

Evaluation Scheme for ODL-MCA 2020

MASTER OF COMPUTER APPLICATIONS

(100 CREDITS)

Counseling Schedule

I SEMESTER					
S.No.	Course Code	Course Title	Credits	No. of Theory sessions (2 hrs each)	No. of Practical sessions (3 hrs each)
1	OMC 101	Computer Fundamentals and Programming Methodology Using C	3	5	2
2	OMC 102	Computer Organization and Architecture	3	5	
3	OMC 103	Discrete Mathematics	3	5	
4	OMC 104	Management Concepts and Accounting	2	3	
5	OMC 105	Unix and Shell Programming	3	5	2
6	OMC 121	C Programming and Computer Organization Lab	2		15
7	OMC 122	Shell Programming Lab	2		15
TOTAL			18		

II SEMESTER					
S.No.	Course Code	Course Title	Credits	No. of Theory sessions (2 hrs each)	No. of Practical sessions (3 hrs each)
1	OMC 201	Data and File Structure Using C	3	5	1
2	OMC 202	Object Oriented Analysis and Design Using C++	3	5	1
3	OMC203	Data Communication and Computer Network	3	5	
4	OMC 204	Data Base Management System	3	5	2
5	OMC205	Operating System	2	2	
6	OMC 221	Data Structure and C++ Programming Lab	2		15
7	OMC 222	DBMS Lab	2		15
TOTAL			18		

III SEMESTER					
S.No.	Course Code	Course Title	Credits	No. of Theory sessions (2 hrs each)	No. of Practical sessions (3 hrs each)
1	OMC 301	JAVA Programming	3	5	1
2	OMC 302	Software Engineering	2	3	
3	OMC 303	Internet Technologies and Applications	3	5	1
4	OMC 304	C Sharp (C#) Programming	3	5	1
5	OMC 305	ELECTIVE –I	3	5	1
6	OMC 321	JAVA Programming and ITA Lab	2		15
7	OMC 322	C#(.Net) and Computer Graphics Lab	2		15
TOTAL			18		

IV SEMESTER					
S.No.	Course Code	Course Title	Credits	No. of Theory sessions (2 hrs each)	No. of Practical sessions (3 hrs each)
1	OMC 401	Data Warehouse and Data Mining	3	5	
2	OMC 402	Design and Analysis of Algorithm	3	5	2
3	OMC 403	ELECTIVE –II	3	5	
4	OMC 421	Design and Analysis Lab	2		15
5	OMC 470	Project Work	7		15
TOTAL			18		

ELECTIVE –I

OMC305(1)	Computer Graphics
OMC305(2)	Cloud Computing
OMC305(3)	System Software
OMC305(4)	Artificial Intelligence and Neural Network

Elective – II

OMC403(1)	Mobile Application Design and Development
TMC403(2)	Cryptography and Network Security
TMC403(3)	Internet of Things
TMC404(4)	Distributed Systems

Semesters and the Number of Sessions:

Semester	No. of Sessions			No. of Hours		
	Theory	Practical	Specific Assignment	Theory	Practical	Specific Assignment
I	23	30	3	46	90	9
II	22	30	3	44	90	9
III	22	30	3	44	90	9
IV	15	30	3	30	90	9
TOTAL	82	120	12	164	360	36

Total Computer time = Practical time + Specific Assignment related time
= 120 Sessions + 12 Sessions
= 360 hours + 36 hours
= 396 hours

First Semester

OMC 101: Computer Fundamentals and Programming Methodology Using C Credit: 3

Learning Outcomes:

1. Understanding the fundamental concept of IT and working computers to learn about components and internal working of computers.
2. Understand various types of memory and their performances
3. Design algorithms and draw flowcharts to solve computational problems.
4. Make use of various datatypes to propose optimal solution.
5. Divide complex problem into sub problems and develop modular application.
6. Design projects to solve real world problems which helps in implementation of real world problems.
7. Recognize the need of the problem by applying Engineering knowledge and contribute as an individual and team through implementation and application of existing techniques.

Unit No.	Contents	No. of Chapters per unit
1	<p>Introduction: Computer and its characteristics, Definition of Hardware, Input Units, Output Units, Storage Devices: Primary & Auxiliary Memory (Floppy Disk, Hard Disk, Compact Disk, DVD, Flash Disk etc), Others: Network Card, Modem, Sound Card etc.</p> <p>Basic Programming Concepts: Problem solving techniques, Algorithms (Definition and features, criteria to be followed by an algorithm, top down design), Analysis of Algorithm Complexity (Computational Complexity, The Order of Notation, Rules for using the Big - O Notation, Worst and Average Case Behavior), Flowcharts (Basic Symbols used in Flowchart Design). Classification and characteristics of programming languages. Programming paradigms, Procedure Oriented, and Object Oriented Programming Languages.</p>	3
2	<p>Introduction to C Language: Definition and role of computer programs, brief history and salient features of C language, the C compiler, structure of a C program, compiling a C program, run the C program through the menu, run from an executable file. Programming errors (logical and runtime errors), syntax and semantic errors, linker errors. Comment styles in 'C'.</p> <p>Variables and constants: character set, identifiers and keywords, rules for forming identifiers and keywords. Data types and storage, data type qualifiers. Variables, declaring and initializing variables. Constants, integer constants, floating point constants, character constants, string constants, symbolic constants.</p> <p>Expressions and Operators: Assignment Statements, Arithmetic Operators, Relational Operators, Logical Operators, Comma and Conditional Operators, Type Cast Operator, Size of Operator, Bitwise operators, C Shorthand, Associativity and precedence of Operators. Formatted and unformatted input-output.</p> <p>Controlling flow to execution: decision control statements, <i>if</i> and <i>switch</i> statements. Loop control statements, the <i>while</i>, <i>do-while</i>, and <i>for</i> loop statements. Nested loops. The <i>goto</i> statement, the <i>break</i> statement,</p>	4

	the <i>continue</i> statement.	
3	<p>Arrays: Array Declaration, Size Specification, Array Initialization, Initialization of Array Elements in the Declaration, Character Array Initialization, Subscript, Processing the Arrays. Multi-Dimensional Arrays, Multi-Dimensional Array Declaration, Initialization of Two-Dimensional Arrays. Strings: Declaration and Initialization of Strings, Display of Strings Using Different Formatting Techniques, Array of Strings, Built-in String Functions and Applications. Functions: Definition of a Function, Declaration of a Function, Function Prototypes, The Return Statement, Types of Variables and Storage Classes (Automatic Variables, External Variables, Static Variables, Register Variables), Types of Function Invoking, Call by Value, Call by address. Passing array to the function, passing string to the function. Recursion . Structures and Unions: Declaration of Structures, Accessing the Members of a Structure, Initializing Structures, Structures as Function Arguments, Structures and Arrays, Unions, Initializing a Union, Accessing the Members of a Union.</p>	4
4	<p>Pointers: Pointers and their Characteristics, Address and Indirection Operators, Pointer Type Declaration and Assignment, Pointer to a Pointer, Null Pointer Assignment, Pointer Arithmetic, Passing Pointers to Functions, A Function Returning More than One Value, Function Returning a Pointer, Arrays and Pointers, Array of Pointers, Pointers and Strings. Dynamic memory management, <i>malloc()</i>, <i>calloc ()</i>, <i>realloc ()</i>, <i>free ()</i> functions.</p> <p>Files: File Handling in C Using File Pointers, Open a file using the function <i>fopen ()</i>, Close a file using the function <i>fclose()</i>, Input and Output using file pointers, Character Input and Output in Files, String Input / Output Functions, Formatted Input / Output Functions, Block Input / Output Functions , Sequential Vs Random Access Files, Positioning the File Pointer.</p> <p>The C Preprocessor: #define to Implement Constants, # define to Create Functional Macros, Reading from Other Files using # include, Conditional Selection of Code using #ifdef, Using #ifdef for different computer types, Using #ifdef to temporarily remove program statements, Other Preprocessor Commands, Predefined Names Defined by Preprocessor, Macros vs Functions.</p>	4

References:

Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICAT ION
1.	Norton, Peter, "Introduction to Computers", McGraw-Hill.	2009
2.	Leon, Alexis & Leon, Mathews, "Introduction to Computers", Leon Tech World	2010
1	E.Balagurusamy, "Programming in ANSI C", 8 th Edition, Tata McGraw Hill.	2019
2	YashwantKanetkar, "Let Us C", 15 th Edition, BPB Publication.	2018
3	S.K. Srivastava, "C in Depth", 2 nd Edition, BPB Publication.	2012
4	B. W. Kernighan and D. M. Ritchie, "ANSI C: The C Programming Language", 2 nd Edition, Pearson Publication.	2015

Learning Outcomes:

1. Understand the basic components of a computer and milestones in their historical development.
2. Discuss the operation of the arithmetic unit including the algorithms & implementation of fixed-point and floating-point addition, subtraction, multiplication & division.
3. Have a clear understanding of the elements of CPU working and Instruction Set Architecture
4. Identify the impact of the hierarchical memory system including cache memories and virtual on the overall computer system design
5. Evaluate the various aspects I/O operations and their impact on the overall performance and functioning of computers
6. Review the current trends in development of processor architectures with emphasis on instruction level parallelism, latency operations in pipeline design, fault tolerance etc

Unit No.	Contents	No. of Chapters per unit
1	Introduction: Representation of Information and Basic Building Blocks: Introduction to Computer, Computer hardware generation, Number Systems, Character Codes (BCD, ASCII, EBCDIC), Logic gates, Boolean Algebra, K-map simplification, Combination circuits, Adders, Decoder, Encoders, Multiplexer, De-multiplexer, Sequential circuits, Flip-Flops, Registers, Counters (synchronous & asynchronous), IEEE standard for Floating point numbers. Division algorithm and Booth's multiplication algorithm.	4
2	Basic Organization: Processor and its Components, ALU, CU and Special Purpose Registers, Von Neumann Machine (IAS Computer), Register Transfer Language, Bus and Memory Transfers, Common Bus System, Instruction Cycle Operational flow chart (Fetch, Decode), Organization of Central Processing Unit, Hardwired & micro programmed control unit, Single Organization, General Register Organization, Stack Organization, Addressing modes, Instruction formats, I/O Organization, Bus Architecture, Programming Registers	4
3	I/O Organization: Peripheral devices, I/O interface, Modes of Transfer, Priority Interrupt, Direct Memory Access (DMA), Input-Output Processor, Serial Communication. I/O Controllers, Asynchronous data transfer, Strobe Control, Handshaking. Memory Organization: Memory Hierarchy, Main Memory (RAM/ROM chips), Auxiliary Memory, Magnetic Disk and its Performance, Magnetic Tape etc, Associative Memory, Cache Memory, Hit/Miss Ratio, Virtual Memory, Memory Management Hardware.	4
4	Parallel Processing Concepts: Introduction and Advantages, Pipeline & Vector Processing, Arithmetic Pipeline, Instruction Pipeline, RISC pipeline, Vector Processors. Multiprocessors: Characteristics, Interconnection Structures, Interprocessor Arbitration, Interprocessor communication and Synchronization, Cache Coherence.	4

References:

1. Mano moris, "computer system architecture", PHI.
2. William Stalling , "Computer organization & Architecture", Pearson education Asia.
3. Zaky&hamachar, " Computer organization". McGraw Hill.
4. B.Ram, "Computer fundamental Architecture & organization", New age

OMC 103: Discrete Mathematics**Credit: 3****Learning Outcomes:** On successful completion of the course students will be able to:

- Understand the theory and techniques of logic, graphs and trees, and algebraic systems
- Apply the knowledge and skills obtained to investigate the problems
- solve a variety of discrete mathematical problems.
- Communicate mathematical ideas.
- Make effective use of appropriate technology

Unit No.	CONTENT	No. of chapters per unit
1	Matrices: Notation and Definition, Types of Matrices, Algebra of Matrices, Transpose of a Matrix, Solution of linear Equations by Matrix method, Rank of matrix, Eigen values and Eigen vectors, Cayley Hamilton theorem.	4
2	Boolean algebra: Basic operations, Boolean functions, Boolean expression, De-Morgan's theorem, Logic gates, SOP and POS forms, Normal forms, Simplification of Boolean expression, Logic and switching networks, Karnaugh map method for simplification of Boolean expression	4
3	Graph theory: Definition and application of graphs, Konigsberg bridge problem, Simple graph, multi graph and pseudo graph, directed and undirected graphs, degree of a vertex, handshaking theorem, Types of graphs, sub graphs and isomorphic graphs, bipartite graphs, operations of graphs, representation of graphs.	4
4	Paths, Cycles ,cut vertex, cut set and bridge, Connectedness in directed and undirected graphs, Connectivity, Eulerian graph, Hamiltonian graph, Dijkstra's algorithm for shortest path, planar graphs, Euler's formula, Graph coloring, Wetch Powell algorithm, Chromatic polynomial, Decomposition theorem. TREES: Trees and their 3 properties, Rooted tree, Spanning tree, minimal spanning tree, fundamental circuits, rank and nullity, Kruskal's algorithm, Binary tree.	4

References

Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICAT ION
1	J. K. Sharma, "Discrete Mathematics", Macmillan	2006
2	Liptschutz, Seymour, "Discrete Mathematics", TMH	2007
3	Trembley, J.P & R. Manohar, "Discrete Mathematical Structure with Application to Computer Science", TMH	2004
4	Kenneth H. Rosen, " Discrete Mathematics and its applications", TMH	2002
5	Doerr Alan & Lvasseur Kenneth, "Applied Discrete Structures for Computer Science", Galgotia Pub. Pvt. Ltd	2001

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

- Prepare financial statements in accordance with Generally Accepted Accounting Principles.
- Employ critical thinking skills to analyze financial data as well as the effects of differing financial accounting methods on the financial statements.
- Effectively define the needs of the various users of accounting data and demonstrate the ability to communicate such data effectively, as well as the ability to provide knowledgeable recommendations.
- Recognize circumstances providing for increased exposure to fraud
- to define preventative internal control measures.

Unit No.	CONTENT	No. of chapters per unit
1	<p>Nature of Management: Meaning, Definition, it's nature purpose, importance & Functions, Management as Art, Science & Profession- Management as social System Concepts of management-Administration-Organization</p> <p>Evolution of Management Thought: Contribution of F.W.Taylor, Henri Fayol ,Elton Mayo, Chester Barhard& Peter Drucker to the management thought.</p>	4
2	<p>Functions of Management:</p> <p>Planning - Meaning - Need & Importance, types levels – advantages & limitations. Forecasting - Need & Techniques. Decision making steps and Process, Organizing - Elements of organizing & processes: Types of organizations, Delegation of authority - Need, difficulties in delegation – Decentralization</p>	4
3	<p>Functions of Management:</p> <p>Staffing - Meaning & Importance, Direction-Nature–Principles Communication-Types & Importance, Motivation - Importance – theories Leadership - Meaning - styles, qualities & functions of leaders</p> <p>Controlling - Need, Nature, importance, Process & Techniques Coordination - Need – Importance</p>	4
4	<p>Strategic Management:</p> <p>Definition of strategy and strategic management, various levels of strategies – corporate, business and functional level, Relevance of Strategic Management and its Benefits, Strategic Management in India.</p> <p>Recent Trends in Management:</p> <p>Social Responsibility of Management – Management of Crisis, Total Quality Management, Stress Management, Management of Change.</p>	4

References

Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICA TION
1	S.L.Maheshwari, S.K.Maheshwari," Financial Accounting", Vikas Publication	2011
2	JawaharLal, "Financial Accounting" Vikas Publication.	2012
3	P.C.Tulsian,"Financial Accounting", Pearson Education	2010
4	Khan &Jain,"Management Accounting" , TMH	2011
5	K.G Gupta, Management Accounting, Kalyani Publication	2012

Learning Outcomes:

1. Show the importance of Unix.
2. Show how the Files and directories work in UNIX.
3. Explain the concepts of Shell Programming..
4. Explain the concepts of threads.
5. Understand how to implement Unix File System.
6. Understand how to implement UNIX System Tools and Administration.

Unit No.	Contents	No. of Chapters per unit
1	<p>Introduction: Introduction to Unix, Unix system organization (the kernel and the shell), Files and directories, Library functions and system calls. General Purpose Utilities: mkdir, rmdir, rm, cat, touch, chmod, chgrp, cp, mv, links(hardlink/symlink), relative/absolute path, ls, vi, more, head, grep, file, wc, cmp, comm., diff, lp, lpr, sort, find, banner, cal, date, who, tty, etc.</p> <p>Unix Shell programming : Types & functions of Shells, Shell Metacharacters, Shell variables, Shell scripts, the environment, Integer arithmetic and string Manipulation, command line arguments and positional parameters, Decision making and Loop control, controlling terminal input, trapping signals, arrays.</p>	3
2	<p>Process, Process structure, starting new process, waiting for a process, zombie process and process related system calls (fork, exit, wait and exec) .Threads : Creation, Threads synchronization, Threads VS Process. Signals. Interprocess Communication (Message queue, semaphores and shared memory) , sockets and pipes, streams.</p>	3
3	<p>Unix File System : File System, File attributes, file structure, file types (regular , directory and device files) boot block, inode block, super block , mounting and unmounting file system, System booting, run level, shutting down, use of open , create, read, write , close , lseek ,stat , fstat, umask, dup system calls , file maintenance system calls (chmod, chown, unlink, link, symlink, mkdir, rmdir, chdir).</p>	3
4	<p>UNIX System Tools and Administration System development tools: lint, make, SCCS (source code control system), Language development tools: YACC, LEX, handling user accounts, Backup, recovery, security, storage of files, configuring network printer, disk related commands.</p>	3

References

1. Parata, “Advanced Unix programming guide”, BPB
2. Yashwant Kanitkar, “Unix Shell Programming”, BPB
3. Meeta Gandhi, Tilak Shetty, Rajiv Shah, “The ‘C’ Odyssey Unix – the open boundless C”, BPB
4. Sumitabh Das, “Unix Concepts and applications”, TMH
5. Mike Joy, Stephen Jarvis, Michael Luck, “Introducing Unix and Linux”, Palgrave Macmillan.
6. Rachel Morgan, Henry McGilton, “Introducing Unix System V”, TMH

Second Semester

OMC 201 Data and File Structure Using C

Credit: 3

Learning Outcomes:

1. Describe the concept of Data Structures and assess how the choice of data structures impacts the performance of programs
2. Compare and contrast merits and demerits of various data structures in terms of time and memory complexity.
3. Identify and propose appropriate data structure for providing the solution to the real-world problems.
4. Implement operations like searching, insertion, deletion, traversing mechanism etc. on various data structures
5. Be familiar with advanced data structures such as balanced search trees, hash tables, AVL trees, priority queues, ADT etc.
6. To augment merits of particular data structures on other data structure to develop innovation in subject of study.

Unit No.	Contents	No. of Chapters per unit
1	Introduction: Basic Terminology, Pointer and dynamic memory allocation, Elementary Data Organization, Algorithm Complexity and Time-Space trade-off. Arrays: Array Definition, Representation and Analysis, Single and Multidimensional Arrays, Sparse Matrices. Recursion: Definition, Tail recursion.	3
2	Stack: Array Representation of stack, Linked Representation of Stack, Infix, Prefix and Postfix Expressions. Queues: Array and linked representation and of queues, Circular queue, D-queue, and Priority Queue. Linked list: Representation of Singly Linked Lists, Two-way Header List, Doubly linked list, Generalized linked list.	3
3	Trees: Binary Trees, Binary Search tree, algebraic Expressions, Complete Binary Tree. Extended Binary Trees, Threaded Binary trees, AVL Tree, Huffman algorithm & Huffman tree. Searching and Hashing: Sequential search, binary search, Hash Table, Hash Functions, Collision Resolution Strategies.	3
4	Sorting: Insertion Sort, Bubble Sorting, Quick Sort, Two Way Merge Sort, Heap Sort. Graphs- BFS, DFS. Spanning tree: Minimum Spanning tree, Kruskal's Algorithm, Prim's Algorithm, Applications of graph. File Structures: Physical Storage Media File Organization, Organization of records into Blocks, Sequential Files, Indexing, Primary indices, Secondary indices, B+ Tree index Files, B Tree index Files, Indexing and Hashing Comparisons.	3

References:

1. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia
2. Lipschutz, "Data Structures", TMH
3. R. Kruse et al., "Data Structures and Program Design in C", Pearson Education
4. A.M. Tenenbaum et al., "Data Structures and Program Design in C", Pearson Education
5. K. Loudon, "Mastering Algorithms with C", Shroff Publishers and Distributors

Learning Outcomes:

1. Show the importance of systems analysis and design in solving complex problems.
2. Show how the object-oriented approach differs from the traditional approach to systems analysis and design
3. Explain the importance of modeling and how the Unified Modeling Language (UML) represents an object-oriented system using a number of modeling views.
4. Construct various UML models (including use case diagrams, class diagrams, interaction diagrams, state chart diagrams, activity diagrams, and implementation diagrams) using the appropriate notation.
5. Understand how to implement inheritance, encapsulation, polymorphism, operator overloading.
6. Understand how to implement virtual functions and basic file I/Os.

Unit No.	Contents	No. of Chapters per unit
1	Introduction of object-oriented programming (OOP) and comparison with structured programming. OOP's concepts. Object modeling: Object and Classes. Object and Class Diagrams, Links and associations. Generalization and Aggregation. Dynamic modeling: Events and states, State diagram, Concurrency. Functional modeling: Data flow diagrams Specifying operations. Introduction of UML. Moving from C to C++: Variable definition, strict type checking, inline function, function overloading, default arguments, structure, runtime memory management. Operators and expression: Operators, promotion and type conversion. Control flow, arrays and strings, pointers.	3
2	Classes & Objects: Class Specification, Objects, Arrays of objects, Dynamic objects, Pointers to objects, Access specifiers, defining member functions, Data hiding, Constructors, Destructors, Static members, Functions with class: Friend functions, passing objects as arguments, Returning objects, friend class. Operator overloading: Operator overloading, limitations of increment/decrement operators, overloading using friend functions.	3
3	Inheritance: Inheritance and protected members, protected base class inheritance, Inheriting multiple base classes; Constructors, Destructors and Inheritance, Passing parameters to base class constructors, Granting access, Virtual base classes. Virtual Functions: Virtual function, calling a Virtual function through a base class reference, Pure virtual functions, Abstract classes, Using virtual functions, Early and late binding.	3
4	Generic Programming with template: Function template, overloading function templates, class template, inheritance of class templates. I/O System Basics, File I/O: C++ stream classes, Formatted I/O, I/O manipulators, file stream and the File classes, File operations	3

	Exception Handling, STL: Exception handling fundamentals, Exception handling options STL: An overview, containers, vectors, lists, maps.	
--	---	--

Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLIC ATION
1	Schildt, Herbert: The Complete Reference C++, 4th Edition, Tata McGraw Hill.	2003
2	G. Booch, Object Oriented Analysis and Design, Addison-Wesley.	2004
3	Rumbaugh, J., Object Oriented Modeling and Design, Prentice Hall of India,	2006
4	Stanley B.Lippmann, JoseeLajore: C++ Primer, 4th Edition, Addison Wesley,	2005
5	Paul J Deitel, Harvey M Deitel: C++ for Programmers, Pearson Education.	2009
6	K R Venugopal, RajkumarBuyya, T Ravi Shankar: Mastering C++, Tata McGraw Hill.	1999

Learning Outcomes: A student who successfully fulfills the course requirements will

- a. Build an understanding of the fundamental concepts of computer networking.
- b. Familiarize the student with the basic taxonomy and terminology of the Data Communication area.
- c. Understand the concept data communication within the network environment.
- d. Understand the conflicting issues and resolution techniques in data transmission.
- e. Understand the concept of layering and functions of each layer.
- f. Understand Routing Algorithms and how they work in the internet

Unit No.	Contents	No. of Chapters per unit
1	<p>Introduction:Goals and Applications of Computer Networks, networks hardware& software, Layering, The OSI reference model & TCP/IP model and their comparison, theoretical basis for data communication, Standardization organizations.</p> <p>Physical Layer:Services, Networks Topology, Encoding and modulation methods, Transmission Media & impairment, switching systems.</p>	3
2	<p>Data Link Layer: Design issues, framing, error detection& correction, Flow control and Elementary data link protocols: sliding window protocols, examples of data link layer protocols.</p> <p>Medium Access Control Sub Layer: Channel allocation, LAN protocols, multiple access protocols, Ethernet, Collision free Protocols, IEEE standards.</p>	3
3	<p>Network Layer: Design issues, routing algorithms, congestion control algorithms, internetworking, IP packet & IP Addressing, Subnetting.</p>	3
4	<p>Transport Layer: Design issues, Transport service, element of transport protocols, connection management, TCP window Management, TCP & UDP.</p> <p>Application Layer Overview: Email, DNS, WWW, telnet, FTP. RSA algorithms, Cryptography and Compression Techniques.</p>	3

Suggested Books:

Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLIC ATION
1	A. S Tanenbaum, “Computer Networks, 3 rd Edition”, PHI	2007

2	W. Stallings, "Data and Computer Communication", Macmillan Press	2009
3	Comer, "Computer Networks & Internet", PHI.	2007

Learning Outcomes: A student who successfully fulfills the course requirements will

1. Understand the role of DBMS in an organization.
2. Understand basic database concepts, including the structure and operation of the relational data model.
3. Construct simple and moderately advanced database queries using Structured Query Language (SQL).
4. Understand and successfully apply logical database design principles, including E-R diagrams and database normalization.
5. Design and implement a small database project.
6. Understand the concept of a database transaction and related database facilities, including concurrency control, journaling, backup and recovery, and data object locking and protocols.

Unit No.	Contents	No. of Chapters per unit
1	Introduction to DBMS: An overview of database management system, Database System Vs File System, Database system concepts and architecture, data models, schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure. Data Modeling and Entity Relationship Model: ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree.	3
2	Relational Data Base Model (RDBMS): Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra. Introduction to SQL: Characteristics of SQL, Advantages of SQL, SQL data types and literals, Types of SQL commands, DDL, DML & DCL statements, SQL operators and their use, Tables, views and indexes, Queries and sub queries, Aggregate functions, Insert, update and delete operations, Joins, Unions, Intersection, Minus. Overview of PL/SQL, Triggers and cursors.	3
3	High Level Data Base Design & Normalization: Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependencies, loss less join decompositions, normalization using FD, MVD, and Jds.	3
4	Transaction Processing Concepts: Transaction system, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling. Concurrency Control: Concurrency control mechanism, locking Techniques for concurrency control.	3

Suggested Books:

SI.NO	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICATION/ REPRINT
1.	Database System Concepts” by Abraham Silberschatz and S Sudarshan	
2.	Database Management Systems” by Raghu Ramakrishnan	

1. Understand structure of an Operating system
2. Implement concepts of process management
3. Demonstrate concepts of memory management
4. Understand and implement concepts of file management
5. Analyze various security policies

Unit No.	Contents	No. of Chapters per unit
1	Introduction: Definition and history of Operating System, types of operating systems, Operating system structure, Operating system components, services of Operating System, Introduction to multi-threading. Process concept, PCB, scheduling queues and schedulers, CPU scheduling criteria, Process scheduling, Scheduling algorithms, Multiple-processor scheduling, Real-time scheduling and Algorithm evaluation	3
2	Inter-process communication, Process Synchronization, The Critical-Section problem, synchronization hardware, Semaphores, Classical problems of synchronization, Monitors, Threads: creation, deletion and synchronization. Deadlock, System model, Characterization, Deadlock prevention, Avoidance and Detection, Recovery from deadlock.	3
3	Memory management, Logical and Physical Address Space, Swapping, Paging, Segmentation, Virtual Memory, Demand paging, Page replacement algorithms, Allocation of frames, Thrashing, Page Size and other considerations, Demand segmentation. File systems, File concept, access methods, directory implementation. File system structure,	3
4	Disk Management: Disk structure, Free space management, Disk scheduling methods. Protection and Security, Goals of protection, Domain of protection, Access matrix, Implementation of access Matrix. Case Study: Unix/Windows	3

References:

1. Silberschatz, Galvin” Operating Systems Concept, 7th edition, John wiley and sons
2. William stalling: Operating Systems: Internal and design principles, 7th edition PHI
3. D M Dhamdhare ,“Operating Systems: A Concept-Based Approach” TMH
4. Charles Crowley , “Operating System: A Design-oriented Approach”
5. Gary J Nutt, “Operating Systems: A Modern Perspective”.

Third Semester

OMC 301: Java Programming

Credit: 3

Learning Outcomes:

- Implement Object Oriented Programming Concepts
- Use and create packages and interfaces in a Java program
- Use graphical user interface in Java programs
- Create Applets
- Implement exception handling in Java
- Implement Remote Method Invocation.

Unit No.	CONTENT	No of chapters per unit
1	<p>Introduction and Evolution of Java: Evolution of Java, Byte Code, JDK, JVM, JRE, Data type, Variable, Arrays, Operator, Control Statements, Classes & Objects, Constructor, Methods, this, super keyword, Inheritance, static blocks.</p> <p>Packages, Defining Packages, Using Packages, import and static import, jar utility, classes modifiers: abstract, final; member modifiers: public, protected, default, private, static, final, abstract, synchronized, native, transient, volatile, strictfp, instance of operator.</p>	3
2	<p>Interface: Defining Interfaces, abstract methods declarations, implementing interfaces, extended interfaces, interface references and constants in interfaces</p> <p>Fundamental Classes: Object class, Wrapper classes, String class, immutability, StringBuffer and StringBuilder.</p> <p>Exception handling: Exception Types, Exception class, RuntimeException Class, Error Class, Checked and unchecked Exceptions, Defining new exceptions; Handling: try, catch and finally; throw statement, throws clause.</p>	3
3	<p>Thread: Overview of threads, thread Creation; implementing the runnable interfaces, extending the thread class, Thread States, methods: Running, Yielding, sleeping, joining, waiting and notifying. Synchronized and static synchronized threads.Object Lifetime: Garbage Collection, Reachable Objects, Object Finalization. Nested and Inner Classes</p>	3
4	<p>I/O: The File class, FilenameFilter, Byte Streams: Input and Output streams, Character streams: readers and writers; object serialization</p> <p>Applet: Applet basics, Applet Architecture, Applet Life cycle; Event Handling: Event handling mechanisms, the Delegation Event Model, Event classes, sources of events, Event Listener Interfaces, Adapter classes</p> <p>AWT: AWT Controls, Layout Managers, Frame, Images, Graphics, Fonts, Cursors, Colors, FileDialog box. Swing- Introduction, Advantages over AWT, Swing applications.</p> <p>Networking: Networking Basics, Java and the Net, TCP/IP Client sockets, URL, URLConnection, TCP/IP Server sockets, Datagram</p> <p>Introduction to RMI(Remote Method Invocation): RMI, remote Interfaces, Stubs and skeletons, RMI registry, Bind and Rebind, a simple client server</p>	3

	application using RMI. Introduction to Generic Classes and Collection (List, Set, Map) Vector,ArrayList etc.	
--	---	--

Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICAT ION
1	Naughton, Schildt, "The Complete Reference JAVA2", TMH, 7 th edition	2007
2	Khalid A. Mughal: A Programmer's Guide To JAVA, Addison Wesley, 3 rd edition.	2008

Learning Outcomes:

1. an ability to apply knowledge of mathematics, science, and engineering.
2. an ability to design and conduct experiments, as well as to analyze and interpret data.
3. an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
4. an ability to function on multi-disciplinary teams.
5. an ability to identify, formulate, and solve engineering problems.
6. Able to develop the ability of Software Maintenance.

Unit No.	Contents	No. of Chapters per unit
1	<p>Introduction: Introduction to Software, Software Crisis, Software Processes & Characteristics, Introduction to Software engineering, Software life cycle models, Process Models. Software quality.</p> <p>Software Quality and Reliability: Introduction, Importance of Software Reliability, Failure and Faults, Reliability Models, Software Quality Models, compression of CMM and ISO 9001, Six-Sigma, just in time, total quality management etc.</p>	3
2	<p>Software Project Management: Software Project Management life cycle, Software Project Planning</p> <p>Project Estimation: Size Estimation: Lines of Code & Function Count, Cost Estimation Models, Risk Estimation and Management, Effort estimation, Project monitoring and control.</p> <p>Software Requirements Analysis & specifications: Requirement Engineering, Elicitation techniques, Requirements analysis using DFD, Data dictionaries & ER Diagrams, Requirements documentation, Characteristics & organization of SRS, IEEE Standard for SRS.</p>	3
3	<p>Software Development: Agile method Methodology; Extreme programming; Rapid application development; Clean Room Software Development; Component Based Software Development.</p> <p>Software Design: Design Concepts, Cohesion & Coupling, Function Oriented Design, Object Oriented Design, Software coding techniques and guidelines.</p> <p>Software Testing: Objectives, lifecycle, Testing process, Design of test cases, Levels of Testing, Debugging, testing techniques, Blackbox testing techniques, Whitebox testing techniques, Introduction to functional testing & Structural testing, Object oriented testing and web-based software testing.</p>	3
4	<p>Software Maintenance: Management of Maintenance, Maintenance Process, Maintenance Models, Reverse Engineering, Software Re-engineering, Software Configuration Management. Software Configuration Management vs Software maintenance.</p> <p>Project Scheduling: Basic Concepts, Project Scheduling- Basic Principles, Relationship between People and Effort, Task Network, Scheduling, Gantt and PERT charts, Staffing.</p>	3

Suggested Books:

Sl.NO	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICATION/ REPRINT
1.	Pressman, Roger S., "Software Engineering: A Practitioner's Approach Ed. Boston: McGraw Hill, 2001	2004
2.	Jalote, Pankaj, "Software Engineering Ed.2", New Delhi: Narosa 2002	2002
3.	Sommerville, Ian, "Software Engineering", AWL, 2000 Fairly, "Software Engineering", New Delhi: TMH	2000

Learning Outcomes:

1. Learn the fundamental concept of Internet and Master the fundamentals of website development using HTML.
2. Implement the concepts of designing websites, layouts and formatting using CSS, and data exchange concepts using XML.
3. Building dynamic websites by applying Javascript and DOM.
4. Master the fundamentals Dynamic website development using jquery.
5. Implement Server side scripting concept and database driven approach using PHP.
6. Build dynamic, database-driven web applications, such as use of a WAMP framework (Windows, Apache, MySQL, and PHP) and JavaScript, among others, to develop robust online programs.

Unit No.	Contents	No. of Chapters per unit
1	<p>Internet&HTML Basic - Introduction to Internet, TCP/IP protocol suite, Client Server architecture, n-tier approach, WWW, Web Browsers, E-Mail architecture, searching tools and Search engine Security on Web.</p> <p>HTML -Structure of HTML documents, HTML Elements, Linking in HTML, Image preliminaries, Meta Information, list, Tables, Audio and Video Support with HTML.</p> <p>Basic Interactivity and HTML: Interactive Layout with Frames, FORMS, Form Control, New and emerging Form Elements. use of<div>&</p>	3
2	<p>CSS:Introduction, Benefits of CSS, how css works, types of CSS, class Selector forms text formatting properties, CSS Border, margin properties, Positioning Use of classes in CSS, color properties, concept of Ids pseudoclasses</p> <p>XML understandingIntroduction, Syntax, Document structure, Document type definitions, Namespaces, XML schemas.</p>	3
3	<p>Overview of Javascript, Object orientation and Javascript, JavaScript identifiers, operators, control & Looping structure, Intro of Array, Array with methods, User defined & Predefined functions,Errors and Exception Handling, Event handling, Validations on Forms</p>	3
4	<p>Introduction to PHP, A First PHP Web Page , Variables, Operators and Expressions ,Control Statements , Functions , Arrays, String Handling in PHP, Using Ms-Access/MySQL Databases in PHP Pages.</p>	3

Suggested Books:

Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLIC ATION

1	Computer Networks and Internets with Internet Applications (Third Edition) Author: Douglas E. Comer Publisher: Prentice Hall, ISBN: 0-13-091449-5.	2001
2	Ivan Bayross,” HTML, DHTML, Java Script, Perl & CGI”, BPB Publication.	2002
3	Thomas A Powell, HTML-The Complete Reference, Tata McGraw Hill.	2003
4	Deitel M., Deitel P. J., Goldberg, A. B.: Internet & World Wide Web How to Program, 3rd Edition, Pearson education.	2004
5	<p>Reference Sites:</p> <p>1. www.w3schools.com</p> <p>2. www.devguru.com</p> <p>3: www.python.org/</p>	

Learning Outcomes:

1. Understand and explore Microsoft .Net Framework
2. Describe and apply the advanced techniques of object-oriented programming in web application development.
3. Investigate and create solutions to programming problems using the C# programming language that contain an analysis, solution design and documented code.
4. Design and implement applications and distributed systems on the .NET platform.
5. Deploying .NET Applications

Unit No.	Contents	No. of Chapters
1	<p>Overview of the Microsoft .NET Platform: Future of computing and the Microsoft’s vision, Understanding the motivation behind the .NET platform, Constituents of the .NET platform, CLR, Forces behind the fame, CTS, CLS, Know the role of CIL, Platform independence in .NET, .NET [R]Evolution, advantages, & prospects, Understanding the core: Assemblies (DLL HELL, Metadata, Namespace & Versioning), Deploying the .NET runtime.</p>	3
2	<p>C# Language Syntax: Working with data types & conversions: Strings, Dates & Time, Integers, Performing calculations with mathematical operators, Converting between data types Controlling program execution: IF statements, CASE (SWITCH) statements, FOR, FOREACH Loops, WHILE, DO-WHILE Loops, Storing multiple values with arrays.</p>	3
3	<p>C# .NET Object Oriented Programming:Coding object oriented applications: Dividing code into classes, Adding fields, method properties, events and constructors to classes, Defining scope & visibility, Garbage collector, Inheritance & polymorphism, Overloading methods, Handling errors: Throwing exceptions, Try.....Catch.....Finally, Simplifying maintenance through inheritance: Implementing a base class, Defining virtual and abstract methods, Overriding methods in derived classes, Building applications with Visual Studio: Managing projects with the solution explorer, Setting project properties and adding references, Adding files, folders and code, Compiling, debugging and testing programs, Automating testing with Visual Studio: Creating Visual Studio test projects, Writing Unit tests, Testing classes, properties, method and exceptions.</p>	3
4	<p>Programming Web Applications with ASP.NET:Constructing ASP.NET Web Sites with Visual Studio: Writing HTML pages and forms, Maintaining consistency with Master pages, Designing pages with ASP.NET controls, Styling sites with ASP.NET themes. Processing ASP.NET Web Forms: Initializing Web Forms controls, Activating Web Forms with events, ASP.NET AJAX, Working with XML, ASP.NET MVC Incorporating Relational Databases: Relational database concepts: Selecting, inserting, updating and deleting query syntax.</p>	3

	<p>Creating a SQL Server Database: Defining primary & foreign key relationships.</p> <p>Accessing the database with ADO.NET: Connecting to the database and running SQL commands, Storing user information, Retrieving existing records.</p> <p>Deploying .NET Applications: Installing the .NET Framework, Moving ASP.NET applications to the Web.</p>	
--	--	--

Suggested References

1. Andrew Troelsen , “Pro C# 2010 and the .NET 4 Platform, Fifth edition, A Press, 2010.
2. Ian Griffiths, Matthew Adams, Jesse Liberty, “Programming C# 4.0”, Sixth Edition, O’Reilly, 2010.

Learning Outcomes:

1. Students will create interactive graphics applications in C using one or more graphics application programming interfaces.
2. Students will write program functions to implement graphics primitives.
3. Students will write programs that demonstrate geometrical transformations.
4. Students will demonstrate an understanding of the use of object hierarchy in graphics applications.
5. Students will write programs that demonstrate computer graphics animation.
6. Students will write programs that demonstrate 2D image processing techniques.

Unit No.	Contents	No. of Chapters per unit
1	Graphics Primitives: Display Devices: Refresh Cathode Ray Tube, Raster Scan Display, Plasma display, Liquid Crystal display, Aliasing and Anti-Aliasing. Input-Output Devices. Mathematics for Computer Graphics: Point representation, Vector representation, Matrices, Vector addition and vector multiplication, Line Drawing Algorithms: DDA algorithms, Bresenham’s algorithm, Circle, Ellipse generation algorithms. Viewing Transformation, Clipping: Point Clipping, Line Clipping. Polygon Clipping.	3
2	Filling: Inside Tests, Flood fill algorithm, Boundary-Fill Algorithm and scan-line polygon fill algorithm. Transformation: 2D transformation, Basic Transformations, Composite transformations: Reflection, Shearing, And Transformation between coordinate systems. 3 D transformations.	3
3	Projections: Parallel projection, Perspective projection, Quadric surfaces: Sphere, Ellipsoid, Spline & Bezier Representations: Interpolation and approximation, parametric continuity conditions, Spline specifications. Bezier curves and surfaces. Visible lines and surfaces identification, Hidden surface removal, Fractal Theory.	3
4	Rendering: Illumination models, polygon mesh shading, transparency, shadow, texture. Some advance topics/applications: (i) Animation and morphing, (ii) Virtual reality, (iii) User-interface design, (iv) Fractal graphics, (v) 3D visualization.	3

Suggested Books:

Sl.NO	NAME OF AUTHORS/BOOKS/PUBLISHERS	YEAR OF PUBLICATION/ REPRINT
1.	Donald Hearn and M Pauline Baker, "Computer Graphics" PHI	2001
2.	Godfrey, "Computer Graphics", Schaum Series, TMH	2003
3.	Steven Harrington, "Computer Graphics: A Programming Approach" TMH	2001
4.	Prajapati A.K. "Computer Graphics" PPMEd2	2004

OMC 305(2): Cloud Computing**Credit: 3**

1. Understand basics of cloud infrastructure
2. Understanding the concepts of Data centers
3. Understanding different components of service oriented architecture
4. Getting insight of the cloud storage
5. Demonstration of the cloud infrastructureservices
6. Use the cloud infrastructure services

Unit No.	CONTENT	CONTACT HOURS
1	<p>Introduction to Cloud Infrastructure Cloud Evolution, Cloud Services, Cloud Deployment Types, Main Challenges of Cloud Infrastructure, Cloud Reference Model, Cloud Management, Cloud Structure, Infrastructure Components, Cloud Layers, cloud Dynamics, essential characteristics of cloud computing reference architecture,</p> <p>Exploring Cloud Infrastructures and Virtualized Data Center Architecture Cloud infrastructures; public, private, hybrid. Service provider interfaces; Saas, Paas, Iaas. evolution of data centers, architecture of a modern data center, VDC environments; concept, planning and design, business continuity and disaster recovery principles. cloud service functions, catalog, portal, service lifecycle, automation, and orchestration, physical, virtual, software-defined infrastructure, and infrastructure deployment options</p>	3
2	<p>Understanding Services Oriented Architecture SOA : Introduction , Event driven SOA , SOA 2.0 , Enterprise Service Bus , Service catalogues, Defining SOA Communications , Managing & Monitoring SOA , SOA Security , Relating SOA & Cloud Computing</p>	3
3	<p>Exploring Cloud Infrastructure Services Overview of cloud Infrastructure Services, Measuring the Digital Universe: Cloud storage in the Digital Universe, Cloud storage definition, Provisioning Cloud Storage: Unmanaged cloud storage, Managed cloud storage, creating cloud storage systems, Virtual storage, containers, Exploring Cloud Backup Solutions: Backup types, Cloud backup features, Cloud attached backup, Cloud Storage Interoperability: Cloud Data Management Interface (CDMI), Open Cloud Computing Interface (OCCI).</p>	3
4	<p>Case Study: AWS Cloud Infrastructure Services AWS networking and databases: Virtual private clouds, Cloud models, Private DNS servers (Route 53)), Relational database service – DynamoDB, ElastiCache, Redshift.</p>	3

Suggested Books:

Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICATION
1	Barrie Sisisky ,“Cloud Computing Bible”, Published by Wiley Publishing,Inc.	2011
2	Thomas ,“Cloud Computing: Concepts, Technology & Architecture” ,Erl	2013
3	David S. Linthicum ,“Cloud Computing and SOA convergence in your Enterprise, a step by step guide”	2009
4	Cloud Computing by A. Srinivasan	2014

Learning Outcomes:

1. Define system software and differentiate system software with other softwares.
2. Understand the working of Assembler, Loader/Linker and Macroprocessor.
3. Understand the concept of passes in translators
4. Determine the purpose of linking, and types of linking.
5. Develop the system software according to machine limitations.
6. Compare and Contrast the various text editors.

Unit No.	CONTENT	CONTACT HOURS
1	Machine Architecture: Introduction, System Software and its relation to Machine Architecture, Simplified Instructional Computer (SIC), Architecture of SIC Machine , SIC Programming Examples. Assemblers: Basic Assembler Functions, A Simple SIC Assembler, Algorithm and Data Structures for Assemblers, Machine Dependent Assembler Features - Instruction Formats & Addressing Modes, Program Relocation. Machine Independent Assembler Features – Literals, Symbol-Definition Statements, Expression, Program Blocks, Control Sections and Programming Linking, Assembler Design Operations - One-Pass Assembler, Multi-Pass Assembler	3
2	Loaders and Linkers: Basic Loader Functions - Design of an Absolute Loader, A Simple Bootstrap Loader, Machine-Dependent Loader Features – Relocation, Program Linking, Algorithm and Data Structures for a Linking Loader; Machine-Independent Loader Features - Automatic Library Search, Loader Options, Loader Design Options - Linkage Editor, Dynamic Linkage, Bootstrap Loaders, Editors and Debugging Systems: Text Editors - Overview of Editing Process, User Interface, Editor Structure, Interactive Debugging Systems - Debugging Functions and Capabilities, Relationship With Other Parts Of The System, User-Interface Criteria	3
3	Macro Processor: Basic Macro Processor Functions - Macro Definitions and Expansion, Macro Processor Algorithm and Data Structures, Machine- Independent Macro Processor Features - Concatenation of Macro Parameters, Generation of Unique Labels, Conditional Macro Expansion, Keyword Macro Parameters, Macro Processor Design Options, General- Purpose Macro Processors, Macro Processing Within Language Translators	3
4	Lex and Yacc :Lex and Yacc - The Simplest Lex Program, Recognizing Words With LEX, Symbol Tables, Grammars, Parser-Lexer Communication, The Parts of Speech Lexer, A YACC Parser, The Rules Section, Running LEX and YACC, LEX and Hand- Written Lexers, Using LEX – Regular Expression, Examples of Regular Expressions, A Word Counting Program, Parsing a Command Line. Using YACC – Grammars, Recursive Rules, Shift/Reduce Parsing, What YACC Cannot Parse, A YACC Parser - The Definition Section, The Rules Section, Symbol Values and Actions, The LEXER, Compiling and Running a Simple Parser, Arithmetic Expressions and Ambiguity, Variables and Typed Tokens	3

11 Suggested Books:

Sl.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF
------------	---	----------------

NO.		PUBLICATION
1	Leland.L.Beck: “ System Software: an introduction to systems programming”, 3 rd Edition, Addison-Wesley,	1997
2	John.R.Levine,” Tony Mason and Doug Brown: Lex and Yacc”, O'Reilly, SPD	1998

OMC 305(4): Artificial Intelligence and Neural Network**Credit: 3**

1. Understand the concept of intelligence, types of intelligence.
2. Explain AI, its history and applications.
3. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
4. Implement basic AI algorithms (e.g., standard search algorithms or dynamic programming).
5. To explain the concept of artificial neurons
6. To implement neural networks to solve complex problems

Sl.NO	CONTENT	CONTACT HOURS
1.	Introduction: Introduction to Artificial Intelligence, Foundations and History of Artificial Intelligence, Applications of Artificial Intelligence, Intelligent Agents, Structure of Intelligent Agents The concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation	3
2.	Introduction to Search : Searching for solutions, Uniformed search strategies: Breadth first search, depth first Search, Informed search, strategies, Local search algorithms and optimistic problems, Adversarial Search, Search for games, Alpha - Beta pruning. Knowledge Representation & Reasoning: Propositional logic, Theory of first order logic, Inference in First order logic, Forward & Backward chaining, Resolution, Probabilistic reasoning, Utility theory, Hidden Markov Models (HMM), Bayesian Networks.	3
3.	Characteristics of Neural Networks, Historical Development of Neural Networks Principles, Artificial Neural Networks: Terminology, Models of Neuron, Topology, Basic Learning Laws, Pattern Recognition Problem, Basic Functional Units, Pattern Recognition Tasks by the Functional Units.	3
4.	Feedforward Neural Networks: Introduction, Analysis of pattern Association Networks, Analysis of Pattern Classification Networks, Analysis of pattern storage Networks. Analysis of Pattern Mapping Networks. Feedback Neural Networks Introduction, Analysis of Linear Auto-associative FF Networks, Analysis of Pattern Storage Networks.	3
Sl. NO	NAME OF AUTHORS/BOOKS/PUBLISHERS	YEAR OF PUBLICATION/REPRINT
1.	Nils J. Nilsson, "Artificial Intelligence: A new Synthesis", Harcourt Asia Pvt. Ltd.	2000
2.	Elaine Rich and Kevin Knight, "Artificial Intelligence", 2nd Edition, Tata McGraw-Hill.	2003
3.	Stuart Russell, Peter Norvig, "Artificial Intelligence – A Modern Approach", 2nd Edition, Pearson Education / Prentice Hall of India	2004

Fourth Semester

OMC 401:Data ware house and Data Mining

Credit: 3

Learning Outcomes:

1. define what knowledge discovery and data mining are;
2. define the concept, structure and major issues of data warehousing;
3. develop general awareness of data warehousing project management;
4. apply multi-dimensional modeling techniques in designing data warehouses;
5. apply the online analytical processing (OLAP) technology for decision support;
6. apply data cubing techniques;

Unit No.	Contents	No. of Chapters per unit
1	Introduction to Data Warehousing Concept of Data Warehouse, DBMS verses data warehouse, Data Marts, Metadata, Top down and bottom-up approaches. Dimensional data model, Snow Flake and Star Schemas, ROLAP, Multidimensional database (MOLAP), HYBRID, Data warehouse –Dimensions and their categorization, Measures their categorization and computation, Multi-dimensional database hierarchies.	3
2	Data Warehouse Architecture Operations in OLAP, Advantages of OLAP over OLTP, Three-Tier Data Warehouse architecture, OLAP Guidelines, Multidimensional versus Multi-relational OLAP, Categories of Tools, OLAP Tools and the Internet. SQL Server and Data Warehousing: - Features, System Databases, Creating Databases and Tables, Constraints, Data Manipulation Language, Monitoring and Tuning SQL Server,	3
3	Business Expert Systems, OLTP & OLAP, Data Marts, Data Warehousing, Data Warehouse Architecture. Data Mining core applications, relationship between CRM and BI - Customer Intelligence principles and application options, relations to BI and other. BI market segment, BI trends, BI product examples- SAP, Oracle, Data Mining and Knowledge Discovery: Data Mining Techniques; Applications of Data Mining	3
4	Knowledge Management: Concept of Learning, Knowledge discovery, Analytical Intelligence, Enterprise Intelligence Knowledge Assets, Knowledge Generation, Knowledge Storage, Knowledge Utilization. Knowledge Management Technologies. Quantitative methods for data analysis and knowledge extraction: classification and regression, clustering, association rule.	3

Suggested References:

Sl.NO	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICATION/REPRINT
1.	Alex Berson and Stephen J. Smith, “ Data Warehousing, Data Mining & OLAP”, TataMcGraw – Hill Edition	2007
2.	Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”,Second Edition, Elsevier	2007

Learning Outcomes:

1. Analyze and compare complexity for different types of algorithms for different types of problems.
2. Apply mathematical preliminaries to the analyses and design stages of different types of algorithms.
3. Choose among different types of data structures the best one for different types of problems.
4. Recognize the general principles and good algorithm design techniques for developing efficient computer algorithms.
5. Familiarize with specific algorithms for a number of important computational problems like sorting, searching, and graphs, etc.
6. Decide on the suitability of a specific algorithm design technique for a given problem.

Unit No.	CONTENT	CONTACT HOURS
1	<p>Algorithms: Analysis of Algorithms, Design of Algorithms, and Space and time Complexity of Algorithms, Asymptotic Notations, Growth of function, Recurrences relations, Masters Theorem ,Divide and Conquer Relations.</p> <p>Sorting in polynomial Time: Insertion sort, Merge sort, Heap sort, and Quick sort. Sorting in Linear Time: Counting sort, Radix Sort, Bucket Sort.</p> <p>Binary Search Tree, Red Black Trees and operations, Binomial Heap, B-Tree, Fibonacci Heap.</p>	3
2	<p>Advanced Design and Analysis Techniques: Dynamic programming approach and its application, Chain Matrix Multiplication Problem, Optimal Binary search tree ,TSP, 0/1 knapsack problem, Greedy and Backtracking approach, N-queens problem Hamiltonian cycles, Branch-and-Bound Techniques , and Amortized Analysis.</p>	3
3	<p>Graph Algorithms: Elementary Graph Algorithms, Breadth First Search, Depth First Search, Minimum Spanning Tree, Kruskal’s Algorithms, Prim’s Algorithms, Single Source Shortest Path, Bellman Ford Algorithm, All pair Shortest Path, Flow Network ,.Maximum flow -min cut theorem, Flow Maximization Problem.</p>	3
4	<p>Randomized Algorithms, String Matching,Rabin-Karp algorithm, NP-Hard and NP-Completeness, Cooks theorem, Approximation Algorithms, Sorting Network, Matrix Operations, Polynomials & the Fast Fourier Transformation, Number Theoretic Algorithms.</p>	3

Suggested Books:

Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICATION
1	Horowitz Sahani, “Fundamentals of Computer Algorithms”, Goltgia	2008
2	CoremenLeisersonetal, “Introduction to Algorithms”, PHI	2000
3	Brassard Bratley, “Fundamental of Algorithms”, PHI	2002

Learning Outcomes:

1. Students will able to Use the development tools in the Android development environment.
2. Make UI-rich apps using all the major UI components and describe the life cycles of Activities, Fragments and Intent.
3. Make UI-rich apps using all the widgets and multimedia components that will show the power of Android Programming like Graphics, Sensors etc.
4. Store and manipulate data internally and externally in the Files and also in SQLite Database.
5. Understand the concept of web services, Telephony, and instant messaging and utilize the Google map to add location to their apps.
6. Prepare their apps for distribution on the Google Play Store.

Unit No.	CONTENT	CONTACT HOURS
1	<p>Introduction to Android: Introduction of Android, OHA, Features of Android, History, versions, Android Architecture, Android core building blocks, Android Emulator, Configuring Android Development Environment: Downloading and installing JDK and Net beans, Downloading and installing Android Studio. Creating First Android App: Creating new Android Project, Creating AVD. Android project files: Android Manifest.xml, MainActivity.java, R.java, activity_main.xml etc. Android Tools: DVM, AVD Manager, Android SDK Manager, Android Emulator, DDMS, resource folder. App Components: Activity, Service, Broadcast Receiver, and Content Provider.</p>	3
2	<p>Fragment: Introduction, needs of fragment and Lifecycle of Fragment. Intent: What is Intent, Why Intent? Types of Intent. Layouts: Linear, Relative, Tabular, Frame, and Absolute. Views: Button, Text View, Edit Text, Radio Button, Image View, Toast, Adapter, Spinner, List View, Grid View. Adaptor: Adaptor and Adaptor View. Android Menu: Option menu, context menu, and popup menu. Events: Event Handling, and Listeners.</p>	3
3	<p>Multimedia: Playing audio, video file and Camera. Graphics: Android Graphics and Animation Sensors: How sensors work, category of sensors, sensor framework, and classes and interfaces used to access sensor. Bluetooth: Bluetooth Adapter class, Paired devices, Enable/Disable devices. Thread: Introduction, category and AsyncTask. Maps & Location: What is Google map, Types of Google map, Methods of Google map, Need of Callback methods, and Geocoder class?</p>	3
4	<p>Content Provider: How ContentProvider works, steps to create content provider, URI, CRUD. Files: Internal and External storage. Shared Preference. Storage: Introduction to SQLite Database, SQLiteOpenHelper class, and Cursor. Android Web Services: What is web service, Soap and Restful web service? P2P Communication: Introduction of Instant Messaging.</p>	3

	Introduction of Telephony. Versioning the app: Setting the version publishing the app on Google play, Moniterizing the Android app.	
--	--	--

Suggested Books:

Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICATION
1	Programming Android: Expert Android application Development and professional Android by Reto Meier.	2014
2	Android Black Book, programming android by Zigurd Mednieks, Laird Dormin, G. Blake Meike & Masumi Nakamura.	2015
3	Android Application Development, Black Book by Pradeep Kothari	2014

Learning Outcomes:

1. Understand the security issues of Computer Networks.
2. Describe network security services and mechanisms.
3. Symmetrical and Asymmetrical cryptography.
4. Data integrity, Authentication, Digital Signatures.
5. Various network security applications, IPSec, Firewall, IDS, Web security, Email security, and Malicious software etc
6. Describes the web security and access control methods.

Unit No.	CONTENT	CONTACT HOURS
1	<p>Information Security: Introduction to Security: Attacks, Services & Mechanisms, Security Requirements,</p> <p>Classical Encryption Techniques: Substitution & Transposition techniques, Cryptography, Cryptanalysis, Steganography.</p> <p>Data Encryption Standard: Simplified-DES, DES, Double DES, Meet-in-the-Middle attack, Triples DES, Cryptanalysis, Block Ciphers, Block Cipher Modes Of Operation, Stream Ciphers. Introduction to AES.</p> <p>Encryption Algorithms: Blowfish, International Data Encryption Algorithm, Key Distribution, Diffie-Hallman Algorithm.</p>	3
2	<p>Public Key Encryption: Principles Of Public-Key Cryptosystems, RSA Algorithm, Key Management, Fermat’s & Euler’s Theorem.</p> <p>Message Authentication & Hash Functions: Authentication Requirements, Message Authentication Codes, Hash Functions, Message Digest Algorithm, Secure Hash Algorithm, Digital Signatures, Authentication Protocol, Digital Signature Standard.</p>	3
3	<p>Network & System Security: Authentication Applications: Kerberos X.509, Directory Authentication Service, Electronic Mail Security, Pretty Good Privacy (PGP), S / MIME, IPSec: Authentication Header.</p>	3
4	<p>Web Security: Secure Socket Layer & Transport Layer Security, Secure Electronic Transaction.</p> <p>System Security: Intruders, Viruses, Firewalls, Trusted Systems.</p> <p>Access Control: Authentication Methods, Passwords, Biometrics; Authorization- Access Control Matrix, Multilevel Security Models, Multilateral Security, Inference Control, CAPTCHA, Signature-Based IDS, Anomaly-Based IDS.</p>	3

Suggested Books:

Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICATION
1	William Stallings, “Cryptography and Network Security: Principles and Practice”, 4 th Ed., Prentice Hall, New Jersey.	2009
2	Johannes A. Buchmann, “Introduction to Cryptography”, Springer-Verlag.	2008
3	Atul Kahate, “Cryptography and Network Security”, 2 nd Ed., TMH	2009
4	Bruce Schneier, “Practical Cryptography”, Wiley DreamTech India	2009
5	Eric Cole, “Network Security Bible,” Wiley DreamTech India Pvt. Ltd	2003

Learning Outcomes:

1. Explain what IoT is, evolution of IoT from Internet
2. Describe key technologies in Internet of Things.
3. Identify components needed to provide a solution for certain applications.
4. Understand the challenges as well as application domains of IoT
5. Analyze basic protocols in an IoT environment
6. Design IoT applications in different domain and be able to analyze their performance

UNIT	CONTENTS	Contact Hrs
1	Introduction to Internet of Things: History of IoT, From Internet to Internet of Things, Overview and Motivations. Internet of Things: Definitions and Frameworks and Architecture, General Observations, ITU-T Views, Links to Current and Historical State of the Art.	3
2	Physical Design of IoT– IoT Protocols, IoT communication models, IoT Communication APIs. IoT and M2M – Machine to Machine, Difference between IoT and M2M Functional blocks of IoT, IoT Hardware, IoT Software IoT enabling Technologies– Wireless Sensor Networks, Cloud Computing, Big data analytics, Communication protocols, Embedded Systems.	3
3	Resource Management In The Internet Of Things:Clustering, Synchronisation and Software Agents Potential Application Domains: Home automation, healthcare applications, Industry applications, Transportation, Traffic management, Surveillance applications, Other IoT applications.	3
4	Developing IoTs: Introduction to Python, Introduction to different IoT tools, Developing applications through IoT tools, Developing sensor based application through embedded system platform, Implementing IoT concepts with python. IoT Protocols: Z- wave, BLE, Zigbee, IPv6, 6 LoWPAN, RPL, CORPL, CARP, TLS, DTLS, CoAP, XMPP, MQTT, oneM2M, ETSI M2M, IoT Physical Devices and Endpoints: Raspberry PI/ Arduino as a Case Study	3
Suggested Books:		
Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICATION
1	Daniel Minoli, “Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications”, ISBN: 978-1-118-47347-4, Willy Publications	2012
2	Bernd Scholz-Reiter, Florian Michahelles, “Architecting the Internet of Things”, ISBN 978-3- 642-19156-5 e-ISBN 978-3-642-19157-2, Springer	2010
3	Hakima Chaouchi, “The Internet of Things Connecting Objects to the Web” ISBN : 978-1- 84821-140-7, Willy Publications	2012

Learning Outcomes:

- 2 Distinguish the theoretical and conceptual foundations of distributed computing.
- 2 Recognize the inherent difficulties that arise due to distribution of computing resources.
- 3 Understanding the architectural models of distributed systems
- 4 Recognize the feasibilities and the impossibilities in managing resources.
- 5 Identify the problems in developing distributed applications.

Unit No.	CONTENT	CONTACT HOURS
1	<p>Characterization of Distributed Systems: Introduction, Examples of distributed Systems, Goals and Challenges of Distributed System Pitfalls of Distributed Systems.</p> <p>System Models: Physical Models, Architectural models, Fundamental Models, Limitation of Distributed system, absence of global clock, shared memory, Logical clocks, Lamport’s & vectors logical clocks, Causal ordering of messages, global state, termination detection.</p>	3
2	<p>Distributed Mutual Exclusion: Classification of distributed mutual exclusion, requirement of mutual exclusion algorithms, Token based and non-token based algorithms, performance metric for distributed mutual exclusion algorithms.</p> <p>Distributed Deadlock Detection: system model, resource Vs communication deadlocks, deadlock prevention, avoidance, detection & resolution, centralized dead lock detection, distributed dead lock detection, path pushing algorithms, edge chasing algorithms.</p>	3
3	<p>Agreement Protocols: Introduction, System models, classification of Agreement Problem, Byzantine agreement problem, Consensus problem, Interactive consistency Problem, Solution to Byzantine Agreement problem, Application of Agreement problem</p>	3
4	<p>Distributed Scheduling: Issues in Load Distribution, load Balancing vs Load Sharing. Preemptive Vs Non-Preemptive Transfers</p> <p>Components of Load Distributing Algorithms: Transfer Policy, Selection Policy, Location Policy, Information Policy.</p> <p>Load Distributing Algorithms: Sender Initiated Algorithm, Receiver Initiated Algorithm, Symmetrically Initiated Algorithm, Adaptive Algorithm</p>	3

Suggested Books:

Sl. NO.	NAME OF AUTHERS/BOOKS/PUBLISHERS	YEAR OF PUBLICATION
1	Singhal & Shivaratri, "Advanced Concept in Operating Systems", McGraw Hill	2001
2	Coulouris, Dollimore, Kindberg, "Distributed System: Concepts and Design”, Pearson Ed.	2001
3	Gerald Tel, "Distributed Algorithms", Cambridge University Press	2005

The project proposal should be prepared in consultation with your guide. The project proposal should clearly state the project objectives and the environment of the proposed project to be undertaken. **The project work should compulsorily include the software development.** The project proposal should contain complete details in the following form:

Synopsis of the project proposal (6-8 pages) covering the following aspects may be prepared and send it to your Internal Guide.

- (i) Title of the Project
- (ii) Introduction and Objectives of the Project
- (iii) Project Category (RDBMS/OOPS/Networking/Multimedia/ArtificialIntelligence/Expert Systems etc)
- (iv) Tools /Platform, Hardware and Software Requirement specifications
- (v) Problem Definition, Requirement Specifications (Detailed functional Requirements and Technical Specifications), Project Planning and Scheduling (Gantt chart/PERTchart),
- (vi) Scope of the solution,
- (v) Future scope and further enhancement of the project,
- (vi) Bibliography

PREPARATION FOR THE PROJECT REPORT

1. Project Report Formulation:

The project report **should** contain the following:

- i) Original copy of the Approved Performa and Project Proposal
- ii) Certificate of Originality
- iii) Project documentation

2. The **project documentation** may be about (140 to 180) pages (excluding coding). The project documentation details should not be too generic in nature. Appropriate project report documentation should be done, like, **how you have done the analysis, design, coding, use of testing techniques / strategies, etc., in respect of your project.** To be more specific, whatever the theory in respect of these topics is available in the reference books should be avoided as far as possible. **The project documentation should be in respect of your project only.** The project documentation should include the topics given below. Each and every component shown below carries certain weightage in the project report evaluation.

1. Table of Contents / Index with page numbering
2. Introduction / Objectives
3. System Analysis
4. Identification of Need
5. Preliminary Investigation
6. Feasibility Study
7. Project Planning
8. Project Scheduling
9. Software requirement specifications (SRS)
10. Software Engineering Paradigm applied
11. Data model, Control Flow diagrams, state Diagrams/Sequence diagrams ERD's/Class Diagrams/CRC Models/Collaboration Diagrams/Use-case Diagrams/Activity Diagrams depending upon your project requirements.
12. System Design

13. Modularization details
14. Data integrity and constraints
15. Database design/Procedural Design/Object Oriented Design
16. User Interface Design
17. Coding
18. Validation checks
19. Testing
20. Testing techniques and Testing strategies used *along with the test case designs* and test report.
21. System Security measures (Implementation of security for the project developed)
22. Database/data security
23. Creation of User profiles and access rights
24. Cost Estimation of the Project
25. Reports (sample layouts should be placed)
26. PERT Chart, Gantt chart
27. Future scope and further enhancement of the Project
28. Bibliography
29. Appendices (if any)
30. Glossary.

1. The project report should normally be printed with font size 12(Time Roman) &1.5 line spacing on A4 paper (one side only).
- 2.All the pages, tables and figures must be numbered. Tables and figures should contain titles.
3. If any project report is received in the absence of the approved project proposal Performa (in original), project synopsis, certificate of originality and CD it will be summarily rejected and returned to the student.
4. Throughout the project report, the title of the project should be the same as per the approved synopsis. Signature of the Project Guide in the Certificate of Originality should match with the signature in the Project Proposal Performa (Approved Synopsis) also.
5. **Only one copy of the original project report** in the bound form along with the CD (containing the executable file(s) of the project should be enclosed in the last page) is to be submitted.
7. Title of the project should be kept the same through out the project.