionary, DoS and DDoS Attack (TCP-SYN, UDP Flooding, BOTNET), Spoofing, Man-in-the-Middle, Mail Bombing, Sniffers, Social Engineering, Pharming. Time Bomb, Case Study : STUXNET and Salami Attack.

UNIT III

Attack Prevention, Detection and Mitigation. Packet Filtering: Firewall Basics, How a Firewall Protects a Network, Packet Characteristic to Filter, Network Address Translation (NAT) and Port Forwarding. VPN: Virtual Private Networks. Snort: Introduction Detection System. Case Study: VPN

UNIT IV

Darknet, Security Testing and Penetration Testing, Overview of Kali Linux and basic attacking tools. Digital Forensics, Operating System Forensics, cyber security policy and plan, management plan., Risk Management, National cyber security policy and strategy.

Reference books:

- Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives by Sumit Belapure and Nina Godbole, Wiley India Pvt. Ltd.
- Information Warfare and Security by Dorothy F. Denning, Addison Wesley.
- Security in the Digital Age: Social Media Security Threats and Vulnerabilities by Henry A. Oliver, Create Space Independent Publishing Platform.
- Data Privacy Principles and Practice by Natraj Venkataramanan and Ashwin Shriram, CRC Press.
- Information Security Governance, Guidance for Information Security Managers by W. KragBrothy, 1st Edition, Wiley Publication.
- Auditing IT Infrastructures for Compliance By Martin Weiss, Michael G. Solomon, 2nd Edition, Jones Bartlett Learning.

Course Code: IT23207DCE
Course Title: System Programming

UNIT I

Introduction, Machine Structure , Evolution of the Components of programming system Evolution of Operating Systems, General Machine Structure , General Approach to a New Machine , Machine Structure 360-370, Machine Language. Assembly Language

UNIT II

General Design Procedure, Assemblers, Design of a single pass assembler and multi pass assembler, Macros: two pass algorithm, single pass algorithm, Implementation of macro calls within macros.

Loaders and Linkers, Loader Schemes, subroutine linkages, relocating loaders, linking loaders, Design of an absolute loader, Design of a direct linking loader.

UNIT III

Formal Systems and Programming Languages: Formal specification, Hierarchy of Languages, BNF, Canonic Systems and Formal Systems.

Compilers, Statement of problem, phases of Compiler-Lexical phase, syntax phase Interpretation phase, optimization, storage assignment code generation and assembly phase, Passes of a compiler.

Reference Books:

- John J. Donovan, “Systems Programming”, Tata McGrawHill
- Leland L.Beck."System Software" 4th edition Pearson 1997 Barron.D.W."Assemblers and Loaders" Mc Donald and Javes 1978
- Ullman.J.D."Fundamentals of Programming System" Addison and Wesley D.M.Dhamdhere."System Programming and Operating Systems"2nd edition

Course Code: IT23002GE
Course Title: Fundamentals of Network and Internet

UNIT I

Definitions of data communication and network; communication modes: (simplex, duplex, half duplex), point-to-point, and broadcast. Analog & Digital signal, types of networks: (local area network, wide area network, metropolitan area network). A Data Communication Model and Examples.

Networking Devices: Repeaters, Hubs, Switches, Bridges, Routers, and Gateways. Network Topologies; Bus, Star, Ring, Mesh, Tree and Hybrid.

UNIT II

A Computer network diagram. Addressing: Logical and Physical. Concept of a Protocol. Programs and Processes, Protocol Layering Concepts, Encapsulation and Decapsulation. OSI and TCP/IP model Introduction.

Transmission Medium. Types of transmission media: cable/wired media (coaxial, fibre-optic, twisted pair), wireless media (satellite, microwave, infra-red), wireless network technology: (Bluetooth, Wi-Fi), hotspot, modem, bandwidth. Concept of Internet and Intranet

Server, Client and The Client/Server Model. Mail server and Chat Server, HTTP, File Transfer Protocol (FTP), upload, download, World Wide Web (WWW), web browser, web page, website, blogging, Social Networking sites, URL, e-commerce, e-learning, , pod-casting, bulletin board, VoIP. Information Security Basics.

Reference books:

- "Data Communications and Networking Hardcover" Behrouz A. Forouzan, McGraw Hill Education; 5 edition
- Andrew Tanenbaum, "Computer Networks", Pearson Education 4/e.

Choice based Credit System (CBCS)
Scheme and course structure for

Masters in Information Technology 3rd semester effective from academic session 2023 and onwards

Course Code	Name of the Subject	Paper Category	Hours/Week			Credits
			L	T	P	
IT23301CR	Design and Analysis of Algorithms	Core	3	0	2	4
IT23302CR	Python Programming	Core	3	0	2	4
IT23303CR	Dot Net Technologies	Core	3	0	2	4
IT23304CR	Pervasive Computing	Core	2	0	0	2
Discipline Centric Elective (8 credits)						
IT23305DCE	Discrete Mathematics	Elective (DCE)	3	1	0	4
IT23306DCE	Object Oriented Analysis and Design	Elective (DCE)	3	1	0	4
IT23307DCE	Computer Graphics	Elective (DCE)	3	1	0	4
IT23003OE	Project Management	Elective (OE)	2	0	0	2
Total Credits						**24**

3rd Semester
OBJECTIVE AND LEARNING OUTCOME

M.Sc.IT 3rd Semester

Course Title: Design and Analysis of Algorithms

Course Code: IT23301CR **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The primary objectives of this course in Algorithms and Algorithm Analysis are to provide students with a strong foundation in algorithmic thinking, design, and analysis. It begins with an introduction to algorithms and their analysis, covering topics such as growth functions, asymptotic notations, recurrences, and time and space complexity analysis. The course delves into randomized algorithms, divide and conquer, greedy methods, dynamic programming, backtracking, and branch and bound techniques. Students will also explore lower bound theory, complexity classes (P, NP), NP-hard and NP-complete problems, and the concept of approximate algorithms.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Algorithm Fundamentals: Develop a deep understanding of algorithmic concepts, including algorithm analysis, growth functions, and asymptotic notations. Acquire the ability to design and analyze algorithms for various problems.

2. Algorithmic Techniques: Master key algorithmic paradigms such as divide and conquer, greedy methods, dynamic programming, backtracking, and branch and bound. Learn how to apply these techniques to solve real-world problems, including knapsack problems, shortest paths, and traveling salesman problems.

3. Complexity Theory: Gain knowledge of lower bound theory and complexity classes, including P and NP problems. Understand the concept of NP-hard and NP-complete problems, and appreciate the need for approximate algorithms to address complex computational challenges.

4. Practical Algorithm Design: Apply algorithmic principles to practical problem-solving scenarios. Develop the skills to design efficient algorithms for a wide range of applications, and analyze their time and space complexity.

Course Title: Python Programming

Course Code: IT23302CR **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The primary objectives of this course in Python Programming are to provide students with a comprehensive understanding of Python as a programming language and its applications. It begins with an introduction to Python, including installation, basic syntax, and program execution. The course covers fundamental data types, variable declarations, arithmetic and logical operations, and input-output operations. Students will learn about sequence and control structures, including loops, conditional statements, and control statements. The course also introduces string manipulation, lists, dictionaries, tuples, functions, modules, file input-output, exception handling, regular expressions, database programming, and key data science tools.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Python Fundamentals: Develop a strong foundation in Python programming, including syntax, data types, variable handling, and basic operations. Gain proficiency in writing, executing, and debugging Python programs.

2. Data Structures and Manipulation: Acquire the skills to work with various data structures, such as strings, lists, dictionaries, and tuples. Learn how to manipulate and perform operations on these data structures effectively.

3. Functions and Modules: Understand the concept of functions, parameter passing, and modular programming. Explore Python modules, including math, random, and time. Learn to create, use, and explore modules for enhanced functionality.

4. File Handling and Exception Handling: Learn file input-output operations, file modes, and directory manipulation. Develop proficiency in handling exceptions using keywords like try, catch, except, else, finally, and raise. Explore the use of regular expressions for pattern matching.

5. Database Connectivity and Data Science Tools: Gain hands-on experience in connecting to databases, performing insertion, retrieval, and updating operations using Python. Additionally, get introduced to data science tools like NumPy, Matplotlib, and SciPy, which are essential for data analysis and visualization.

Course Title: Dot Net Technologies

Course Code: IT23303CR **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

The primary objectives of this course in Web Development and .NET Framework are to provide students with a comprehensive understanding of web development technologies and the .NET framework. It begins with an introduction to HTML, covering concepts of hypertext, HTML versions, syntax, and elements. The course includes building HTML documents, inserting various elements like text, images, hyperlinks, backgrounds, and tables. Students learn about HTML tags, table layout, font size, frames, and web forms. Additionally, the course introduces the .NET framework, its architecture, Common Intermediate Language (CIL), Just-In-Time (JIT) compilation, assemblies, and managed code.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. HTML Web Development: Develop proficiency in HTML web development, including the creation of web documents, insertion of text, images, hyperlinks, and backgrounds. Understand the use of HTML tags, table layout, font control, frames, and web forms to build interactive web pages.

2. Understanding .NET Framework: Gain a clear understanding of the .NET framework, its architecture, and key components such as Common Language Runtime (CLR), Common Language Specification (CLS), and Common Intermediate Language (CIL). Explore concepts of managed code, garbage collection, and metadata.

3. Programming with VB.NET and C#: Acquire programming skills using VB.NET and C# languages. Learn about flow control, type conversions, variable types, arrays, structs, string manipulation, functions, debugging, and error handling. Develop object-oriented programming proficiency, including collections, comparisons, conversions, and generics.

4. Windows and Web Application Development: Learn to develop Windows applications using controls such as buttons, labels, textboxes, and menus. Explore Single Document Interface (SDI) and Multiple Document Interface (MDI) applications. Additionally, delve into web application development using ASP.NET, covering site management, styles, authentication, authorization, and web services. Understand data access techniques, including working with streams, XML, databases, and ADO.NET objects like Connection, Command, Data Reader, Data Adapter, and Data Set.

Course Title: Pervasive Computing

Course Code: IT2304CR **Max. Marks 40**
Credits: 02 **Theory External: 40; Min Marks: 16**
Contact Hrs: 20 **Internal (Continuous Assessment): 10 Marks, Min Marks: 04**

Objectives

The objectives of this course on Pervasive Computing and Mobile Technologies are to provide students with a comprehensive understanding of emerging technologies in the mobile and pervasive computing domain. The course covers technologies from the past, present, and future, including pervasive computing, device technology, operating systems, Java for pervasive devices, device connectivity, web application concepts, WAP (Wireless Application Protocol), and voice technology. Students will explore topics such as speech recognition, personal digital assistants (PDAs), and mobile applications.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Pervasive Computing Fundamentals: Gain a solid understanding of pervasive computing, its historical development, and its significance in modern technology. Explore device technology, including hardware, human-machine interfaces, biometrics, and operating systems.

2. Web Application and Wireless Technologies: Learn about the World Wide Web's history, architecture, and protocols. Understand the principles of transcoding and client authentication via the internet. Explore the Wireless Application Protocol (WAP), its components, infrastructure, security issues, and Wireless Markup Language (WML). Study voice technology basics, speech recognition, standards, and applications in the context of pervasive computing.

3. Mobile Devices and Personal Digital Assistants (PDAs): Explore the history and categories of personal digital assistants (PDAs). Understand PDA operating systems, device characteristics, software components, and industry standards. Gain insight into mobile applications and PDA browsers, which play a crucial role in the pervasive computing landscape.

4. Emerging Trends: Stay updated with emerging trends in pervasive computing and mobile technologies. Gain knowledge of the latest advancements and products in this rapidly evolving field, including i-mode and voice technology applications. Understand the security challenges and considerations associated with pervasive computing and mobile devices.

Course Title: Discrete Mathematics

Course Code: IT23305DCE **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

This course on Discrete Mathematics aims to provide students with a strong foundation in mathematical concepts essential for computer science and problem-solving. The course covers topics such as propositional logic, sets, functions, counting principles, recurrence relations, relations, methods of proof, partially ordered sets, lattices, trees, graph theory, and Boolean algebra. Students will learn to apply these mathematical tools to analyze and solve problems in computer science and related fields.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Mathematical Foundations: Develop a solid understanding of fundamental mathematical concepts, including propositional logic, truth tables, predicates, quantifiers, sets, functions, and their properties. Learn how to apply these concepts in computer science contexts.
2. Counting and Combinatorics: Explore counting principles, permutations, combinations, discrete probability, and advanced counting techniques like inclusion-exclusion. Gain proficiency in solving recurrence relations and understand the pigeonhole principle.
3. Graphs and Relations: Master graph theory, including graph terminology, connectivity, Eularian and Hamiltonian paths, graph representations, and coloring. Understand relations, equivalence relations, diagraphs, and their matrix representations. Learn to compute transitive closures using Warshall's algorithm.
4. Advanced Topics: Dive into advanced topics such as partially ordered sets, lattices, trees (including binary search trees and minimal spanning trees), finite Boolean algebra, and groups. Apply these mathematical concepts to solve problems and make informed decisions in computer science applications.

Course Title: Object Oriented Analysis and Design

Course Code: IT23306DCE **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

This course on Object-Oriented Analysis and Design (OOAD) aims to equip students with the fundamental principles and techniques of OOAD using the Unified Modeling Language (UML) and design patterns. The course covers various phases of the Unified Process (UP), including inception, elaboration, and design. Students will learn to apply OOAD concepts to analyze requirements, create use case models, develop domain models, and design software systems using interaction diagrams, GRASP patterns, and CRC cards.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. OOAD Fundamentals: Gain a comprehensive understanding of OOAD principles, UML notation, and iterative development. Learn to apply UML and design patterns effectively in software analysis and design.

2. Requirements Analysis: Learn how to identify and document different types of requirements, including functional and non-functional. Develop use case models, write use cases, and create System Sequence Diagrams (SSDs) to capture system interactions.

3. Domain Modeling: Visualize and construct domain models, including conceptual class identification, associations, attributes, and operations. Apply UML association notation and operation contracts to enhance domain models.

4. Design and Implementation: Progress from requirements to design by creating interaction diagrams (Sequence and Collaboration diagrams). Understand GRASP (General Responsibility Assignment Software Patterns) principles, use CRC (Class-Responsibility-Collaboration) cards, and design class diagrams. Map design models to code for effective implementation.

Course Title: Computer Graphics

Course Code: IT23307DCE **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

This course in Computer Graphics aims to provide students with a foundational understanding of computer graphics principles and techniques. It covers a wide range of topics, including 2-D and 3-D graphics, transformations, clipping, hidden surface removal, curves, and surfaces, as well as fractals. The course prepares students to apply these concepts in various applications and industries where computer graphics plays a crucial role.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Foundational Understanding: Gain a solid grasp of computer graphics fundamentals, including display devices, double buffering, and the use of lookup tables for efficient rendering.

2. 2-D Graphics: Learn techniques for drawing lines, circles, and ellipses using algorithms like Bresenham's and DDA. Explore 2-D transformations, window and viewport concepts, and filling and boundary algorithms.

3. 3-D Graphics: Understand 3-D graphics, including projections (perspective and parallel), transformations, and hidden surface removal techniques such as the Z-buffer algorithm and back face detection.

4. Curves, Surfaces, and Fractals: Explore curve and surface representations through splines, including Bezier and B-Splines. Learn about fractal generation procedures and their application in computer graphics.

Course Title: Project Management

Course Code: IT23003OE **Max. Marks 50**
Credits: 02 **Theory External: 40; Min Marks: 16**
Contact Hrs: 20 **Internal (Continuous Assessment): 10 Marks, Min Marks: 04**

Objectives

This course in Software Project Management aims to equip students with the essential knowledge and skills required to effectively plan, execute, and monitor software projects. It covers various aspects of project management, from project definition and planning to risk management and team dynamics. By the end of this course, students will be prepared to lead and contribute to successful software projects in a professional setting.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Project Management Fundamentals: Develop a comprehensive understanding of software project management, including project definition, contract management, and the stepwise project planning process.

2. Cost Benefit Analysis and Risk Assessment: Learn techniques for evaluating the feasibility of software projects through cost-benefit analysis. Identify, assess, and manage project risks effectively to minimize potential issues.

3. Project Evaluation and Monitoring: Gain insights into risk management, including hazard identification, analysis, and control. Understand how to create a framework for data collection and implement monitoring techniques such as Earned Value to track project progress and make informed decisions.

4. Organizational Behavior and Team Dynamics: Explore the principles of organizational behavior, team formation, motivation, leadership, and decision-making within project teams. Understand how to foster a productive and cohesive project team.

Choice based Credit System (CBCS)
Scheme and course structure for

Masters in Information Technology 3rd semester effective from academic session 2023 and onwards

Course Code	Name of the Subject	Paper Category	Hours/Week			Credits
			L	T	P	
IT23301CR	Design and Analysis of Algorithms	Core	3	0	2	4
IT23302CR	Python Programming	Core	3	0	2	4
IT23303CR	Dot Net Technologies	Core	3	0	2	4
IT23304CR	Pervasive Computing	Core	2	0	0	2
Discipline Centric Elective (8 credits)						
IT23305DCE	Discrete Mathematics	Elective (DCE)	3	1	0	4
IT23306DCE	Object Oriented Analysis and Design	Elective (DCE)	3	1	0	4
IT23307DCE	Computer Graphics	Elective (DCE)	3	1	0	4
IT23003OE	Project Management	Elective (OE)	2	0	0	2
Total Credits						**24**

Detailed Syllabus of

3rd Semester

Course Code: IT23301CR
Course Title: Design and Analysis of Algorithms

UNIT I

Introduction to Algorithms, Analysis of algorithms, Designing Algorithms, Growth of Functions, Asymptotic notations, Recurrences , Substitution method , Iteration method, Recursion trees , The Master Method, Time and Space Complexity study of some basic algorithms. [8L]

Randomized Algorithms: Identifying the repeated element, Primality testing, Advantages and Disadvantages. [4L]

UNIT II

Divide and Conquer, General method, Binary search, Quick sort, Merge sort [6L]

Greedy Method, General method, Knapsack problem, Single source shortest paths, MST and algorithms for MST [6L]

UNIT III

Dynamic programming, General methods, All pair shortest paths, Traveling salesman problems. Backtracking, General method, 8-Queen problem, Sum of subsets, Knapsack problem. Branch and Bound, General method, Least Cost Branch and Bound, Traveling salesperson problem. [12L]

UNIT IV

Lower boundary theory, Lower bound theory through reductions, P and NP problems. NP hard and NP complete problems. Approximate Algorithms and their need, the vertex Cover Problem, The traveling salesman problem, The subset sum problem. [12L]

Text booka:

1. Pearson Horowitz, Sahni, “Fundamentals of Computer Algorithms”, Galgotia Publications
2. Goodrich and Tamassia “Algorithm design”

Reference books:

1. Coremen, Leiserson, Rivest,Stein, “Introduction to Algorithms”, 2nd edition, PHI.
2. Aho, Hopcroft and Ullman, “The Design and Analysis of Computer Algorithms”, Pearson.

Course Code: IT23302CR
Course Title: Python Programming

UNIT I

Introduction to Python Programming: installing python, basic syntax, Writing and executing simple program. Basic Data Types.

Declaring variables, Performing assignments; arithmetic, logical and relational operations, Comments, Simple input-output. Sequence, Control – Precedence of operators, Type conversion

Conditional Statements: if, if-else, nested if –else Looping: for, while, nested loops. Control statements: Terminating loops, skipping specific conditions, use of compound expression in conditional and looping construct.

UNIT II

String Manipulation: declaring strings, escape characters, string functions.

Lists: basics, list operators (joining, slicing, +, *, in , not in); inserting, replacing, and removing an element, searching and sorting list. List functions and methods: len( ), insert( ), append( ), extend( ), sort( ), remove( ), reverse( ), pop( ), list( ), count( ), extend( ), index( ), cmp( ), max( ), min( ).

Dictionaries: Basics, Concept of key-value pair, creating, initializing and accessing the elements in a dictionary, traversing, appending updating and deleting elements. Dictionary Functions and methods: cmp( ), len( ), clear( ), get( ), has_key( ), items( ), key( ), update( ), values( ), pop( ), from keys(), dict().

Tuples: Immutable concept, creating, initializing and accessing elements in a tuple, Tuple assignment, Tuple slices, Tuple indexing; Tuple Functions: cmp(), len(), max(), min(), tuple(), index(), count(), sum(), any(), all(), sorted(), reversed().

UNIT III

Functions: Defining a function, calling a function, Advantages of functions, types of functions, function parameters, Formal parameters, Actual parameters, anonymous functions, global and local variables Modules: Importing module, Creating & exploring modules, Math module, Random module, Time module.

File Input-Output: Opening and closing file, Various types of file modes, reading and writing to files, manipulating directories Exception Handling: What is exception, Various keywords to handle exception such try, catch, except, else, finally, raise–Regular Expressions–Concept of regular expression, various types of regular expressions, using match function.

UNIT IV

Database and Python: Connecting to MySQL, Insertion, Retrieval, Updation. Basic Database programming with Python.

Introduction to Data Sciences and Tools: NumPy, Matplotlib, SciPy

Reference Books

- Python in a Nutshell, 3rd Edition, A Desktop Quick Reference, Alex Martelli, Anna Ravenscroft, Steve Holden,O'Reilly Media, 2017
- Python: A Step by Step for Beginners, Tom Tiffel
- Beginning Programming with Python for Dummies John Mueller, Wiley
- Python Crash Course: A Hands-On, Project-Based Introduction to Programming, Eric Matthes
- Beginning Python: Using Python 2.6 and Python 3, James Payne, Wiley India, 2010.

Course Code: IT23303CR
Course Title: Dot NET Technologies

UNIT I

HTML - Concepts of Hypertext, Versions of HTML, Elements of HTML syntax, Head & Body Sections, Building HTML documents, Inserting texts, Images, Hyperlinks, Backgrounds and Colour controls, Different HTML tags, Table layout and presentation, Use of front size & Attributes List types and its tags, Use of Frames and Forms in web pages
Introduction to .NET Framework, .NET Architecture, CIL and JIT, Assemblies, Managed Code, Garbage Collection, MSIL and Metadata, CLR, CLI, CLS.

UNIT II

VB.NET Concepts: Flow Control, Type Conversions, Complex Variable Types, Arrays, Structs, String Manipulation. Functions, Debugging and Error Handling. Object Oriented Programming using C#, Collections, Comparisons and Conversions. Generics.

UNIT III:

Basic Windows Programming: Controls, Button, Label and Link Label, Text Box, Radio and Checkbox, RichTextBox, List and CheckBoxes, TreeView and ListView Controls, Tab Control. Menus and ToolBars, SDI and MDI Applications.

UNIT IV

ASP.NET Web Programming: Site Management (Client and Server Side), Styles, Master Pages, Site Navigation, Authentication and Authorization, Web Service. **Data Access:** Streams, XML, Connection and Command Objects, Data Reader, Data Adapter, Data Set.

Reference books:

- Professional VB.NET 2010 by Christian Nagel , Bill Evgen , Jay Glynn Wrox Publications , 2006.
- Dietel&Dietel , “VB.NET , How to Program”,Pearson Education.
- Visual Basic.Net by John Sharp & John Jagger, PHI, New Delhi.
- Visual Studio .Net by Francisco, Microsoft Publication.

Course Code: IT23304CR
Course Title: Pervasive Computing

UNIT I

Technologies : Past, Present, Future , Pervasive Computing , The pervasive computing market, Device Technology : Hardware , Human-machine interfaces , Biometrics , Operating Systems , Java for Pervasive devices , Device Connectivity : Protocols , Security , Device Management , Web Application Concepts : History of World wide Web . World Wide Web Architecture, Protocols, Transcoding, Client Authentication via the Internet.
WAP : Introduction , Components of the WAP architecture , WAP infrastructure , WAP Security Issues , Wireless Markup Language , WAP push , Products , i-mode , Voice Technology :

UNIT II

Wireless Markup Language , WAP push , Products , i-mode , Voice Technology : Basics of Speech Recognition , voice standards , speech applications , speech and pervasive computing, security, Personal Digital assistants : History , Device Categories , PDA Operating Systems , Device Characteristics , Software Components , Standards , Mobile Applications , PDA browsers.

Reference books :

- JochenBurkhardt, Dr. Horst Henn , Stefan Hepper , Klaus Rintdorff, Thomas schack " Pervasive Computing " Technology and Architecture of Mobile Internet Applications , Pearson Education.

Course Code: IT23305DCE
Course Title: Discrete Mathematics

UNIT I

Introduction and Foundation: Proposition, Logic, Truth tables, Propositional Equivalence, Logical Equivalence, Predicates and Quantifiers, Sets: operations on sets, Computer representation of sets, Cardinality of a Set. Functions: Domain, Range, One- to-One, Onto, Inverses and Composition, fog, gof etc.

UNIT II

Counting: Basic Counting Principle, The Pigeon-Hole Principle, Permutation, combinations, repetitions, discrete probability, Advanced Counting Techniques: Inclusion-Exclusion, Applications of inclusion-exclusion principle. Recurrence relations, solving recurrence relation. Relations: Relations and their properties, Binary Relations, Equivalence relations, Diagraphs, Matrix representation of relations and digraphs, Computer representation of relations and digraphs, Transitive Closures, Warshall's Algorithm.

UNIT III

Methods of Proof: Different methods of proof, Direct Proof, Indirect Proof, Mathematical Induction for proving algorithms. Partially Ordered Sets (Posets), External elements of partially ordered sets, Hasse diagram of partially ordered set, isomorphic ordered set, Lattices: Properties of Lattices, complemented Lattices. Trees: Rooted trees, Application of trees: Binary Search Trees, Decision Trees, Prefix Codes, Tree traversal, trees and sorting, spanning trees, minimal spanning trees.

UNIT IV

Graph theory: Introduction to graphs, Graph Terminology Weighted graphs, Representing Graphs, Connectivity of Graphs: Paths and Circuits, Eularian and Hamiltonian Paths, Matrix representation of graphs. Graph Coloring. Finite Boolean algebra, Functions on Boolean algebra, Boolean functions as Boolean polynomials. Groups and applications: Subgroups, Semigroups, Monoids.

Text book :

- Kenneth H. Rosen "Discrete Mathematics and Its Applications" The Random House/Birkhauser Mathematics series

Reference books:

- LIU "Elements of Discrete Mathematics " Tata McGraw Hill
- Schaums "Discrete Mathematics " Tata McGraw Hill
- Kolman/Rehman "Discrete Mathematical Structures " Pearson Education
- Nicodemi "Discrete Mathematics " CBS

Course No: IT23306DCE
Course Title: Object Oriented Analysis & Design

UNIT I

OOAD – Introduction , Applying UML and Patterns in OOAD , Assigning Responsibilities , What is analysis and Design , An Example , The UML , Iterative Development –an Unified Process idea , Additional UP Best Practices and Concepts , The UP Phases and Schedule oriented Terms , The UP disciplines. Process Customization and the development case. The Agile UP.The Sequential Waterfall Lifecycle. Inception. Artifacts that may start in inception, Understanding requirements, types of requirements.

Unit II

Use –case Model , Writing requirements in context , goals and stories , background , use cases and adding value , use cases and functional requirements , use case types and formats . Goal and scope of a use case , Finding primary actors , goals and use cases , writing use cases in an essential UI-free style , Actors , Use Case Diagrams , Use Cases within the UP , Case Study. Identifying other requirements. From inception to elaboration.

Use Case Model : Drawing System Sequence Diagrams. Example of an SSD. SSDs and Use Cases , System Events and the System Boundary , Name System Events and Operations , Showing Use Case Text , SSDs within the UP

Unit III

.Domain Model : Visualizing Concepts , Domain Models , Conceptual Class Identification , Candidate Conceptual classes , The UML association notation , NextGen POS Domain Model Associations , NextGen POS Domain Model , Adding Attributes , Non Primitive Data Type Classes , Adding Detail with Operation Contracts , Contract Sections , Post Conditions , Contracts , Operations and the UML. Operation Contracts within the UP.

UNIT IV

From Requirements to Design , Interaction Diagram Notation , Sequence and Collaboration Diagrams , GRASP , Responsibilities and methods , interactions diagrams , Patterns , GRASP : Pattern of General Principles in Assigning Responsibilities , Information Expert , creator , Low Coupling , High Cohesion , Controller , Object Design and CRC Cards , Design Model : Use Case Realization with GRASP Patterns , Determining Visibility , Creating Design Class Diagrams , Mapping Design to Code

Reference Books:

- James Rumbaugh, “Object Oriented Models and Design” Pearson Education 2/e
- Ali Bahrani “Object Oriented Systems Development” McGraw -Hill 1999
- Lafore Robert, “Object Oriented Programming in C++”, Galgotia Publications.
- Balagurusami, E, “Object Oriented with C++”, Tata McGraw-Hill.
- D. Ravichandran, “Programming with C++”, McGraw-Hill Publications

Course Code: IT23307DCE
Course Title: Computer Graphics

UNIT I
Introduction to Computer Graphics. Applications of Computer Graphics. Graphic Display Devices_ Raster, Refresh, Random. Display Buffer, Concept of Double Buffering and Segmentation of Display Buffer. Use of Lookup tables.

UNIT II
2-D Graphics. Cartesian and Homogeneous Coordinate Systems. Line drawing algorithms (Bresenhams and DDA). Circle and Ellipse Drawing Algorithms. 2-Dimensional Transformations, Concepts of Window & Viewport, Window to Viewport Transformations. Filling, Boundary and Floodfill algorithms.

Unit III
Clipping, Line Clipping Algorithms (Cohen-Sutherland Algorithm), 3-D Graphics, Projections: perspective and parallel projection transformations. 3-Dimensional Transformations. Hidden Surface Removal Techniques, Z-Buffer Algorithm, Back Face Detection.

Unit IV:
Curves and Surfaces, Splines, Spline specification, Interpolated & Approximated Splines. Bezier Splines, Bezier Curves, Cubic Bezier Curves, Bezier Surfaces. B-Splines curves and surfaces. Fractals - Fractal Generation Procedure.

Text book:

- Hearn and Baker “Computer Graphics” 2nd Edition , Pearson Education.

Reference books:

- W.M.Newman and Sproull. “Principles of interactive Computer Graphics” ,TMH 2.Steven Harrington.” Computer Graphics a Programming Approach” McGraw Hill.
- Plastock and Kelley. “Schaums outline of theory and problems of computer Graphics”
- David F Frogers and J Alan Adams. “Procedural Elements of Computer Graphics” McGraw Hill
- David F Rogers and J Alan Adams. “Mathematical Elements of Computer Graphics” McGraw Hill
- James. D. Foley, A Vandam etal “Computer Graphics” Pearson.

Course Code: IT23003OE
Course Title: Project Management

Unit I - Introduction to Software Project Management

Project Definition – Contract Management – Activities Covered By Software Project Management – Overview of Project Planning – Stepwise Project Planning. Strategic Assessment – Technical Assessment – Cost Benefit Analysis. Cash Flow, Forecasting.

Cost Benefit Evaluation Techniques – Risk Evaluation. Objectives – Project Schedule – Sequencing and Scheduling Activities –Network Planning Models – Forward Pass – Backward Pass – Activity Float – Shortening Project Duration – Activity on Arrow Networks.

Unit II - Project Evaluation and Monitoring

Risk Management – Nature Of Risk – Types Of Risk – Managing Risk – Hazard Identification – Hazard Analysis – Risk Planning And Control. Creating Framework – Collecting The Data – Visualizing Progress – Cost Monitoring – Earned Value – Prioritizing Monitoring – Getting Project Back To Target – Change Control.

Understanding Behavior Organizational Behavior: A Background –Selecting The Right Person For The Job – Instruction In The Best Methods – Motivation– The Old man – Hackman Job Characteristics Model – Working In Groups – Becoming A Team – Decision Making – Leadership – Organizational Structures.

Text book:

- Bob Hughes, Mikecotterell, “Software Project Management”, Third Edition, Tata McGraw Hill, 2004.

Reference books:

- Ramesh, Gopalaswamy, "Managing Global Projects", Tata McGraw Hill, 2001.
- Walker Royce, “Software Project Management - A Unified Framework”, Pearson Education,2004.
- Jalote, “Software Project Management in Practice”, Pearson Education, 2002. 2. Humphrey Watts, “Managing the Software Process”, Addison Wesley, 1989.

Choice based Credit System (CBCS)
Scheme and course structure for

Masters in Information Technology 4th semester effective from academic session 2023 and onwards

Course Code	Name of the Subject	Paper Category	Hours/Week			Credits
			L	T	P	
IT23401CR	Project	Core	0	0	16	8
IT23402CR	Software project/Demo/Dissertation	Core	0	4	0	4
IT23403CR	Internet of Things (IoT)	Core	2	0	0	2
Discipline Centric Elective (8 credits)						
IT23405DCE	Data Warehouse	Elective (DCE)	3	1	0	4
IT23406DCE	Finite Automata and Formal Languages	Elective (DCE)	3	1	0	4
IT23407DCE	Machine Learning	Elective (DCE)	3	1	0	4
IT2300O4GE	Management Information Systems (MIS)	Elective (GE)	2	0	0	2
Total Credits						**24**

4th Semester
OBJECTIVE AND LEARNING OUTCOME

M.Sc.IT 4th Semester

Course Title: Internet of Things

Course Code: IT23403CR **Max. Marks 40**
Credits: 02 **Theory External: 40; Min Marks: 16**
Contact Hrs: 20 **Internal (Continuous Assessment): 10 Marks, Min Marks: 04**

Objectives

This course on the Internet of Things (IoT) provides an in-depth understanding of IoT concepts, technologies, and challenges. It covers the physical and logical design of IoT systems, communication models, and APIs. Additionally, it explores the historical evolution of IoT, adoption trends, and the associated risks and security concerns. By the end of this course, students will be equipped to design, develop, and address the challenges of IoT solutions.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. IoT Fundamentals: Gain a solid foundation in IoT, including its characteristics, physical and logical design, and functional blocks. Understand different communication models and APIs used in IoT systems.

2. Historical Context and Trends: Explore the history and evolution of IoT and analyze current trends in IoT adoption. Identify the risks, privacy concerns, and security issues associated with IoT implementations.

3. Network and Communication Aspects: Delve into wireless medium access challenges, MAC protocols, routing protocols, sensor deployment, and data aggregation in IoT networks. Understand the complexities of IoT communication.

4. Challenges and Tools: Recognize the design, development, security, privacy, and risk challenges in IoT projects. Gain an introduction to various IoT tools that facilitate IoT solution development.

Course Title: Data Warehouse

Course Code: IT23405DCE **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives
This course on Data Warehousing provides comprehensive knowledge and skills for designing, planning, and implementing data warehousing solutions. It covers the fundamental concepts of data warehousing, architectural principles, data extraction, transformation, and loading (ETL), data quality, information delivery, and data mining. By the end of this course, students will be prepared to design and deploy data warehousing solutions effectively.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Data Warehousing Foundations: Acquire a deep understanding of data warehousing, its features, components, and data mart concepts. Explore metadata's role and importance in data warehousing and learn how to plan a data warehousing project, including business requirements and data design.

2. Architecture and Infrastructure: Grasp the key concepts of data warehouse architecture, infrastructure components, including hardware, software, and metadata. Learn principles of dimensional modeling, STAR and Snowflake schemas, and utilize CASE tools.

3. Data Extraction, Transformation, and Loading (ETL): Master the ETL process, data quality challenges, and the use of data quality tools. Understand information access and delivery, along with OLAP concepts, models, and platforms.

4. Data Mining and Physical Design: Explore data mining definitions, techniques, and applications. Learn about physical design considerations, RAID technology, storage size estimation, indexing methods, and performance enhancement techniques like data partitioning and clustering. Understand data warehouse deployment strategies.

Course Title: Finite Automata and Formal Languages

Course Code: IT23406DCE **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

This course in Automata Theory and Formal Languages aims to provide students with a strong foundation in the theory of computation. It covers various types of automata, regular expressions, context-free grammars, pushdown automata, Turing machines, and their relationships. Students will learn about decidability, time and space complexity, and the concept of NP-completeness. By the end of the course, students should be proficient in analyzing the computational complexity of problems.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Finite Automata: Gain a deep understanding of finite automata, including deterministic and nondeterministic types, as well as those with ε-transitions. Learn how to apply finite automata in practical scenarios.

2. Regular Languages and Grammars: Master regular languages, regular expressions, Arden's Method, and the Pumping Lemma. Understand how to minimize finite automata and work with context-free grammar.

3. Pushdown Automata and Context-Free Languages: Explore pushdown automata, their languages, and their equivalence with context-free grammar. Learn about deterministic pushdown automata and Chomsky hierarchies of grammar.

4. Decidability and Complexity: Delve into decidability concepts, the halting problem, resource-bounded computation, time and space complexity, NP-completeness, and reductions. Understand the relationships between different classes of languages and grammar.

Course Title: Machine Leaning

Course Code: IT23407DCE **Max. Marks 100**
Credits: 04 **Theory External: 80; Min Marks: 32**
Contact Hrs: 40 **Internal (Continuous Assessment): 20 Marks, Min Marks: 08**

Objectives

This course in Machine Learning aims to introduce students to the fundamental concepts and techniques of machine learning, including various types of learning and algorithms. It covers classification, regression, clustering, reinforcement learning, and artificial neural networks. Students will gain a comprehensive understanding of the principles and applications of machine learning.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Machine Learning Basics: Understand the foundational concepts of machine learning, its historical development, and the machine learning process. Recognize the types of problems suited for machine learning and the key aspects of training data.

2. Supervised and Unsupervised Learning: Differentiate between supervised, unsupervised, and reinforcement learning. Explore classification and regression algorithms, including Support Vector Machines and Decision Trees. Learn about clustering algorithms such as k-Nearest Neighbors and K-means.

3. Reinforcement Learning: Delve into reinforcement learning, Markov Decision Processes, and the distinctions between reinforcement and supervised learning. Identify applications of reinforcement learning in various domains.

4. Artificial Neural Networks (ANNs): Gain insight into artificial neural networks, their biological inspiration, neuron models, and the backpropagation algorithm. Explore different types of ANNs, including single-layer and multilayer perceptron.

Course Title: Management Information System

Course Code: IT23004GE **Max. Marks 50**
Credits: 02 **Theory External: 40; Min Marks: 16**
Contact Hrs: 20 **Internal (Continuous Assessment): 10 Marks, Min Marks: 04**

Objectives

This course in Information Systems Management and Analysis aims to provide students with a comprehensive understanding of how information systems are organized, their impact on business operations, and the stages involved in system analysis and development. Students will also learn about various system development models and the roles of key professionals in the field.

Learning Outcomes

After successful completion of the course, the students should be able to:

1. Information Systems and Business Impact: Understand the relationship between information systems, business operations, and the changing business environment. Explore the influence of IT/IS on organizational structure, managers, and decision-making processes.

2. Types of Information Systems: Categorize information systems based on their nature and characteristics, including Transaction Processing Systems, Office Automation Systems, Management Information Systems, Decision Support Systems, Expert Systems, and Executive Support Systems. Recognize their roles in facilitating different levels of decision-making.

3. System Analysis and Development: Comprehend the need for system analysis and the structured stages involved in this process. Gain familiarity with tools such as Data Flow Diagrams, Context Diagrams, Decision Tables, and Structured Diagrams.

4. Specialized Information Systems: Explore how information systems are tailored to specific functions within organizations, including Accounting, Finance, Production and Manufacturing, Marketing, HRM, and various industry sectors such as healthcare, hospitality, and banking. Learn about Enterprise Resource Planning (ERP), Supply Chain Management (SCM), and Customer Relationship Management (CRM) systems, their features, selection criteria, and implementation challenges.

Detailed Syllabus of

4th Semester

Course Title: Project

Course Code: IT23401CR (8 Credits)

Course Title: Software projects/Demonstration/Dissertation

Course Code: IT23402CR (4 Credits)

Course Code: IT23403CR
Course Title: Internet of Things (IOT)

UNIT I

Introduction to IoT, Defining IoT, Characteristics of IoT, Physical design of IoT, Logical design of IoT, Functional blocks of IoT, Communication models & APIs

Brief History and evolution of IoT, Trends in the Adoption of IoT, Risks, Privacy, and Security

IoT & M2M Machine to Machine, Difference between IoT and M2M, Software define Network

UNIT II

Network & Communication aspects - Wireless medium access issues, MAC protocol survey, Survey routing protocols, Sensor deployment & Node discovery, Data aggregation & dissemination.

Challenges in IoT- Design challenges, Development challenges, Security challenges, Risks and Privacy challenges, Introduction to different IoT tools.

Reference Books:

- Vijay Madisetti, Arshdeep Bahga, “Internet of Things: A Hands-On Approach”
- Waltenegus Dargie, Christian Poellabauer, "Fundamentals of Wireless Sensor Networks: Theory and Practice"
- The Internet of Things: Enabling Technologies, Platforms, and Use Cases", by Pethuru Raj and Anupama C. Raman (CRC Press).

Course Code: IT23405DCE
Course Title: Data Warehousing

UNIT I

Data warehouse: Definitions, features, building blocks/ components, data marts, Meta data in data warehouse; planning a data warehouse, The project team, project management considerations, Business requirements; data design, the architectural plan, Data storage specifications, Information delivery strategy.

Unit II

Architecture and Infrastructure: Concept of data warehouse architecture, operational infrastructure, physical infrastructure, hardware and operating systems, database software, tools. The role of metadata, metadata types, metadata requirements. Principles of dimensional modelling: Dimensional modelling basics, Use of CASE tools, The STAR schema, The Snowflake Schema.

Unit III

Data Extraction, Data Transformation, Data Loading. Data Quality: Need, Data Quality Challenges, Data Quality Tools. Information access and delivery, Information delivery tools. Online Analytical Processing (OLAP): Features, functions, OLAP models, Implementation considerations, OLAP platforms, OLAP tools and products.

Unit IV

Introduction to Data Mining: definitions, Data mining techniques, applications. Physical Design in data warehouse: Steps, Physical Design considerations, Physical storage. RAID technology, estimating storage sizes, Indexing the data warehouse: B-Tree Index, Bitmapped Index, Clustered Index Performance Enhancement Techniques: Data Partitioning, Data Clustering, Parallel processing, data arrays. Data warehouse deployment.

Text book:

- Paulraj Pooniah , " Data Warehousing Fundamentals " Wiley

Reference books:

- Alex Berson , Stephen J. Smith " Data Warehousing , Data Mining and OLAP , Tata McGraw Hill , 2004 Tenth reprint 2007.
- Sam Anahory , Dennis Murray ," Data Warehousing in the real world " , Pearson Education.

Course Code: IT23406DCE
Course Title: Finite Automata & Formal Languages

UNIT I

Introduction to finite Automata: Basic Terminology: Alphabet, Introduction to finite automata, Deterministic finite automata, Nondeterministic finite Automata, Equivalence of DFA and NDFA, Non-Deterministic Finite automata with ε-Transitions. Application of finite Automata [12L]

UNIT II

Regular Languages and Regular Expressions, Arden's Method. [4L]

The Pumping Lemma for Regular sets, Application of Pumping Lemma, Closure Properties of Regular Sets, Minimization of Finite Automata [4L]

Context free grammars, Parse trees, Ambiguity in Grammars, Left Recursion, Eliminating Epsilon Productions, Eliminating Unit productions, Chomsky hierarchies of grammars [4L]

UNIT III

Definition of the pushdown automata, the languages of a PDA, Equivalence of PDA's and CFG's, Deterministic Pushdown Automata [6L]

The Turing Machine, Programming techniques for Turing machines. Recursively Enumerable and Recursive Languages, Unrestricted Grammars, Context Sensitive Language and Grammar. Relation between Languages of classes [6L]

UNIT IV

Decidable Languages: Decidability, Relationship between Solvability and Decidability, halting problem [4L]

Introduction to resource-bounded computation, Time and Space, A Time Hierarchy, Nondeterministic Polynomial Time, some NP-Complete Problems Reductions and its applications [8L]

Reference books:

- Hopcroft, J., and Ullman, J. (1979), "Introduction to Automata Theory, Languages and Computation", Pearson Education.
- P. Linz, "Introduction to Formal Languages and Automata", 3rd edition, 2000, Jones and Barlett, PWS Publishing Company.
- K.L.P. Mishra: Theory of Computer Science, Automata, Languages and Computation, 3rd edition, PHI, 2007
- Eiton Gurarri : Introduction to Theory of computation, Computer Science press

Course Code: IT23407DCE
Course Title: Machine Learning

UNIT I

Introduction -What is Machine Learning – how do machines learn, History of machine learning, Machine learning Process, Types of Machine Learning, Why machine learning. Types of problems in machine learning, Aspects of inputs to training Learning systems.
Supervised learning, unsupervised learning, Reinforcement learning. Applications of Machine Learning.

UNIT II

Introduction to Classification, Classification Model, Classification Learning Steps, and Classification Algorithms: Regression, Linear Regression Algorithms Support Vector Machine (SVM), Decision tree learning algorithms: Introduction, Inference model, general domains, symbolic decision trees, consistency, learning trees from training examples, entropy, Information gain , mutual information, ID3 algorithm.

UNIT III

Clustering Algorithms: k-Nearest Neighbour (kNN) and effect of various distance algorithms (Euclidean, Manhattan, Mahalanobis Distances, etc.), K-mean clustering, fuzzy K-means, hierarchical clustering
Reinforcement Learning, Types of Reinforcement learning, Markov Decision Process, Difference between Reinforcement Learning and Supervised Learning, Reinforcement Learning Applications.

UNIT IV

Introduction to Artificial Neural Networks: ANN and applications, Biological Neuron and electrical Model, Usefulness and capabilities, McCulloch and Pitts neural network model, Gradient Decent Algorithm, Types of Networks, Single Layer Perceptron, Multilayer Perceptron, back Propagation Algorithm.

References books:

- Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press.
- Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer.
- Christopher Bishop, Pattern Recognition and Machine Learning, Springer.
- Tom.M.Mitchell, Machine Learning, McGraw Hill International Edition
- Ethern Alpaydin, Introduction to Machine Learning. Eastern Economy Edition, Prentice Hall of India.

Course Code: IT23004GE
Course Title: Management Information systems

UNIT I

Organization and Information Systems , 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. , 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).

Need for System Analysis - Stages in System Analysis - Structured SAD and tools like DFD, Context Diagram Decision Table and Structured Diagram

UNIT II

System Development Models: Water Flow, Prototype, Spiral, RAD – Roles and responsibilities of System Analyst, Database Administrator and Database Designer. Information systems for Accounting, Finance, Production and Manufacturing, Marketing and HRM functions - IS in hospital, hotel, bank.

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): Phases

Recommended books:

- "Management Information Systems", Kenneth J Laudon, Jane P. Laudon, Pearson/PHI,10/e, 2007
- "Management Information Systems", W. S. Jawadekar, Tata McGraw Hill Edition, 3/e, 2004
- Turban, Efraim, Ephraim McLean, and James Wetherbe. 2007. Information Technology for Management: Transforming Organizations in the Digital Economy. New York, John Wiley & Sons