Detailed Syllabus

For

Post Graduate Programme

Regulation 2020

MASTER OF COMPUTER APPLICATIONS

Master of Computer Applications 2020-21
1st Semester [First Year]

	Course Particulars		Credit Particulars				Evaluation							
			Hours Per Week				THEORY			PRACTICAL				
Category	CourseCode	CourseName	L	T	P	C	CIA	SEE	Total	CIA	SEE	Total	Total Marks	Credits
Professional Core Courses	MCA20101	Problem Solving and Data Structures	4			3	70	30	100	-	-	-	100	3
Professional Core Courses	MCA20102	Computer Organization and Architecture	3			3	70	30	100	-	-	-	100	3
Professional Core Courses	MCA20103	Operating Systems	3			3	70	30	100	-	-	-	100	3
Professional Core Courses	MCA20104	Computer Oriented Numerical Methods	5			4	70	30	100	-	-	-	100	4
Professional Core Courses	MCA20105	Business Communication	3			3	70	30	100	-	-	-	100	3
Professional Core Courses	MCA20106	Problem Solving and Data Structures using C Laboratory			2	1				30	70	100	100	1
	MCA20107	Computer Organization and Architecture Laboratory			2	1				30	70	100	100	1
	MCA20108	Operating Systems Laboratory			2	1				30	70	100	100	1
	MCA20109	Business Communication Laboratory			2	1				30	70	100	100	1
Total			**18**		**8**	**20**			**500**			**400**	**900**	**20**

Semester - I

Subject Code	Title of the subject	L	T	P	C	QP
MCA20101	**Problem Solving and Data Structures**	4	0	0	3	A

UNIT:1 **10 Hours**

Representation of Algorithm, Flowchart/Pseudo-code with examples.From algorithms to programs; source code, variables (with data types) variables and memory locations.
Introduction to C: Variables, Keywords, Tokens, Loops: writing programs and evaluation of loops, While, do-while and for loop, break, continue, nested loop.
Functions: Functions (including using built in libraries), Parameter passing in functions, call by value, Passing arrays to functions: idea of call by reference. Function Recursion Examples.

UNIT:2 **10 Hours**

Arrays: Concept, Declaration and Manipulation of Arrays, One Dimensional, Multidimensional Arrayand their Applications,Memory Calculation, Sparse Matrixes.
Pointers: Idea of pointers, Defining pointers, Dynamic Memory Allocation, Use of Pointers in self-referential structures, notion of linked list (no implementation)
Structure: Structures, Defining structures and Array of Structures, Pointers to Structures.

UNIT:3 **10 Hours**

Introduction to Data Structures: Classification of Data Structures, Algorithms, Measuring Space and Time Complexities, Asymptotic Notations, Abstract Data Types.
Stacks and Queues: Introduction to Stack & Queues, Operations on Stacks and Queues, Applications of Stack and Queues.
Linked Lists: Dynamic Memory Management, Single Linked Lists, Double Linked Lists, Circular Linked Lists, Linked Stack & Queues.

UNIT:4 **10 Hours**

Trees: Terminology, Representation, Binary Trees, Binary Search Trees, Searching, Insertion and Deletions Operations in a Binary Search Tree, Height Balanced Trees, B-Trees, B+ Trees, Application of Trees.
Graphs: Terminology, Representation, Path Matrix, Graph Traversal, Shortest Path Problems, Topological Sort.
Searching and Sorting Techniques: Linear and Binary Search, Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Comparison of Sorting Techniques.

UNIT:5 (as per choice of faculty) (06 Hours)
Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures/Lecture by Industry Expert/MOOCS

Text Books

1. PradipDey, ManasGhosh, "Programming in C", First Edition,OxfordUniversity Press, 2011.
2. E. Balagurusamy, "Programming in ANSI C", 4th edition, 2007, McGraw-Hill Publication, New Delhi.
3. Data Structures Using C - Aaron M. Tenenbaum

4. Tremblay, Jean-Paul, and Paul G. Sorenson, "An introduction to data structures with applications", McGraw-Hill, Inc., 1984.

Ref. Books

1. K.R. Venugopal, S.R. Prasad, “Mastering C”, McGraw-Hill Education India.
2.Ellis Horowitz, SartajSahni, Susan Anderson-Freed, “Fundamentals of Data Structures in C”, Second Edition, 2008, Universities Press Pvt. Ltd. Hyderabad.
3. Seymour, Lipchitz. “Data Structures with C.”TMH (2010).

Subject Code	Title of the subject	L	T	P	C	QP
MCA20102	**Computer Organization and Architecture**	3	0	0	3	A

UNIT:1 **(10 Hours)**

Introduction: Basic architecture of computer, Functional units, Operational concepts, Busstructures, Von Neumann Concept.

Basic Processing: Instruction code, Instruction set, Instruction sequencing, InstructionCycle & Execution Cycle, Instruction format, Addressing modes, Micro instruction, Datapath and control path design, Micro programmed vs. Hardwired controlled unit, RISC vs.CISC.

Arithmetic: Design of ALU, Binary arithmetic, Addition and Subtraction of signed number,Multiplication of Positive number, Signed operand multiplication, Division, Floating point
number representation and arithmetic.

Digital Electronics: Boolean algebra, Digital Logic, Truth Tables, K map, Number system,Flip - Flop

UNIT:2 **(10 Hours)**

Memory: Memory Hierarchy, RAM, ROM, Cache memory organization, Mappingtechniques, Virtual memory, Memory Interleaving, Secondary Storage, Flash drives.

UNIT:3 **(10 Hours)**

Input/output: Accessing I/O devices, I/O mapped I/O, Programmed I/O, Memory MappedI/O, Interrupt Driven I/O, Standard I/O interfaces, Synchronous and Asynchronous Datatransfer, DMA data transfer.Introduction to Parallel processing: Flynn's Classification, Pipelining, Super Scalarprocessors, Array processing, vector processing.

UNIT:4 **(10 Hours)**

8085 Microprocessor and Assembly level Programming using 8085 microprocessorModule 5 (6 hours)(as per choice of faculty)Portion covered can be tested through Internal evaluation only not to be included inUniversity examination)

UNIT:5 (as per choice of faculty) (06 Hours)

Portion covered can be tested through Internal evaluation only not to be included in University examination)

Text Books

1. Mano.M. "Computer System and Architecture" (3rd Ed) (PHI).
2. Computer Architecture by Hwang and Briggs. (MGH).
3. Fundamentals of Computer Organisation by M V L N Raja Rao; Scitech publ.
4. Carl Hamacher, ZvonkoVranesic, SafwatZaky, "ComputerOrganization", 5th Edition, McGraw-Hill Education India

Ref. Books

1. William Stalling,"Computer Organization and Architecture", Pearson Education
2. J. P. Hayes, "Computer Architecture and Organization",MGH
3. A.S. Tananbaum, "Structured Computer Organization", Pearson Education

Subject Code	Name of the Subject	L	T	P	C	QP
MCA20103	**Operating systems**	**3**	**0**	**0**	**3**	A

UNIT:1 (10 Hours)
Operating System Introduction- Functions, Characteristics, Structures - Simple Batch, Multi programmed, timeshared, Personal Computer, Parallel, Distributed Systems, Real-TimeSystems, System components, Operating-System services, System Calls, Virtual Machines.Process and CPU Scheduling - Process concepts and scheduling, Operation on processes,Cooperating Processes, Threads, and Interposes Communication Scheduling Criteria,Scheduling Algorithm, Multiple -Processor Scheduling, Real-Time Scheduling.

UNIT:2(10 Hours)
Memory Management and Virtual Memory - Logical versus Physical Address Space,Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging.Demand Paging, Performance of Demanding Paging, Page Replacement, Page ReplacementAlgorithm, Allocation of Frames, Thrashing.

UNIT:3(10 Hours)
File System Interface and Implementation -Access methods, Directory Structure,Protection, File System Structure, Allocation methods, Free-space Management, DirectoryManagement, Directory Implementation, Efficiency and Performance. Process Managementand Synchronization - The Critical Section Problem, Synchronization Hardware,Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors.

UNIT:4 (10 Hours)
Deadlocks - System Model, Dead locks Characterization, Methods for Handling DeadlocksDeadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery fromDeadlock. I/O Management – I/O software and its types, Disk Scheduling. ShellProgramming: Concept of shell, Types of shell, Editors for shell programming (e.g. vi),basics of Shell programming. Case Study- UNIX, LINUX, and Windows NT.

UNIT:5 (as per choice of faculty) (06 Hours)
Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books
1. Silberschatz& Galvin: Operating System Concept, Wiley, Latest Edition.
2. Milan Milenkovic: Operating Systems, Tata McGraw – Hill, Latest Edition.
3. William Stallings: Operating Systems, PHI, Latest Edition.

Reference Books:
1. YashawantKanetkar: Unix Shell Programming, BPB.
2. A.S. Tanenbaum: Modern Operating Systems, latest edition Pearson/PHI.
3. Dhamdhere: Operating Systems, Tata McGraw Hill.
4. Any other book(s) covering the contents of the paper in more depth.

Subject Code	Name of the Subject	L	T	P	C	QP
MCA20104	**Computer Oriented Numerical Methods**	**5**	**0**	**0**	**4**	A

UNIT:1 (10 Hours)
Computing Arithmetic, Significant Digits and Numerical Instability, Root finding methods-Bisection, Newton Raphson, Secant and RegulaFalsi, methods for multiple roots.

UNIT:2(10 Hours)
System of Linear Algebraic Equations and Eigenvalue problems-Gauss Elimination, LU Decomposition- Jacobi-Gauss-Seidel and SOR methods, Interpolation and Approximation spline approximation- Linear, quadratic and Cubic,

UNIT:3(10 Hours)
Differentiation and Integration-Richardson's extrapolation, Gauss Quadrature methods, ordinary differential equations-Initial and Boundary Value Problems, introduction to numerical solutions of Partial Differential Equations.

UNIT:4 **(10 Hours)**
Flowchart and Algorithms and programming in C implementations.

UNIT:5 (as per choice of faculty) **(06 Hours)**
Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Reference Books:
1. Numerical Methods for Scientific and Engineering Computation by M.K. Jain, SRK Iyengar and R.K.Jain
2. Numerical Methods for Engineers by S.C. Chopra and Raymond P. Canale
3. Introductory Methods of Numerical Analysis by Sastry
4. Numerical Analysis by E.W. Cheney and D.R.Kincaid

Subject Code	Name of the Subject	L	T	P	C	QP
MCA20105	**Business Communication**	**3**	**0**	**0**	**3**	A

UNIT:1 (10 Hours)

Nature and scope, the theory of demand, demand function, law of demand and its exceptions, Elasticity of demand, Law of supply and elasticity of supply. Determination of equilibrium price under perfect competition (Simple Numerical problems to be solved). Theory of production and cost, Law of variable proportion, Law of returns to scale

UNIT:2(10 Hours)

Time value of money-Simple and Compound Interest, Cash Flow Diagram, Principle of Economic Equivalence Evaluation of Engineering projects- Present worth method, Future worth method, Annual worth method, Internal rate of return method, Cost-benefit analysis in public projects. Depreciation Policy, Depreciation of capital assets, Causes of depreciation, Straight line method and declining balance method.

UNIT:3(10 Hours)

Fundamentals of Accounting; Accounting as a business function and language of business, Functions and objective of Accounting, Users of Accounting information, Limitations of Accounting, Cyclical nature of business and Accounting cycles, Accounting equations, Accounting events and transactions, classification of transaction and their effect on Accounting

UNIT:4 (10 Hours)

Recording transaction: The journal, The ledger postings, Subsidiary Books ao Accounts, Capital and revenue transactions, Fixed assets and depreciation policy Preparation of Financial Statements: Trial balance, Trading Account, Manufacturing Account, Profit and Loss account, Balance sheet.

UNIT:5 (as per choice of faculty) (06 Hours)

Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books:

1. Riggs, Bedworth and Randhwa , “Engineering Economics”, McGraw Hill Education India
2. R. R. Paul , “Money banking and International Trade”, kalyanipubluisher, NewDelhi
3. H.L. Ahuja , “Principle of Economics”, S. Chand & Co
4. Bal and Sahoo , “Financial Accounting”, S. Chand Publication
5. Jain and Narang , “Financial Accounting” Kalyani Publisher

Reference Books:

1. . A. K. Bhatacharya , “Financial Accounting”, Prentice Hall of India

PRACTICAL / SESSIONAL

Lab Code	Name of the Lab	L	T	P	C	QP
MCA20106	**Problem Solving and Data Structures using C Laboratory**	**0**	**0**	**2**	**1**	

List of Experiments:

Simple Programs using c :

1. Find Area, Perimeter of Square & Rectangle.
2. Find max. Among 3 nos.
3. Check leap year

Programs using Loop :

4. Factorial of Number
5. Prime Number.
6. Perfect Number.
7. Armstrong Number.
8. Floyd's Triangle

Function Programs :

1. Simple Function Problems.
2. Function with call by reference
3. Recursion function e.g. sum of digit, reverse of digit
4. Fibonacci Series
5. Inter conversion of Decimal, Binary & Hexadecimal no.
1. LCM & GCD of numbers

Array & Structure Operations

1. Insert & Delete an element at given location in array.
2. Transpose of matrices
3. Multiplication of matrices
4. Display upper & lower diagonal of matrices
5. Array of Structure e.g. student result, Employee pay slip , Phone bill

Data Structure Programs

1. ADT Stack implementation and use it for evaluation of post-fix expression.
2. Conversion of prefix expression into post-fix form using recursion.
3. Implementation of circular queue (using array) with menu options like insert, delete, display and exit.
4. Implementation of a priority queue (using pointers) and use it to organize student records prioritized by marks.
5. Implementation of ADT doubly linked circular list to hold strings and use it for organizing a sequence of cities constituting at our program.
6. Implementation of a binary search tree with menu options: Construct a tree, insert anode, delete anode, traverse and display preorder, in order and post order sequence of its nodes.
7. Implementation of di-graphs using adjacency matrix and find the transitive closure using Warshall's algorithm.

8. Implementation of a weighted graph and find minimal cost spanning tree using PRIM's Algorithm.
9. Generate 70 random integers in a given range and sort them using quick sort. Apply both binary search and Interpolation search to locate a given integer and compare the search algorithms based on the number of comparisons / probes required or a successful as well as unsuccessful search..
10. Implementation of Sorting and Searching

Lab Code	Name of the lab	L	T	P	C	QP
MCA20107	**Computer Organization and Architecture Lab**	**0**	**0**	**2**	**1**	

List of Experiments:

I–CYCLE: DigitalLogicDesignExperiments:
1. Multiplexers & Decoders
2. Counters
3. Shift Registers
4 Binary Adders&Subtractors
5 A L U

II–CYCLE: 8085AssemblyLanguageProgramming:
1. 8085AssemblyLanguageProgramming according to the course microprocessorsusingthefollowingtrainers:
KeyboardMonitorof8085µPTrainer.SerialMonitorof8085µPTrainerwithTerminal 8085LineAssemblerof8085µPTrainerwithPCasTerminal8085CrossAssemblerusingIn-CircuitEmulator(ICE)with8085µPTrainerandPCasTerminalGradedProblemsaretobeusedaccording tothesyllabusof
COMPUTER ORGANIZATION
PentiumclassPCarchitecturefamiliarizationhardware&softwarepartsdemonstration, roubleshooting of PC, Laptops, Server and Loading of Operating System, Antivirus andother software packages

Lab Code	Name of the Lab	L	T	P	C	QP
MCA108	**Operating System Lab**	**0**	**0**	**2**	**1**	

OBJECTIVES:

1. The student should be made to:
2. Learn shell programming and the use of filters in the UNIX environment.
3. Be exposed to programming in C using system calls.
4. Learn to use the file system related system calls.
5. Be exposed to process creation and inter process communication.
6. Be familiar with implementation of CPU Scheduling Algorithms, page replacement algorithms and Deadlock avoidance

List of Experiments:

1. Basics of UNIX commands.
2. Shell Programming.
3. Implement the following CPU scheduling algorithms
   a) Round Robin b) SJF c) FCFS d) Priority
4. Implement all file allocation strategies
   a. Sequential
   b. Indexed
   c. Linked
5. Implement Semaphores
   Implement all File Organization Techniques

a. Single level directory
b. Two level
c. Hierarchical
d. DAG

6. Implement Bankers Algorithm for Dead Lock Avoidance
7. Implement an Algorithm for Dead Lock Detection
8. Implement e all page replacement algorithms
    a. FIFO
    b. LRU
    c. LFU
9. Implement Shared memory and IPC
10. Implement Paging Technique of memory management.
11. Implement Threading & Synchronization Applications

Lab Code	Name of the lab	L	T	P	C	QP
MCA20109	**Business Communicative Practice Lab**	**0**	**0**	**2**	**1**	

List of Experiments:

I–CYCLE: Speaking : Oral communication

Work-related situations, e.g.: Greeting an acquaintance/ friend, introducing oneself, introducing a friend to another friend, breaking off a conversation politely, leave-taking; making and responding to inquiries; expressing an opinion; expressing agreement/ disagreement, contradicting/ refuting an argument; expressing pleasure, sorrow, regret, anger, surprise, wonder, admiration, disappointment etc. Narrating or reporting an event; Describing people, objects, places, processes etc. Ordering / directing someone to do something Making requests; accepting / refusing a request Expressing gratitude; responding to expressions of gratitude Asking for or offering help; responding to a request for help

Asking for directions (e.g. how to reach a place, how to operate a device etc.) and giving directions asking for and granting/ refusing permission prohibiting someone from doing something suggesting, advising, persuading, dissuading, making a proposal praising, complimenting, felicitating expressing sympathy (e.g. condolence etc.) Complaining, criticizing, reprimanding

II–CYCLE: Reading

Students will be given practice in reading and comprehending 6-8 simple passages of 100-300 words each, on topics of general as well as professional interest. The texts will be supported by suitable exercises designed to foster comprehension skills and vocabulary enrichment, together with study skills (note making) and reference skills (using a dictionary). Practice will be provided in the important sub-skills of reading which are introduced in Module 2 of the theory component.

II–CYCLE: Writing

Writing short paragraphs on given topics or topics of one's choice; social and business letters; reports; applications ; resumes ; summaries The principles of 'Process Writing' should be used to teach writing skills. i pre-writing : generating ideas, brain-storming, idea mapping, outlining ii writing : generating a first draft ; reviewing, redrafting, editing iii post-writing : making a presentation ; discussion and feedback, preparing the final draft. b. Soft skills practice 10 hours Activities designed to highlight leadership and 'team' skills; Group discussion

Master of Computer Applications 2020-21
2nd Semester [First Year]

	Course Particulars		Credit Particulars				Evaluation							
			Hours Per Week				THEORY			PRACTICAL				
Category	CourseCode	CourseName	L	T	P	C	CIA	SEE	Total	CIA	SEE	Total	Total Marks	Credits
Professional Core Courses	MCA20201	Design and Analysis of Algorithms	4			4	70	30	100	-	-	-	100	4
Professional Core Courses	MCA20202	Object Oriented Programming using JAVA	3			3	70	30	100	-	-	-	100	3
Professional Core Courses	MCA20203	Database Management Systems	3			3	70	30	100	-	-	-	100	3
Professional Core Courses	MCA20204	Discrete Mathematics	5			4	70	30	100	-	-	-	100	4
Professional Elective Courses	MCA20205	Professional Elective 1 *	3			3	70	30	100	-	-	-	100	3
	MCA20206	Design and Analysis of Algorithms Laboratory			2	1				30	70	100	100	1
	MCA20207	Object Oriented Programming using JAVA Laboratory			2	1				30	70	100	100	1
	MCA20208	Database Management Systems Laboratory			2	1				30	70	100	100	1
Total			**18**		**6**	**20**			**500**			**300**	**800**	**20**

* Professional Elective – I (Choose any one)
1. MCA 205A: Artificial Intelligence & Expert Systems
2. MCA 205B: Cloud Computing
3. MCA 205C: ERP and E-commerce
4. MCA 205D: Advanced Computer architecture
5. MCA 205E: Advanced Computer architecture
6. MCA 205F: PHP and My SQL
7. MCA 205G: Environmental Engineering

Semester – II

Subject Code	Name of the Subject	L	T	P	C	QP
MCA20201	**Design and Analysis of Algorithms**	**4**	**0**	**0**	**4**	A

UNIT:1 Introduction **(10 Hours)**

Notion of an Algorithm – Fundamentals of Algorithmic Problem Solving – Important Problem Types – Fundamentals of the Analysis of Algorithm Efficiency – Analysis Framework – Asymptotic Notations and its properties – Mathematical analysis for Recursive and Non-recursive algorithms. Amortized Analysis.

UNIT:2 Brute Force And Divide-And-Conquer **(10 Hours)**

Brute Force – Closest-Pair and Convex-Hull Problems-Exhaustive Search – Traveling Salesman Problem – Knapsack Problem – Assignment problem. Divide and conquer methodology – Merge sort –Heap Sort- Quick sort – Binary search – Multiplication of LargeIntegers – Strassen's Matrix Multiplication-Closest-Pair and Convex-Hull Problems.

UNIT:3 Dynamic programming and Greedy Technique **(10 Hours)**

Computing a Binomial Coefficient – Warshall's and Floyd's algorithm – Optimal Binary Search Trees – Knapsack Problem and Memory functions. Greedy Technique– Prim's algorithm- Kruskal's Algorithm- Dijkstra's Algorithm-Huffman Trees.

UNIT:4 Iterative Improvement **(10 Hours)**

The Simplex Method-The Maximum-Flow Problem – Maximum Matching in Bipartite Graphs- the Stable marriage Problem.

Coping With The Limitations Of Algorithm Power

Limitations of Algorithm Power-Lower-Bound Arguments-Decision Trees-P, NP and NP-Complete Problems–Coping with the Limitations – Backtracking – n-Queens problem – Hamiltonian Circuit Problem – Subset Sum Problem-Branch and Bound – Assignment problem – Knapsack Problem – Traveling Salesman Problem- Approximation Algorithms for NP – Hard Problems – Traveling Salesman problem – Knapsack problem**.**

UNIT:5 (as per choice of faculty) **(06 Hours)**

(as per choice of faculty)
Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books

1. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", Third Edition, PHI Learning Private Limited, 2012.
2. AnanyLevitin, "Introduction to the Design and Analysis of Algorithms", Third Edition, Pearson Education, 2012. .

Reference Books:

1Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, "Data Structures and Algorithms", Pearson Education, Reprint 2006.

2. Donald E. Knuth, "The Art of Computer Programming", Volumes 1& 3 Pearson Education, 2009. Steven S. Skiena, "The Algorithm Design Manual", Second Edition, Springer, 2008
3. http://nptel.ac.in/6. David Makinson, "Sets, Logic and Maths for Computing", Springer Indian Reprint, 2011.

Subject Code	Title of the subject	L	T	P	C	QP
MCA20202	**Object Oriented Programming using JAVA**	3	0	0	3	A

UNIT:1 **10 Hours**

Fundamentals of object oriented programming: Introduction to Object OrientedParadigm, procedural Paradigm, An overview of classes, objects and Methods, inheritanceand polymorphism.

Features of Java, Data types, operators & expressions, control structures, arrays, Classes, objects & methods, constructors, garbage collection, access qualifiers, string handling – string operations, character extraction, string comparison, searching and modifying strings, String Buffer, packages and interfaces, Wrapper classes.

UNIT:2 **10 Hours**

Inheritance: single and multilevel inheritance, method overriding, abstract class, use of super and final keywords. Exception Handling: Exception types, uncaught exceptions, multiple catch clauses, nested try statements, built-in exceptions, creating your own exceptions. Multithreading: Java thread model, creating multiple threads, thread priorities, synchronization, interthread communication, suspending, resuming and stopping threads. .

UNIT:3 **10 Hours**

Applets: Local & Remote Applets, Applet Architecture, Passing Parameters to Applets, Applet Graphics, Adapter Class. I/O Streams: Console I/O – reading console input, writing console output, Files I/O – Byte Streams, Character Streams, Collection Interfaces & Classes, Delegation Event Model

UNIT:4 **10 Hours**

AWT Classes: Window fundamentals, working with graphics, working with color & fonts. AWT controls, layout managers & working with menus, JFrames. Swing Classes, Java Beans, Servlet classes & Life Cycle.

UNIT:5 (as per choice of faculty) **(06 Hours)**

(as per choice of faculty)

Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures/Lecture by Industry Expert/MOOCS

Text Books

1. Herbert Schildt, The Complete Reference Java 2, Fourth Edition, Tata McGraw Hill-2001
2. Liang Y.Daniel, Introduction to Java Programming (7th Edition), 2009, Pearson Education.

Ref. Books

1. Steven Holzner, Java 1.2, BPB-1998
2. E. Balaguruswami, Programming with Java - Second Edition, Tata McGraw Hill-1998. Mughal K.A., Rasmussen R.W., A Programmer‟s Guide to Java Certification, Addison-Wesley, 2000 `

Subject Code	Name of the Subject	L	T	P	C	QP
MCA20203	**Database Management System**	**3**	**0**	**0**	**3**	A

UNIT:1 Introduction **(10 Hours)**

Introduction and applications of DBMS, Purpose of data base, Data, Independence, Database System architecture- levels, Mappings, Database, users and DBA

Relational Model

Structure of relational databases, Domains, Relations, Relational algebra – fundamental operators and syntax, relational algebra queries, tuple relational calculus

UNIT:2 **(10 Hours)**

Entity-Relationship model: Basic concepts, Design process, constraints, Keys, Design issues, E-R diagrams, weak entity sets, extended E-R features – generalization, specialization, aggregation, reduction to E-R database schema.

Relational Database design:

Functional Dependency – definition, trivial and non-trivial FD, closure of FD set, closure of attributes, irreducible set of FD, Normalization – 1NF, 2NF, 3NF, Decomposition using FD- dependency preservation, BCNF, Multi- valued dependency, 4NF, Join dependency

UNIT:3(10 Hours)

Query Processing & Query Optimization:

Query processing and optimization: Evaluation of Relational Algebra Expressions, Query optimization, Query cost estimation.

Transaction Management:

Transaction processing and concurrency control: Transaction concepts, properties of transaction, concurrency control, locking and Timestamp methods for concurrency control schemes. Database Recovery System, Types of Data Base failure & Types of Database Recovery, Recovery techniques,

UNIT:4 **(10 Hours)**

Detailed Storage Architecture, Storing Data, Magnetic Disk, RAID, Other Disks, Magnetic Tape, Storage Access, File & Record Organization, File Organizations & Indexes, Order Indices, B+ Tree Index Files, Hashing Data Dictionary

UNIT:5 (as per choice of faculty) **(06 Hours)**

(as per choice of faculty) Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, “Database Systems Concepts”, McGraw-Hill Education , New Delhi
2. RamezElmasri and Shamkant B. Navathe, “Fundamentals of Database Systems”, Pearson Education Inc., New Delhi.

Reference Books:

1. Hector Garcia-Molina, Jeffret D. Ullman, JennifferWidom, “Database Systems: A Complete Book”, Pearson Education Inc., New Delhi.
2. C. J. Date “An introduction to Database System”, Pearson Education Inc., New

Delhi.
3. Bipin Desai, "An introduction to Database System", Galgotia Publications.
4. Peter Rob & Carlos Coronel, "Database Systems: Design, Implementation, and Management", CENGAGE Learning India Pvt. Ltd., New Delhi.
5. Mark L. Gillenson, "Fundamentals of Database Management Systems", Wiley India Pvt. Ltd., New delhi.
6. Raghu Ramakrishnan, Johannes Gehrke, "Database Management Systems", McGraw-Hill Education (India), New Delhi.

Subject Code	Name of the Subject	L	T	P	C	QP
MCA20204	**Discrete Mathematical Structures**	**4**	**0**	**0**	**4**	A

UNIT:1 (12 Hours)
Propositional logic, Propositional Equivalence, Predicates and Quantifiers, Nested Quantifiers, Proof methods and Strategies, Sequences and Summations, Mathematical Induction, Recursive definition and structural induction.

UNIT:2(12 Hours)
Recurrence relation, Solution to recurrence relation, Generating functions, Inclusion and exclusion, Relation and their properties, Closure of relations, Equivalence relations, Partial orderings.

UNIT:3(16 Hours)
Algebraic systems, Lattices, Distributive and Complemented Lattices, Sub-lattices, Boolean Lattices and Boolean Algebra, Boolean Functions and Boolean Expressions.
Semi groups, Monoids, Groups, Sub groups, Cosets, Lagrange theorem, Permutation groups, isomorphism, Homomorphism, Normal subgroups

UNIT:4 (10 Hours)
Basic Definitions – Some Special Graphs – Matrix , Representation of Graphs -— Paths and circuits - Eulerian and Hamiltonian Graphs – connected graphs, Planar graph, Graph coloring ,Trees - Spanning Trees - Rooted trees – Binary Trees, Minimum Spanning tree -Kruskal's algorithm , Prim's algorithm , Tree Traversal.

UNIT:5 (as per choice of faculty) (06 Hours)
Portion covered can be tested through Internal evaluation only not to be included inUniversity examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books:
1. *L. Liu and D. Mohapatra, "Elements of Discrete Mathematics", Third Edition, 2008, Tata McGraw Hill Education, New Delhi*

Ref. Books

1. *Kenneth H. Rosen, "Discrete Mathematics and its Applications", Sixth Edition, 2008, Tata McGraw Hill Education , New Delhi*
2. *N. Deo, Graph Theory and Applications to Engineering and Computer Science, Prentice Hall of India*
3. *Discrete Mathematics by Schaum's Outlines(Second Edition)*
4. *Ralph P. Grimaldi, "Discrete and Combinatorial Mathematics", Fifth Edition, 2005,Pearon Education, New Delhi*

Subject Code	Name of the Subject	L	T	P	C	QP
MCA20205A	**Artificial Intelligence and Expert Systems**	**3**	**0**	**0**	**3**	**A**

UNIT:1 **(10 Hours)**

What is Artificial Intelligence? AI Technique, Level of the Model, Problem Spaces, and Search: Defining the Problem as a State Space Search, Production Systems, Problem Characteristics, Production System Characteristics, and Issues in the Design of Search Programs. Heuristic Search Techniques: Generate-and-Test, Hill Climbing, Best-first Search, Problem Reduction, Constraint Satisfaction, Means-ends Analysis

UNIT:2 **(10 Hours)**

Knowledge Representation: Representations and Mappings, Approaches to Knowledge Representation, **Using Predicate Logic**: Representing Simple Facts in Logic, Representing Instance and ISA Relationships, Computable Functions and Predicates, Resolution, Natural Deduction. **Using Rules**: Procedural Versus Declarative Knowledge, Logic Programming, Forward Versus Backward Reasoning, Matching, Control Knowledge. **Symbolic Reasoning Under Uncertainty**: Introduction to Non monotonic Reasoning, Logics for Non monotonic Reasoning, Implementation Issues, Augmenting a Problem-solver, Depth-first Search, and Breadth-first Search. **Weak and Strong Slot-and-Filler Structures**: Semantic Nets, Frames, Conceptual Dependency Scripts, CYC.

UNIT:3 **(10 Hours)**

Game Playing: The Mini-max Search Procedure, Adding Alpha-beta Cutoffs, Iterative Deepening. **Planning**: The Blocks World, Components of a Planning System, Goal Stack Planning, Nonlinear Planning Using Constraint Posting, Hierarchical Planning, Other Planning Techniques. **Understanding**: What is Understanding, What Makes Understanding Hard?, Understanding as Constraint Satisfaction.

Natural Language Processing: Introduction, Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing, Statistical Natural Language Processing, Spell Checking.

UNIT:4 **(10 Hours)**

Learning: Rote Learning, Learning by Taking Advice, Learning in Problem-solving, Learning from Examples: Induction, Explanation-based Learning, Discovery, Analogy, Formal Learning Theory, Neural Net Learning and Genetic Learning. **Expert Systems**: Representing and Using Domain Knowledge, Expert System Shells, Explanation, Knowledge Acquisition.

UNIT:5(As per choice of faculty). **06 Hours**

(as per choice of faculty) Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books:

1. *Elaine Rich, Kevin Knight, &Shivashankar B Nair, Artificial Intelligence, McGraw Hill, 3rd ed.,2009*

Reference Books:

1. *Introduction to Artificial Intelligence & Expert Systems, Dan W Patterson, PHI.,2010*
2. *S Kaushik, Artificial Intelligence, Cengage Learning, 1st ed.2011*

Lab Code	Name of the Lab	L	T	P	C	QP
MCA20206	**Design and analysis algorithms lab**	**0**	**0**	**2**	**1**	

objectives:
Design, develop and implement the specified algorithms for the following problems using C/C++ Language in LINUX /Windows environment.

List of Experiments:

1. Sort a given set of elements using the Quicksort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.The elements can be read from a file or can be generated using the random number generator.
2. Using OpenMP, implement a parallelized Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
3. a. Obtain the Topological ordering of vertices in a given digraph.
   b.ompute the transitive closure of a given directed graph using Warshall's algorithm.
4. Implement 0/1 Knapsack problem using Dynamic Programming.
5. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
6. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.
7. a. Print all the nodes reachable from a given starting node in a digraph using BFS method.

b. Check whether a given graph is connected or not using DFS method.

8. Find a subset of a given set S = {sl, s2,.....,sn} of n positive integers whose sum is equal to a given positive integer d. For example, if S={1, 2, 5, 6, 8} and d = 9 there are two solutions{1,2,6}and{1,8}.A suitable message is to be displayed if the given problem instance doesn't have a solution.
9. Implement any scheme to find the optimal solution for the Traveling Salesperson problem then solve the same problem instance using any approximation algorithm and determine the error in the approximation.
10. Find Minimum Cost Spanning Tree of a given undirected graph using Prim's
    a. algorithm.
11. Implement All-Pairs Shortest Paths Problem using Floyd's algorithm. Parallelize thisAlgorithm
12. implement it using OpenMP and determine the speed-up achieved.
13. Implement N Queen's problem using Back Tracking.

List of Practice Experiments:

1. Write C++ programs to implement the following:

a) Prim's algorithm.
b) Kruskal's algorithm.

2. Write a C++ program to find optimal ordering of matrix multiplication. (Note: Use Dynamic programming method).
3. Consider the problem of eight queens on an (8x8) chessboard. Two queens are said to attack each other if they are on the same row, column, or diagonal.
4. Write a C++ program that implements backtracking algorithm to solve the problem i.e.place eight non-attacking queens on the board.
5. Write a C++ program to find the strongly connected components in a digraph.
6. Write a C++ program to implement file compression (and un-compression) Using Huffman's algorithm.
7. Write a C++ program to implement dynamic programming algorithm to solve all pairs shortest path problem
8. Write a C++ program to solve 0/1 knapsack problem using the following:
   a) Greedy algorithm.
   b) Dynamic programming algorithm.
   c) Backtracking algorithm.
   d) Branch and bound algorithm.
9. Write a C++ program that uses dynamic programming algorithm to solve the optimal binary search tree problem.
10. Write a C++ program for solving traveling sales persons problem using the following:
    a) Dynamic programming algorithm.
    b) The back tracking algorithm.
    c) Branch and Bound.

REFERENCEBOOKS :

1. Richard F.Gilberg, BehrouzA.Forouzan, Thomson, "Data Structures, A PseudocodeApproach with C++", 1st ed., Business Information Press, 2007.
2. D.S.Malik, Thomson, "Data Structures Using C++", 1st ed., Cengage Learning, 2007.
3. Ellis Horowitz, SatrajSahni and Rajasekharam, "Fundamentals of Computer Algorithms", 2nd ed., Galgotia publications pvt.Ltd, 2006.

Lab Code	Name of the Lab	L	T	P	C	QP
MCA20207	**Programming with Java Lab**	**0**	**0**	**2**	**1**	

LIST OF EXPERIMENTS:

1. Programs to illustrate constructors.
2. Programs to illustrate Overloading & Overriding methods in Java.
3. Programs Illustrate the Implementation of Various forms of Inheritance. (Ex. Single, Hierarchical, Multilevel inheritance….)
4. Program which illustrates the implementation of multiple Inheritance using interfaces
5. in Java.
6. Program to illustrate the implementation of abstract class.
7. Programs to illustrate Exception handling
8. Programs to create packages in Java.
9. Program to Create Multiple Threads in Java.
10. Program to Implement Producer/Consumer problem using synchronization.
11. Program to Write Applets to draw the various polygons.
12. .Create and Manipulate Labels, Lists, Text Fields, Text Areas & Panels
13. Handling Mouse Events & Keyboard Events.
14. Using Layout Managers.
15. Create& Manipulate the Following Text Areas, Canvas, Scroll bars, Frames, Menus, Dialog Boxes. Programs, which illustrate the manipulation of strings.
    a. Ex. 1. Sorting an array of Strings.
    1. Frequency count of words & Characters in a text.
16. Programs, which illustrate the use of Streams.
17. Java Program that reads on file name from the user and displays the contents of file.
18. Write an applet that displays a simple message.
19. Write an applet that computes the payment of a loan based on the amount of the loan, theinterest rate and the number of months. It takes one parameter from the browser: Monthlyrate; if true, the interest rate is per month; Otherwise the interest rate is annual.
20. Write a Java program that works as a simple calculator. Use a grid layout to arrange Buttons for the digits and for the + - X % operations. Add a text field to display the result.
21. Write a Java program for handling mouse events.
22. Write a Java program for creating multiple threads
23. Write a Java program that correctly implements producer consumer problem using theconceptof inter thread communication.
24. Write a Java program that lets users create Pie charts. Design your own user interface (with AWT)
25. Write a Java program that allows the user to draw lines, rectangles and ovals.
26. Write a Java program that illustrates how run time polymorphism is achieved.

TEXT BOOK

1. THE COMPLETE REFERENCE JAVA J2SE 5TH EDITION BY – HERBERT SCHILDT (TMH)

REFERENCE BOOKS

1. THE COMPLETE REFERENCE JAVA 2 (Fourth Edition) BY - PATRICK NAUGHTON & HERBETSCHILDT (TMH)
2. PROGRAMMING JAVA - DECKER&HIRSH FIELD VIKAS PUBLISKING (2001)

(THOMSONLEARNING) (SECOND EDITON)
3. INTRODUCTION TO JAVA PROGRAMMING - Y.DANIEL LIANG PHI(2002)
4. OBJECT ORIENTED PROGRAMMING THROUGH JAVA 2 BY - THAMUS WU
(Mc.Graw Hill)
5. JAVA 2 - DIETEL & DIETEL (PEARSON EDUCATION)
6. INTRODUCTION TO JAVA – BALA GURU SWAMY

Lab Code	Name of the Lab	L	T	P	C	QP
MCA20208	**Database Management System lab**	**0**	**0**	**2**	**1**	

CourseDescription:Thiscourseexploresdatabaseprogrammingusingbothnativeand mbeddedANSI-standardStructuredQueryLanguage(SQL).Topicsincludeenterprisedatabasemanagementsystems,databasemiddleware,datadefinitionlanguage,datamanipulationlanguage,datacontrollanguage,databasequeriesreporting,queryoptimization,anddatabaseviews.Studentassignmentsincludedatabasecreation,querydesignandprogramming,anddatabasemanipulationviaembeddedSQLcallsfromaprogramminglanguage.

Course Goal:Successful graduates of this course should be able to:

1. Understand the fundamentalsofarelationaldatabase
2. Understandthefundamentalsofclient-server and multi-tieredapplications
3.UnderstandtheuseofStructuredQueryLanguage(SQL)asadatadefinitionlanguage,datamanipulationlanguage,anddatacontrollanguage
4. Understand and write SQL/PL_SQLqueriestocreate,report,andupdatedatain relationaldatabase
5. Understandthepurposeofandbeabletocreateviews,scripts,triggers,and transactions
6. Understandandbeabletoimplementthefundamentalsofsecurityand permissionsinSQLServer
7. Designentityrelationshipmodelsforabusinessproblemanddevelopa Normalizeddatabasestructure

LIST OF EXPERIMENTS:

1. Creation of a database and writing SQL queries to retrieve information from the database.
2. Performing Insertion, Deletion, Modifying, Altering, Updating and Viewing records based on conditions.
3. Creation of Views, Synonyms, Sequence, Indexes, Save point.
4. Creating an Employee database to set various constraints.
5. Creating relationship between the databases.
6. Study of PL/SQL block.
7. Write a PL/SQL block to satisfy some conditions by accepting input from the
7. user.
8. Write a PL/SQL block that handles all types of exceptions.
9. Creation of Procedures.
10. Creation of database triggers and functions
11. Mini project (Application Development using Oracle/ Mysql)
    a) Inventory Control System.
    b) Material Requirement Processing.
    c) Hospital Management System.
    d) Railway Reservation System.
    e) Personal Information System.
    f) Web Based User Identification System.
    g) Timetable Management System.
    h) Hotel Management System

Reference Books:

1.IntroductiontoRelationalDatabasesandSQLProgramming,ChristopherAllen,SimonChatwin,CatherineA.VrearyTataMcGraw-Hill
2. OracleSQLandPL/SQLHandbook,JohnAdolphPalinski,PearsonEducation

3. Oracle11iPL/SQLProgramming,ScottUrman,TataMcGraw-Hill
4. MySQL:TheCompleteReference,VikramVaswani,TataMcgraw-Hill
5. MySQLBible,SteveSuehring,Wiley

	Course Particulars		Credit Particulars				Evaluation							
			Hours Per Week				THEORY			PRACTICAL				
Category	CourseCode	CourseName	L	T	P	C	CIA	SEE	Total	CIA	SEE	Total	Total Marks	Credits
Professional Core Courses	MCA20301	Compiler Design	3			3	70	30	100	-	-	-	100	3
Professional Core Courses	MCA20302	Computer Networks	3			3	70	30	100	-	-	-	100	3
Professional Core Courses	MCA20303	Python Programming	4			4	70	30	100	-	-	-	100	4
Professional Core Courses	MCA20304	Software Engineering	4			4	70	30	100	-	-	-	100	4
Professional Elective Courses	MCA20305	Professional Elective 2 **	3			3	70	30	100	-	-	-	100	3
	MCA20306	Computer Networks Laboratory			2	1				30	70	100	100	1
	MCA20307	Python Programming Laboratory			2	1				30	70	100	100	1
	MCA20308	Software Engineering Laboratory			2	1				30	70	100	100	1
Total			**17**		**6**	**20**			**500**			**300**	**800**	**20**

**Professional Elective-II (Choose any one)
1. MCA 305A: Data Analytics
2. MCA 305B: Cryptography and Network Security
3. MCA 305C: Machine Learning
4. MCA 305D: Computer Vision
5. MCA 305E: Big Data Analytics
6. MCA 305F: Entrepreneurship Development
7. MCA 305G: Software Testing

Semester –III

Subject Code	Title of the subject	L	T	P	C	QP
MCA20301	**Compiler Design**	3	0	0	3	A

UNIT:1 **10 Hours**

Introduction to Compilers: Compilers and translators, Phases of compiler design, cross compiler, Bootstrapping, Design of Lexical analyser, LEX programming.Syntax Analysis: Specification of syntax of programming languages using CFG, Topdown parser, design of LL (1) parser, bottom up parsing technique, LR parsing algorithm, Design of SLR, LALR, CLR parsers.YACC programming**.**

UNIT:2 **10 Hours**

Syntax directed translation: Study of syntax directed definitions & syntax directed translation schemes, implementation of SDTS, intermediate notations: postfix, syntax tree, TAC, translation of expression, controls structures, declarations, procedure calls, Array reference.
Storage allocation & Error Handling: Run time storage administration, stack allocation, symbol table management, Error detection and recovery: lexical, syntactic, semantic.

UNIT:3 **10 Hours**

Code optimization: Important code optimization techniques, loop optimization, control flow analysis, data flow analysis, Loop invariant computation, Induction variable removal, Elimination of Common sub expression.

UNIT:4 **10 Hours**

Code generation – Problems in code generation, Simple code generator, Register allocation and assignment, Code generation from DAG, Peephole optimization.

UNIT:5(As per choice of faculty) Graphics using C. **06 Hours**

(as per choice of faculty) Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures/Lecture by Industry Expert/MOOCS

Text Books

1. Compilers: Principles Techniques and Tools 1st edition by A. V. Aho, Sethi, Ullman, Pearson education.
2. Principal of Compiler Design – Alfred V. Aho& Jeffery D. Ullman ,Narosa Pub. House.
3. Principles of Compiler Design by Alfred V. Aho., Jeffrey D. Ulman.
4. “Compilers: Principles, Techniques and Tools” Aho, Ravi Sethi, Ullman, Pearson Education, VIII Ed. 2002.

Ref. Books

1.Lex and Yacc by Johan R. levine, Tonny Mason, et. al. O” Reilly and Assosiates.
2. “Compilers Design in C” Allen I. Holub, PHI eastern economy edition 2003.

Subject Code	Name of the Subject	L	T	P	C	QP
MCA20302	**Computer Networks**	**3**	**0**	**0**	**3**	A

UNIT:1 Introduction **(10 Hours)**

Network architecture – layers – Physical links – Channel access on links –Hybrid multiple accesstechniques - Issues in the data link layer - Framing –Error correction and detection – Link-level FlowControl

UNIT:2(10 Hours) Medium access – CSMA – Ethernet – Token ring – FDDI - Wireless LAN – Bridges andSwitches,Circuit switching vs. packet switching / Packet switched networks – IP – ARP – RARP – DHCP – ICMP –Queueing discipline – Routing algorithms – RIP – OSPF – Subnetting– CIDR – Interdomain routing – BGP – Ipv6 – Multicasting – Congestion avoidance in network layer

UNIT:3(10 Hours)

UDP – TCP – Adaptive Flow Control – Adaptive Retransmission -Congestion control – Congestionavoidance – QoS

UNIT:4 **(10 Hours)**

Email (SMTP, MIME, IMAP, POP3) – HTTP – DNS- SNMP – Telnet – FTP –Security – PGP - SSH.

UNIT:5 (as per choice of faculty) **(06 Hours)**

(as per choice of faculty) Portion covered can be tested through Internal evaluation only not to be included in University examination) Preferably use of NetSim, NS2

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books

1.Larry L. Peterson, Bruce S. Davie, “Computer Networks: A Systems Approach” ,Third Edition,Morgan Kauffmann Publishers Inc., 2003.

Reference Books:

1. JamesF.Kuross,KeithW.Ross,“ComputerNetworking,AtopDownApproachFeaturingtheInternet”,Third Edition,Addison Wesley, 2004.
2. NaderF.Mir,“Computer andCommunicationNetworks”,PearsonEducation,2007
3. Comer, “ComputerNetworksandInternetswithInternetApplications”,Fourth Edition, PearsonEducation,2003.
4. Andrew S.Tanenbaum,“Computer Networks”,FourthEdition,2003.
5. William Stallings, “Data and Computer Communication”, Sixth Edition, Pearson Education,2000

Subject Code	Name of the Subject	L	T	P	C	QP
MCA20303	**Python Programming**	**4**	**0**	**0**	**4**	A

UNIT:1 Introduction **(10 Hours)**

Introduction: Installation, First Python Program: Interactive Mode Programming, Script Mode Programming; Identifiers, Reserved Words, Lines and Indentation, Multi-Line Statements, Quotation & Comments; Assigning Values to Variables, Multiple Assignment.

UNIT:2(10 Hours) Standard Data Types: Numbers, Strings, Lists, Tuples, Dictionary; Data Type Conversion; Basic Operators: Arithmetic, Comparison, Assignment, Bitwise; Operators: Logical, Membership, Identity; Operators Precedence; Python Numbers & Mathematical functions.

Data Type Conversion: Basic Operators: Arithmetic, Comparison, Assignment, Bitwise; Basic Operators, Python Numbers & Mathematical functions; Python Strings.

UNIT:3(10 Hours)

Python statements and Loops: if, if-else, While, for loops, break, continue, pass, Python Function; Files I/O.

Functions: Definition, call, positional and keyword parameter. Default parameters, variable number of arguments. Modules - import mechanisms. Functional programming - map, filter, reduce, max, min. lambda function - list comprehension.

UNIT:4 **(10 Hours)**

Object Oriented Programming: classes and objects - Inheritance – Polymorphism overloading; Error handling & Exceptions - try, except and raise - exception propagation

File Processing: reading and writing files

UNIT:5 (as per choice of faculty) **(06 Hours)**

(as per choice of faculty)

Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books

1. *Stephen J. Chapman-'Electric Machinery and Fundamentals'- McGraw HillInternational Edition, (Fourth Edition), 2015.*
2. *M.G.Say-'Alternating Current Machines', English Language Book Society(ELBS)/ Longman, 5th Edition, Reprinted 1990.*

Reference Books:

1. Learning Python", Mark Lutz, O'Reilly Media, Inc., Fifth Edition, 2013.
2. "Introduction to Computer Science Using Python", Charles Dierbach, Wiley Publication, Second Edition, 2012.

Subject Code	Title of the subject	L	T	P	C	QP
MCA20304	**Software Engineering**	4	0	0	4	A

UNIT:1 **10 Hours**

Introduction to Software Engineering, Software Process, Perspective and Specialized Process Models: The Waterfall Model, Incremental Process Models, The RAD Model, Prototyping Model, V- Model, Spiral Model, Agile and Scrum Model. Software Project Management: Estimation, LoC and FP based Estimation, COCOMO Model, and Project Scheduling: Scheduling, Earned Value Analysis, and Risk Management.

UNIT:2 **10 Hours**

Software Requirements: Functional and Non-functional, User requirements, System requirements, Software Requirements Document, Requirement Engineering Process: Feasibility Studies, Requirement elicitation and analysis, Requirement validation, Requirements Management, Classical Analysis: structured system analysis, Petri Nets – Decision table, Decision tree, Documentation and Gunning's fog Index. CASE TOOL: Application on Documentation

UNIT:3 **10 Hours**

Design process: Design concepts, Design Model, Design Heuristic Architectural Design: Architectural Mapping using Data Flow Structure Chart design using DFD. Transform Analysis, Transaction Analysis: OOD Modeling Using UML, User Interface Design.
CASE TOOL: Application on Design

UNIT:4 **10 Hours**

Software implementation techniques: coding practices: Refactoring, Code Review, Code Inspection, Driver and Stub Module. Software Testing Fundamentals: Internal and external views of Testing- White Box Testing, Basis path testing, control structure testing. Black Box Testing: Regression Testing, Unit Testing, Integration Testing, Validation Testing, System Testing and Debugging. CASE TOOL: Test-Case Design, Software Reliability Measures and Growth Modeling: SEI CMM: Characteristics of Software Maintenance, Reverse Engineering, Re-engineering.).

UNIT:5(As per choice of faculty) Graphics using C. **06 Hours**

(as per choice of faculty) Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures/Lecture by Industry Expert/MOOCS

Text Books

1. Fundamentals of Software Engineering, Rajib Mall, PHI, 2014.
2. Software Engineering, APractitioner‟s Approach, Roger S. Pressman, TMG Hill.

Ref. Books

1. Software Engineering, I. Somerville, 9th Ed. , Pearson Education.

Subject Code	Name of the Subject	L	T	P	C	QP
MCA20305A	**Data Analytics**	**3**	**0**	**0**	**3**	**A**

UNIT:1 **(10 Hours)**

Predictive Analytics: Linear Methods for Regression and Classification: Overview of supervised learning, Linear regression models and least squares, Multiple regression, Multiple outputs, Subset selection, Ridge regression, Lasso regression , Linear Discriminate Analysis , Logistic regression, Perception learning algorithm.

UNIT:2 **(10 Hours)**

Neural Networks (NN), Support Vector Machines (SVM) and K-nearest Neighbor: Fitting neural networks, Back propagation, Issues in training NN, SVM for classification, Reproducing Kernels, SVM for regression, K-nearest–Neighbor classifiers (Image Scene Classification).

UNIT:3 **(10 Hours)**

Unsupervised Learning and Random forests: Association rules, Cluster analysis, Principal Components, Random forests and analysis.

Inferential Statistics and Prescriptive analytics Assessing Performance of a classification Algorithm (t-test, McNemar's test, Paired t-test, paired F-test), Analysis of Variance, Creating data for analytics through designed experiments. Introduction to big data and Challenges for big data analytics.

UNIT:4 **(10 Hours)**

Implementation of following methods using R or Mat lab (One of the class tests with a weight age of 15 marks be used to examine these implementations): Simple and multiple linear regression, Logistic regression, Linear discriminate analysis, Ridge regression, Cross-validation and boot strap, Fitting classification and regression trees, K-nearest neighbors, Principal component analysis, K-means clustering.

UNIT:5(As per choice of faculty). **06 Hours**

(as per choice of faculty) Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books:

1. *Trevor Hastie, Robert Tibshirani,Jerome Friedman , The Elements of Statistical Learning-Data Mining, Inference,andPrediction,Second Edition , Springer Verlag, 2009.*
2. *2. G.James,D.Witten,T.Hastie,R.Tibshirani-An introduction to statistical learning with applications in R,Springer,2013.*
2. *3 E.Alpaydin, Introduction to Machine Learning, Prentice Hall Of India,2010.*

Reference Books:

1. *C.M.Bishop –Pattern Recognition and Machine Learning,Springer,2006*
3. *L.Wasserman-All of statistics*

Lab Code	Name of the Lab	L	T	P	C	QP
MCA20306	**Computer Network lab**	**0**	**0**	**2**	**1**	

Objectives

1. PC-to-PCCOMMUNICATIONSUNDERWIN98/WIN2000'sDIRECTCABLE ONNECTION with NULL MODEM
   a)Using Serial PortsandRS-232CCable Connection
   b)UsingParallelPortsandDirectParallelCableConnection
2. PC-to-PC COMMUNICATIONS UNDER WIN98/WIN2000'sDIALUP NETWORKING withMODEMand4-LINEEXCHANGE
   a. PC-to-PCCOMMUNICATIONS
   b. UNDERWIN98/WIN2000'sHYPERTERMINALwithMODEMand4-LINEEXCHANGE
   c. LANWITHBUS/STAR(SwitchorHub)TOPOLOGYwithaminimumoftwo systems
      i)WindowsPeer-to-PeerNetwork
      ii)WindowsNTClient-ServerNetwork
   d. LANWITHBUS/STAR(SwitchorHub)TOPOLOGYwithaminimumoftwosystemsusing NOVELL Netware
2. TERMINALNETWORKWITHUNIX/LINUXSERVERandoneortwoTerminalsusingSerialPorts
   a. TERMINALNETWORKWITHUNIX/LINUXSERVER,8 portTerminalerverandoneortwoterminals

LIST OF EXPERIMENTS:

1. Implementation of Stop and Wait Protocol and Sliding Window Protocol.
2. Study of Socket Programming and Client – Server model
3. Write a code simulating ARP /RARP protocols.
4. Write a code simulating PING and TRACEROUTE commands
5. Create a socket for HTTP for web page upload and download.
6. Write a program to implement RPC (Remote Procedure Call)
7. Implementation of Subnetting.
8. Applications using TCP Sockets like

a. Echo client and echo server
b. Chat
c. File Transfer

9. Applications using TCP and UDP Sockets like

d. DNS
e. SNMP
f. File Transfer

10. Study of Network simulator (NS).and Simulation of Congestion Control Algorithms using NS/NetSim
11. Perform a case study about the different routing algorithms to select the network path with its optimum and economical during data transfer.

i. Link State routing
ii. Flooding
iii. Distance vector

Lab Code	Name of the Lab	L	T	P	C	QP
MCA20307	**Python Programming Lab**	**0**	**0**	**2**	**1**	

Exercise 1 - Basics

a) Running instructions in Interactive interpreter and a Python Script

b) Write a program to purposefully raise Indentation Error and Correct it

Exercise 2 - Operations

a) Write a program to compute distance between two points taking input from theuser (Pythagorean Theorem)

b) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.

Exercise - 3 Control Flow

a) Write a Program for checking whether the given number is a even number or not.

b) Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, . . . ,1/10

c) Write a program using a for loop that loops over a sequence. What is sequence?

d) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

Exercise 4 - Control Flow - Continued

a) Find the sum of all the primes below two million.

Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

b) By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

Exercise - 5 - DS

a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure

b) Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.

Exercise - 6 DS - Continued

a) Write a program combine_lists that combines these lists into a dictionary.

b) Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?

Exercise - 7 Files

a) Write a program to print each line of a file in reverse order.

b) Write a program to compute the number of characters, words and lines in a file.

Exercise - 8 Functions

a) Write a function ball_collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.

Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radiusIf

(distance between two balls centers) <= (sum of their radii) then (they are colliding)

b) Find mean, median, mode for the given set of numbers in a list.

Exercise - 9 Functions - Continued

a) Write a function nearly_equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.

b) Write a function dups to find all duplicates in the list.

c) Write a function unique to find all the unique elements of a list.

Exercise - 10 - Functions - Problem Solving

a) Write a function cumulative_product to compute cumulative product of a list of numbers.

b) Write a function reverse to reverse a list. Without using the reverse function.

c) Write function to compute GCD, LCM of two numbers. Each function shouldn't exceed one line.

Exercise 11 - Multi-D Lists

a) Write a program that defines a matrix and prints

b) Write a program to perform addition of two square matrices

c) Write a program to perform multiplication of two square matrices

Exercise - 12 - Modules

a) Install packages requests, flask and explore them using (pip)

b) Write a script that imports requests and fetch content from the page. Eg. (Wiki)

c) Write a simple script that serves a simple HTTP Response and a simple HTML Page

Exercise - 13 OOP

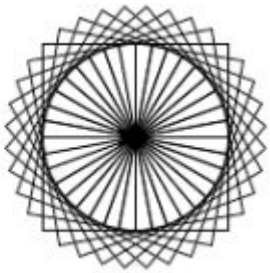
a) Class variables and instance variable

i) Robot

ii) ATM Machine

Exercise - 14 GUI, Graphics

1. Write a GUI for an Expression Calculator using tk.
2. Write a program to implement following figures using turtle



Exercise - 15 - Testing

a) Write a test-case to check the even numbers function even_numbers which return True on passing a list of all even numbers.

b) Write a test-case to check the function reverse_string which returns the reversed.

Exercise - 16 - Advanced

a) Build any one classical data structure.

b) Write a program to solve knapsack problem

Lab Code	Name of the Lab	L	T	P	C	QP
MCA20308	**Software Engineering Lab**	**0**	**0**	**2**	**1**	

List of Experiments:

Develop requirements specification for a given problem (The requirements specification Should include both functional and non-functional requirements. For a set of about 20sample problems, see the questions section of Chap 6 of Software Engineering book of Rajib Mall)

2: Develop DFD Model (Level 0, Level 1 DFD and data dictionary) of the sample problem (Use of a CASE tool required)

3: Develop structured design for the DFD model developed

4: Develop UML Use case model for a problem (Use of a CASE tool any of Rational rose, Argo UML, or Visual Paradigm etc. is required)

5: Develop Sequence Diagrams.

6: Develop Class diagrams.

7: Develop code for the developed class model using Java.

8: Use testing tool such as Junit.

9: Use a configuration management tool.

1. 10: Use any one project management tool such as Microsoft Project or Gantt Project, etc.

	Course Particulars		Credit Particulars				Evaluation							
			Hours Per Week				THEORY			PRACTICAL				
Category	CourseCode	CourseName	L	T	P	C	CIA	SEE	Total	CIA	SEE	Total	Total Marks	Credits
Professional Elective Courses	MCA20401	Professional Elective 3 ***	3			3	70	30	100	-	-	-	100	3
	MCA20402	Technical Seminar Result Writing			2	2	70	30	100	-	-	-	100	2
Professional Core Courses	MCA20403	Industrial Training cum Project/ Entrepreneurship Training cum Project			15	15	70	30	100	-	-	-	100	15
Total			**3**		**17**	**20**			**500**			**300**	**800**	**20**

*** **Professional Elective-III**
1. MCA 401A: Internet of Things
2. MCA 401B: Block Chain
3. MCA 401C: Soft Computing
4. MCA 401D: Foundations of Statistical Natural Processing (NLP)
5. MCA 504E: Software Testing
6. MCA 504F: Marketing Management
7. MCA 505G: Open Source Technology

Subject Code	Name of the Subject	L	T	P	C	QP
MCA20305A	**Internet of Things**	**3**	**0**	**0**	**3**	**A**

UNIT:1 **(8 Hours)**

Introduction to IoT: - Architectural Overview, Design principles and needed capabilities, IoT Applications, Sensing,Actuation, Basics of Networking, M2M and IoT Technology Fundamentals- Devices andgateways, Data management, Business processes in IoT, Everything as a Service(XaaS), Role ofCloud in IoT, Security aspects in IoT.

Domain-Specific IOTs: Home Automation, Cities, Environment, Energy,Retail, Logistics, Agriculture, Industry, Health & Life Style.

UNIT:2 **(9 Hours)**

Elements of IoT: - Hardware Components- Computing (Arduino, Raspberry Pi), Communication, Sensing, Actuation, I/O interfaces.

Software Components- Programming API's (using Python/Node.js/Arduino) for Communication Protocols-MQTT, ZigBee, Bluetooth, CoAP, UDP, TCP.

UNIT:3 **(18 Hours)**

Developing IoT solutions: Introduction to Python, Introduction to different IoT tools, Introduction to Arduino and Raspberry Pi Implementation of IoT with Arduino and Raspberry, Cloud Computing, Fog Computing, Connected Vehicles, Data Aggregation for the IoT in Smart Cities, Privacy and Security Issues in IoT.

UNIT:4 **(10 Hours)**

IoT Application Development: - Solution framework for IoT applications- Implementation of Device integration, Data acquisitionand integration, Device data storage- Unstructured data storage on cloud/local server,Authentication, authorization of devices.

IoT Case Studies: - IoT case studies and mini projects based on Industrial automation, Transportation, Agriculture,Healthcare, Home Automation

UNIT:5(As per choice of faculty). **06 Hours**

(as per choice of faculty) Portion covered can be tested through Internal evaluation only not to be included in University examination)

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books:

1. Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, StamatisKarnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", 1st Edition, Academic Press, 2014.
2. VijayMadisetti, ArshdeepBahga," Internet of ThingsA Hands- On-Approach",2014,ISBN:978 0996025515
3. Dr. SRN Reddy, RachitThukral and Manasi Mishra, "Introduction to Internet of Things: A practical Approach", ETI Labs
4. Pethuru Raj and Anupama C. Raman, "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press

3. Jeeva Jose, "Internet of Things", Khanna Publishing House, Delhi

Reference Books:

1. AdrianMcEwen, “Designing the Internet of Things”, Wiley Publishers, 2013, ISBN:978-1-118-43062-0
2. Daniel Kellmereit, “The Silent Intelligence: The Internet of Things”. 2013, ISBN:0989973700
3. Adrian McEwen, “Designing the Internet of Things”, Wiley
4. Raj Kamal, “Internet of Things: Architecture and Design”, McGraw Hill
5. CunoPfister, “Getting Started with the Internet of Things”, O Reilly Media