BENGALURU NORTH UNIVERSITY

REGULATIONS, SCHEME AND SYLLABUS

For the course

MASTER OF COMPUTER APPLICATIONS (MCA)

I to IV Semesters

(Choice Based Credit System –Y2K20 Scheme)

Revised w.e.f.

Academic Year 2020-21 and onwards

MCA PROGRAMME

BENGALURU NORTH UNIVERSITY,
Tamaka, Kolar

BENGALURU NORTH UNIVERSITY
Regulations of Master of Computer applications (MCA) Course

1 **TITLE OF THE COURSE:** The course shall be called MCA – Master of Computer Applications.

2 **DURATION OF THE COURSE**: The course of study shall be two years.

3 **ELIGIBILITY FOR ADMISSION**: A candidate with any degree of a minimum of 3 years duration (10+2+3) of Bangalore university or of any other University equivalent there in to with a minimum of 50% of marks in the aggregate of all subjects including languages, if any, provided further, that the candidate has studied Mathematics / Computer science /Business Mathematics / Statistics / Computer Applications / Electronics as a subject at PUC level or equivalent HSC (XII Standard) or at Degree level is eligible for admission to MCA Course. Relaxation to SC/ST, Group I be extended as per University norms.

4 **ATTENDANCE**: In each Semester a candidate should be considered to have successfully undergone the prescribed Course of study if the candidate has attended at least 75% of the classes in each subject (Theory, Lab & Practical).

5 **SCHEME OF EXAMINATION**:

A. The Internal Assessment marks should be decided for each of the theory subjects by conducting 2 tests, each of 60 minutes duration, spread over the span of a Semester. A seminar should also be given by the student in the second year and the same to be assessed and evaluated for internal assessment along with two tests.

B. The Internal Assessment marks in Practical course is based on the performance in the Laboratory. The Internal Assessment marks for Project work of a candidate is based on the dissertation and seminar.

6 **ELIGIBILITY TO GO TO THE HIGHER SEMESTER**:

A. A Candidate is allowed to carry over all the previous uncleared (failed) theory papers and Practicals to subsequent semesters from the first to fourth semester.

B. The maximum period for the completion of the course shall be four years from the date of admission.

7 MINIMUM FOR PASS AND DECLARATION OF RESULTS

A. For a pass in a semester, a candidate shall secure a minimum of 40% of the marks prescribed for a subject in the University Examination (Theory, Practical, Project work) and 50% of the marks in the aggregate inclusive of the Internal Assessment marks obtained in all subjects put together.

B. The candidates who do not satisfy 7(A) shall be deemed to have failed and have to take exams in the subjects in which he has secured less than 40% at the University examination.

C. Provision is made for rejection of results of all the subjects of a Semester only once, if the candidate decides to reappear for all the subjects of that semester. Such rejection should be made within 30 days of announcement of result, by making a written application, through the Head of the Institution. If such rejection is in respect of the results of all the subjects of one semester and earn fresh Internal marks as well.

D. The results of any semester will be declared as pass or fail as the case may be in accordance with regulation 7(A).

E. To be eligible for the award of the MCA degree, a candidate shall have completed the scheme of training and passed in all subjects prescribed for the Course.

F. Further to regulation 7(A), the classification followed by the University for all PG courses shall be made applicable for the declaration of results of each Semester.

8 CLASSIFICATION OF RESULT FOR THE MCA COURSE AND DECLARATION OF RANKS:

Further to regulations 7(A) and 7(F), the names of all successful candidates securing First Class with Distinction and First Class in the First attempt shall be arranged in the order of Merit and only first FIVE Ranks shall be declared.

9 A candidate shall complete examinations of all Semesters of MCA Course within - FOUR years from the date of admission

SCHEME OF STUDY AND EXAMINATION FOR MASTER OF COMPUTER APPLICATIONS (MCA)

Sem	Paper Code	Title of the paper	Hours / Week	Marks			Credits	
				IA	Exam	Total	Subject	Sem
I	1MCA1	The Art of Programming	4	30	70	100	4	28
	1MCA2	Discrete Mathematics	4	30	70	100	4	
	1MCA3	Computer Organization and Architecture	4	30	70	100	4	
	1MCA4	Theory of Computation	4	30	70	100	4	
	1MCA5	Object Oriented Programming	4	30	70	100	4	
	1MCA6	Data Structures	4	30	70	100	4	
	1MCA7	Data Structures Lab	8	30	70	100	2	
	1MCA8	Object Oriented Lab	8	30	70	100	2	
II	2MCA1	Operating Systems	4	30	70	100	4	28
	2MCA2	Database Management Systems	4	30	70	100	4	
	2MCA3	Computer Networks	4	30	70	100	4	
	2MCA4	Software Engineering	4	30	70	100	4	
	2MCA5	The Design and Analysis of Algorithm	4	30	70	100	4	
	2MCA6	Artificial Intelligence	4	30	70	100	4	
	2MCA7	Database Management System Lab	8	30	70	100	2	
	2MCA8	Unix Programming Lab	8	30	70	100	2	
III	3MCA1	Research Methodology	4	30	70	100	4	27
	3MCA2	Open Elective	4	30	70	100	3	
	3MCA3		4	30	70	100	4	
	3MCA4	V	4	30	70	100	4	
	3MCA5		4	30	70	100	4	
	3MCA6		4	30	70	100	4	
	3MCA7	MINI PROJECT	8	30	70	100	4	
IV		MAIN PROJECT					17	17

FIRST SEMESTER MCA

1MCA1: THE ART OF COMPUTER PROGRAMMING

Total Teaching Hours: 52 No. of Hours / Week: 04

UNIT - I [13 Hours]

Introduction: The Role of Algorithms in Computing, Algorithms as a technology, Analyzing algorithms, Designing algorithms, Growth of Functions, Asymptotic notation, Standard notations and common functions. Fundamental Algorithms: Exchanging the values of two variables, Counting, Summation of a set of numbers, Factorial Computation, Generating of the Fibonacci sequence, Reversing the digits of an integer, Base Conversion, Character to number conversion.

UNIT - II [13 Hours]

C Programming: Getting Started, Variables and Arithmetic expressions. Input and Output: Standard input and output, formatted output- printf, variable length argument list, formatted input-scanf. Control Flow: Statements and Blocks, If-else, else-if, switch, loops: while loop, for loop, do while, break and continue, goto and labels. Pointers and Arrays: pointers and address, pointers and function arguments, multidimensional array, initialization of pointer arrays, command line arguments.

UNIT - III [13 Hours]

Factoring Methods: Finding the square root of a number, the smallest Divisor of an integer, the greatest common divisor of two integers, generating prime numbers, computing the prime factors of an integer, generation of pseudo random numbers, raising a number to a large power, computing the nth Fibonacci Number. Array Techniques: Array order Reversal, Array counting or Histogramming, Finding the maximum number in a set, removal of duplicates from an ordered array, partitioning an array, Finding the kth smallest element, multiplication of two matrices.

UNIT - IV [13 Hours]

Merging: the two-way merge. Sorting: Sorting by selection, sorting by exchange, sorting by insertion, sorting by diminishing increment, sorting by partitioning. Searching: binary search, hash search. Text processing and Pattern searching: text line length adjustment, left and right justification of text, keyword searching in text, text line editing, linear pattern search, sublinear pattern search.

Text Books:

1. *R.G.Dromey, "How to Solve it by Computer", Pearson Education India, 2008.*
2. *Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, The MIT Press Cambridge, Massachusetts London, England, 2009.*
3. *Brain M. Kernighan, and Dennis M. Ritchie, "The C Programming Language", 2nd edition, Princeton Hall Software Series, 2012.*

Reference Books:

1. *Steven S. Skiena, "The Algorithm Design Module", 2nd Edition, Springer-Verlag London Limited, 2008.*
2. *Donald E. Knuth, The Art of Computer Programming", Volume 1: Fundamental Algorithms, 3rd Edition, Addison Wesley Longman, 1997.*

3. *Donald E. Knuth, The Art of Computer Programming", Volume 2: Seminumerical Algorithms, 3rd Edition, Addison Wesley Longman, 1998.*
4. *Greg Perry and Dean Miller, "C programming Absolute Beginner's Guide", 3rd edition, Pearson Education, Inc, 2014.*

Web Resources:

1. http://algorithmsforinterviews.com "Algorithms for Interviews"

1MCA2: DISCRETE MATHEMATICS

Total Teaching Hours: 52 No. of Hours / Week: 04

UNIT – I [13 Hours]

Set Theory and Logic: Fundamentals of Set theory, Set Operations and the Laws of Set Theory, Counting and Venn Diagrams, Cartesian Products and Relations, Functions–One-to-One, Onto Functions, Function Composition and Inverse Functions. Mathematical Induction, The well ordering principle, Recursive Definitions, Structural Induction, Recursive algorithms. Fundamentals of Logic, Propositional Logic, Logical Connectives and Truth Tables, Logic Equivalence, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs. Proof Methods and strategy.

UNIT - II [13 Hours]

Counting and Relations: Basics of counting, Pigeonhole Principle, Permutation and Combinations, Binomial coefficients. Recurrence relations, Modeling with recurrence relations with examples of Fibonacci numbers and the tower of Hanoi problem. Divide and Conquer relations with examples (no theorems). Definition and types of relations, Representing relations using matrices and digraphs, Partial Orderings, Hasse diagrams, Maximal and Minimal elements, Lattices.

UNIT - III [13 Hours]

Probability: The Concept of Probability-Sample Spaces, Probability as Relative Frequency, Axiomatic Definition of Probability, Properties of Probability, Additive Property, Conditional Probability, Multiplicative Law of Probability, Law of Total Probability, Bayes' Formula, Independent Events. Random Variables, Distribution Functions, Discrete Random Variables, Continuous Random Variables, Probability Mass Function and Probability Density Function, Expectation and Variance, Functions of Random Variables, Some important Probability Distributions: Discrete - Bernoulli Trials and Binomial distribution, Geometric distribution and Poisson distribution, Continuous - Uniform distribution, Normal distribution and Exponential distribution.

UNIT - IV [13 Hours]

Graph Theory: Graphs: Introduction, Representing Graphs, Graph Isomorphism, Operations on graphs. Trees: Introduction, Applications of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees, Prim's and Kruskul's Algorithms. Connectivity, Euler and Hamilton Paths, Planar Graphs. Directed graphs: Fundamentals of Digraphs, Computer Recognition - Zero-One Matrices and Directed Graphs, Out-degree, in-degree, connectivity, orientation, Eulerian and Hamilton directed graphs, tournaments.

Text Books:

1. *Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, 5thEdition, Pearson Education, 2004.*
2. *C. L. Liu: Elements of Discrete Mathematics, Tata McGraw-Hill, 2000.*
3. *Sheldon M Ross: Introduction to Probability Models, 12th edition, Academic Press, 2019.*
4. *F. Harary: Graph Theory, Addition Wesley, 1969.*

Reference Books:

1. *Kenneth H Rosen: "Discrete Mathematics and its Applications", McGraw Hill publications, 7th edition, 2007.*
2. *J. P. Tremblay and R.P. Manohar: Discrete Mathematical Structures with applications to Computer Science, Mc Graw Hill Ed. Inc. 1975.*
3. *Sheldon M Ross: Introduction to Probability and Statistics for Engineers and Scientists, 6th edition, Academic Press, 2020.*
4. *Michael Baron: Probability and Statistics for Computer Scientists, 3rd Edition, CRC, 2019*

Web Resources:

1. *https://www.my-mooc.com/en/categorie/mathematics*
2. *http://www.nptelvideos.in/2012/11/discrete-mathematical-structures.html*
3. *https://ocw.mit.edu/courses/mathematics/*

1MCA3: COMPUTER ORGANIZATION AND ARCHITECTURE

Total Teaching Hours: 52 No. of Hours / Week: 04

UNIT - I [13Hours]

Number Systems: Binary, Octal, Hexa decimal numbers, base conversion, addition, subtraction of binary numbers, one's and two's complements, positive and negative numbers, character codes ASCII, EBCDIC etc. Computer Arithmetic: Addition and Subtraction, Multiplication and Division algorithms, Floating-point Arithmetic Operations, Decimal arithmetic operations. Structure of Computers: Computer types, Functional units, Basic operational concepts, Von-Neumann Architecture, Bus Structures, Software, Performance, Multiprocessors and Multicomputer, Digital Logic Circuits: Logic gates, Boolean algebra, Map Simplification. Combinational Circuits: Half Adder, Full Adder, flip flops.Sequentialcircuits:Shiftregisters,Counters,IntegratedCircuits,Mux,Demux,Encoder,Decoder.Data representation: Fixed and Floating point, Error detection and correction codes.

UNIT - II [13Hours]

Basic Computer Organization and Design: Instruction codes, Computer Registers, Computer Instructions and Instruction cycle. Timing and Control, Memory-Reference Instructions, Input-Output and interrupt. Central processing unit: Stack organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Complex Instruction Set Computer (CISC) Reduced Instruction Set Computer (RISC), CISC vs RISC

UNIT - III [13Hours]

Register Transfer and Micro-operations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro-Operations, Logic Micro-Operations, Shift Micro-Operations, Arithmetic logic shift unit. Micro-programmed Control: Control Memory, Address Sequencing, Micro-Program example, Design of Control Unit. Input Output: I/O interface, Programmed IO, Memory Mapped IO, Interrupt Driven IO, DMA. Instruction level parallelism: Instruction level parallelism (ILP)-over coming data hazards, limitations of ILP

UNIT - IV [13Hours]

Memory System: Memory Hierarchy, Semiconductor Memories, RAM(Random Access Memory), Read Only Memory (ROM), Types of ROM, Cache Memory, Performance considerations, Virtual memory, Paging, Secondary Storage, RAID. Multiprocessors And Thread level Parallelism: Characteristics of multiprocessors, Multi-Threaded Architecture,Distributed Memory MIMD Architectures, Architecture of MultiThreaded Processor, principle of MultiThreading, Interconnection structures, Inter Processor Arbitration, Inter processor Communication and Synchronization, Cache Coherence.

TEXT BOOKS:

1. *Mano M Morris, "Computer System Architecture", 3rd edition Pearson India(2019).*
2. *William Stallings, "Computer Organization and Architecture designing for performance", 10th edition, Pearson(2016)*

REFERENCE BOOKS:

1. *Subrata Ghoshal, "Computer Architecture And Organization", Pearson India(2011).*
2. *Andrew S. Tanenbaum " Structured Computer Organization", 5th edition, Pearson Education Inc(2006).*
3. *Carl Hamacher, Zvonks Vranesic,SafeaZaky, "Computer Architecture And Organization", 5th edition McGraw Hill New Delhi,India(2002).*
4. *Kai Hwang, "Advanced Computer Architecture - Parallelism, Scalability, Programmability", Tata Mcgraw-Hill (2008).*

1MCA4: THEORY OF COMPUTATION

Total Teaching Hours: 52 No. of Hours / Week: 04

UNIT - I [13 Hours]

Introduction to Automata: The Principle of Mathematical Induction, Introduction to formal proof, Additional forms of Proof, Inductive Proofs. Finite Automata: Introduction, Deterministic Finite Automata (DFA) : Formal definition, simpler notations (state transition diagram, transition table), language of a DFA. Nondeterministic Finite Automata (NFA): Definition of NFA, language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata.

UNIT - II [13 Hours]

Regular Expressions: Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA's to Regular Expressions, Converting Regular Expressions to Automata, applications of Regular Expressions. Regular Grammars: Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular. Properties of Regular Languages: The Pumping Lemma for regular languages, Applications of the pumping lemma closure properties of regular languages, Decision properties of regular languages, Equivalence and minimization of automata. Context Free Grammer (CFG): Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFG's, Minimization of CFG's, CNF, GNF, Pumping Lemma for CFL's, Enumeration of Properties of CFL.

UNIT - III [13 Hours]

Pushdown Automata Introduction: Definition, Formal definition of pushdown automata, A graphical notation for PDA's, Instantaneous descriptions of a PDA. Pushdown Automata: Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack and its Equivalence, Equivalence of CFG and PDA. Deterministic Pushdown Automata: Definition of a deterministic PDA, Regular languages and deterministic PDA's, DPDA's and context-free languages, DPDA's and ambiguous grammars. Transducers: Moore machine, Mealy machine, Difference between Moore & Mealy machines, Properties, Equivalence of Moore & Mealy machines. Context Sensitive Languages: Linear bounded automata, Chomsky's hierarchy of languages.

UNIT - IV [13 Hours]

Introduction to Turing Machines: The Turing Machine: The instantaneous descriptions for Turing machines, Transition diagrams for Turing machines, The language of a Turing machine, Turing machines and halting programming techniques for Turing machines, Extensions to the basic Turing machine, Restricted Turing machines, Turing machines and computers. Undecidability: A language that is not recursively enumerable, Enumerating the binary strings, Codes for Turing machines, the diagonalization language, An undecidable problem that is RE: Recursive languages, Complements of recursive and RE languages, The universal languages, Undecidability of the universal language. Undecidable Problems About Turing Machines: Reductions, Turing machines that accept the empty language. Post's correspondence problem: Definition of post's correspondence problem, The "Modified" PCP, Other undecidable problems: Undecidability of ambiguity for CFG's. Unsolvable Problems and Computable Functions: A no recursive Language and Unsolvable Problem, Reducing one problem to another: The Halting Problem, Other unsolvable Problems involving TMs, Rice's Theorem and More Unsolvable problems.

Text Books:

1. *John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2007), Introduction to Automata Theory, Languages and Computation, 3rd Edition, Pearson Education, India.*
2. *K. L. P Mishra, N. Chandrashekaran (2003), Theory of Computer Science-Automata Languages and Computation, 2nd Edition, Prentice Hall of India, India.*

Reference Books:

1. *Harry. R. Lewis and C. H. Papadimitriou - Elements of the Theory of Computation, Second Edition, PHI, 2003.*
2. *John C. Martin - Introduction to Languages and the Theory of Computation, Fourth Edition, TMH, 2011.*
3. *Micheal Sipser - Introduction of the Theory and Computation, Thomson Brokecole, Second Edition, 1997.*
4. *C. K. Nagpal - Formal Languages and Automata Theory, Oxford Higher Education, April 2011.*

Web Resources:

1. *Youtube Channel: nptelhrd, Playlist name: Theory of automata, formal languages and computation.*

1MCA5: OBJECT ORIENTED PROGRAMMING

Total Teaching Hours: 52 No. of Hours / Week: 04

UNIT-I [13 Hours]

Evolution of Higher Level Programming Languages, Complexity of Software and their Attributes Object Orientated Programming Paradigm, Principles of Object Oriented Programming – Data Encapsulation, Polymorphism and Inheritance, Advantages of Object Oriented Programming, Application areas of Object Oriented Programming, Object Oriented Programming Languages, Limitations of Object Oriented Programming.

Introduction to the JAVA language - Evolution of Java, The Salient Features of the JAVA Language, The Java Byte Code and the JVM, The JAVA platform, JAVA, Internet and WWW, JAVA Environment, Tokens, Keywords and Identifiers, Constants and Variables, Data types, Console I/O, Structure of a Java Program, Executing a Java program, Types of Errors, Operators in JAVA, Precedence and Associativity of Operators, Type Conversion,

Selection Structures- Simple-If statement, If-Else statement, Nested If-Else statement, Else-if ladder, The Switch Statement, Looping Structures- The while loop, The For loop, The Do-While loop, Nested Loops, The Break Statement, The Continue Statement, Labelled Loops.

UNIT-II [13 Hours]

Classes, Objects and Methods - Class Definition; Instance Variables and Member Methods, Declaration and Creation of Objects, Accessing Members, Classification of Member Methods, Constructors, Copy Constructor, this keyword , Objects as Arguments to methods, Methods returning an object, Static member data, Static member methods, Static Blocks, Nesting of Methods, Recursion, Nested Classes, Inner Classes, Static nested Classes, Local Classes, Anonymous Classes, Final members, Variable Arguments, Objects of one class as members of another class (Containment), Finalize Method and Garbage Collection.

Inheritance – Single-level Inheritance, Multilevel Inheritance, Hierarchical Inheritance, Multiple Inheritance, Hybrid Inheritance, Constructors and Inheritance, Abstract Classes and Methods, Dynamic Method Dispatch, Object Slicing, Object Typecasting, Final Classes

Interfaces - Defining Interfaces, Implementing Interfaces, Polymorphism through interfaces, Implementing an Interface Partially, Extending Interfaces, Implementing Multiple Interfaces, Multiple Inheritance through Interfaces

UNIT-III [13 Hours]

Packages - Built-in Packages, User-defined Packages, Creating and using a Package, To create a package spread across multiple files, Importing classes from a package, Nested packages, Extending an imported class, Classes and interfaces in a package and using them, Static importing, Access Control. Arrays – One Dimensional Arrays, Two Dimensional Arrays, Three Dimensional Arrays, Arrays and methods, Arrays within classes, Array of objects, Strings – The String Class, The StringBuffer Class, Exception Handling- Types of Exceptions, Default Exception Handling Mechanism, User-Defined Exception Handling Mechanism, Try blocks, Catch Blocks, Nested Try Blocks, Stack Unwinding, Throw Statement, Throws Statement, Finally Statement

UNIT-IV [13 Hours]

Multithreaded Programming - The Java Thread Model, The Life Cycle of a Thread, The Thread Class, The Main Thread, Creating our own Threads -Extending the Thread Class, Implementing the Runnable Interface, Thread Groups, Thread Priorities, Synchronization, Deadlock, Suspending and Resuming Threads, Producer-Consumer Relationship between Threads, Daemon Threads. File Handling - The File Class, The Writer class and its subclasses, The Reader Class and its

subclasses, The OutputStream and its subclasses, The InputStream Class and its subclasses, The DataOutputStream Class, The DataInputStream Class, The ObjectOutputStream Class, The ObjectInputStream Class, Random Access files
Applets - Types of Applets, The Life Cycle of an Applet, Creating and Executing Applets, The Attributes in the <applet> tag, Passing Parameters to Applets, More about the <applet> tag, Working with Graphics

Textbooks:

1. E. Balagurusamy, Programming with JAVA, McGraw Hill, New Delhi, 2007

Reference Books:

1. Raj Kumar Buyya, Object Oriented Programming with JAVA, McGraw Hill, 2009
2. Herbert Schildt, Java A Beginner's Guide – Create, Compile, and Run Java Programs Today, Sixth Edition, Oracle Press, 2014
3. Ken Arnold, James Gosling, "The Java Programming Language, Fourth Edition, Addison Wisely, 2005
4. Herbert Schildt, 'The Complete Reference Java, 7th Edition, McGraw Hill, 2007

Web Resources

1. https://docs.oracle.com/javase/tutorial/
2. https://javabeginnerstutorial.com/core-java-tutorial/

1MCA6: DATA STRUCTURES

Total Teaching Hours: 52 No. of Hours / Week: 04

UNIT-I [13Hours]

Introduction and Overview: Definition, Elementary data organization, Data Structures, data Structures operations, Abstract data types, algorithms complexity, time-space trade off. Preliminaries: Mathematical notations and functions, Algorithmic notations, control structures, Complexity of algorithms, asymptotic notations for complexity of algorithms. String Processing: Definition, Storing Strings, String as ADT, String operations, word/text processing, Pattern Matching algorithms.

UNIT-II [13Hours]

Arrays: Definition, Linear arrays, arrays as ADT, Representation of Linear Arrays in Memory, Traversing Linear arrays, Inserting and deleting, Multi-dimensional arrays, Matrices and Sparse matrices. Linked list: Definition, Representation of Singly Linked List in memory, Traversing a Singly linked list, Searching in a Singly linked list, Memory allocation, Garbage collection, Insertion into a singly linked list, Deletion from a singly linked list; Doubly linked list, Header linked list, Circular linked list. Stacks: Definition, Array representation of stacks, Linked representation of stacks, Stack as ADT, Arithmetic Expressions: Polish Notation, Conversion of infix expression to postfix expression, Evaluation of Post fix expression, Application of Stacks, Recursion, Towers of Hanoi, Implementation of recursive procedures by stack. Queues: Definition, Array representation of queue, Linked list representation of queues. Types of queue: Simple queue, Circular queue, Double-ended queue, Priority queue, Operations on Queues, Applications of queues.

UNIT-III [13Hours]

Binary Trees: Definitions, Tree Search, Traversal of Binary Tree, Tree Sort, Building a Binary Search Tree, Height Balance: AVL Trees, Contiguous Representation of Binary Trees: Heaps, Lexicographic Search Trees: Tries, External Searching: B-Trees, Applications of Trees. Graphs: Mathematical Back ground, Computer Representation, Graph Traversal, Topological Sorting, Greedy Algorithm, Graphs as Data Structure.

UNIT-IV [13Hours]

Searching: Introduction and Notation, Sequential Search, Binary Search, Comparison of Methods. Sorting: Introduction and Notation, Insertion Sort, Selection Sort, Shell Sort, Divide And Conquer, Merge sort for Linked List, Quick sort for Contiguous List. Hashing: Sparse Tables, Choosing a Hash function, Collision Resolution with Open Addressing, Collision Resolution by Chaining.

Text Books:

1. Seymour Lipschutz, "Data Structures with C", Schaum's outLines, Tata Mc Graw Hill, 2011.

2. Robert Kruse, C.L.Tondo, Bruce Leung,Shashi Mogalla,"Data Structures and Program Design using C", Pearson Education, 2009.

ReferenceBooks:

1. Mark Allen Weiss," Data Structures and Algorithm Analysis in C", Second Edition, Pearson Education,2013.

2. *Forouzan,"A Structured Programming Approach using C",2nd Edition, Cengage LearningIndia,2008.*

1MCA7: DATA STRUCTURES LAB PROGRAMS

* For all the programs write the output, flowchart and number of basic operations performed.

1. Given {4,7,3,2,1,7,9,0, find the location of 7 using Binary search and also display its first occurrence.
2. Given {5,3,1,6,0,2,4} order the numbers in ascending order using Quick Sort.
3. Perform the Merge sort on the input {75,8,1,16,48,3,7,0} and display the output in descending order.
4. Write a program to insert the elements 61,16,8,27 into singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
5. Write a program to add $6x^3+10x^2+0x+5$ and $4x^2+2x+1$ using linked list.
6. Write a program to push 5,9,34,17,32 into stack and pop 3 times from the stack, also display the popped numbers.
7. Write a recursive program to find GCD of 4,6,8.
8. Write a program to inert the elements {5,7,0,6,3,9} into circular queue and delete 6,9&5 from it(using linked list implementation).
9. Given S1={“Flowers”} ; S2={“are beautiful”} ,
   a) Find the length of S1.
   b) Concatenate S1 and S2.
   c) Extract the substring “low” from S1.
   d) Find “are” in S2 and replace it with “is”.
10. Write a program to convert an infix expression x^y/(5*z)+2 to its postfix expression.
11. Write a program to evaluate a postfix expression 5 3+8 2 - *.
12. Write a program to create a binary tree with the elements 18,15,40,50,30,17,41 after creation insert 45 and 19 into tree and delete 15,17 and 41 from tree. Display the tree on each insertion and deletion operation.
13. Write a program to create binary search tree with the elements {2,5,1,3,9,0,6} and perform inorder, preorder and post order traversal.
14. Write a program to Sort the following elements using heap sort {9.16,32,8,4,1,5,8,0}.

1MCA8: OBJECT ORIENTED PROGRAMMING WITH JAVA LAB

PART-A

1. Develop a JAVA program to demonstrate the precedence and associativity among arithmetic operators. The program should also demonstrate how the default precedence can be overridden.
2. Write a JAVA program to validate a date. The program should accept day, month and year and it should report whether they form a valid date or not.
3. Write a JAVA program to display the following pattern.

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

4. Write a JAVA program to print the first n members of Fibonacci series.
5. Write a program to generate the multiplication tables of a range of numbers between m and n inclusive and m < n.
6. Write a JAVA program to define a class, define instance methods for setting and retrieving values of instance variables and instantiate its object.
7. Write a JAVA program to demonstrate static member data and static member methods
8. Write a JAVA Program to demonstrate nested classes
9. Write a JAVA program to demonstrate dynamic method dispatch.
10. Write a JAVA program to implement inheritance and demonstrate use of method overriding.

PART-B

11. Write a JAVA program to implement the concept of importing classes from user defined package and creating packages.
12. Write a program to demonstrate abstract class and abstract methods
13. Write a JAVA Program to implement an array of objects of a class.
14. Write a JAVA program to demonstrate String class and its methods.
15. Write a JAVA program to implement the concept of exception handling by creating user defined exceptions.
16. Write a JAVA program using synchronized threads, which demonstrates producer consumer concept.
17. Write a JAVA program that creates three threads. First thread displays “Good Morning” every one second, second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.
18. Write a JAVA program which uses FileInputStream / FileOutPutStream Classes.
19. Write a JAVA program to list all the files in a directory including the files present in all its subdirectories.
20. Write a JAVA program to demonstrate the life cycle of applet.

SECOND SEMESTER MCA

2MCA1: OPERATING SYSTEMS

Total Teaching Hours: 52 No. of Hours / Week: 04

UNIT - I [13 Hours]

Introduction: Computer System Organization, Architecture, Structure, Operations, Process Management, Memory Management, Storage Management, Kernel Data Structures, Computing Environments. Operating System Structures: Services, System Calls, Types, Operating System Structure, System Boot. Processes: Process Concept, Scheduling, Operations, Interprocess Communication. Multithreaded Programming: Multicore Programming, Multithreading Models.

UNIT –II **[**13 Hours]

Process Synchronization: The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization Examples. Process Scheduling: Criteria, Scheduling Algorithms, Multi-Processor Scheduling, Real-time CPU Scheduling. Deadlocks: System model, Characterization, Methods for handling deadlocks, Deadlock Prevention, Avoidance, Detection and Recovery from deadlock.

UNIT – III [13 Hours]

Memory Management Strategies: Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table. Virtual Memory Management: Demand Paging; Copy-on-Write, Page Replacement; Allocation of Frames; Thrashing, Memory-Mapped Files, Allocating Kernel Memory. File System: File Concept, Access Methods, Directory and Disk Structure, Protection. File-System Implementation: Structure, File-System and Directory Implementation, Allocation Methods, Free Space Management, Efficiency and Performance, Recovery. Mass-Storage Structure: Overview, Disk Scheduling, Disk Management.

UNIT - IV [13 Hours]

Protection: Goals, Principles, Domain of Protection, Access Matrix, Implementation of the Access Matrix, Access Control, Revocation of the Access Rights. Virtual Machines: Building Blocks, Types of VMs and their implementations. Distributed Systems: Advantages, Types of Network-based OS, Robustness, Design Issues, Distributed File Systems. Case Studies: The Linux System, Windows 10.

Text Books:

1. *Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating Systems Concepts, 9th Edition, 2016 India, Wiley.*

Reference Books:

1. William Stallings, "Operating Systems-Internals and Design Principles", Pearson, 9th Edition, 2018

2. D M Dhamdhere: Operating Systems – A Concept Based Approach, 3rd Edition, Tata McGraw – Hill, 2015.

3..Harvey M Deitel, Paul J Deitel, Dr Choffnes,"Operating Systems", Pearson Education Limited (Publisher), 3rd Edition, 2013.

4. J. Archer Harris, John Cordani, " Operating Systems", Schaum's Outline, Indian Edition, Mc Graw Hill Education (India), First Edition.

5. Gary Nutt, Nabendu Chaki, Sarmistha Neog, "Operating Systems" Pearson Education Limited, 3rd Edition, 2016.

2MCA2: DATABASE MANAGEMENT SYSTEMS

Total Teaching Hours: 52 No. of Hours / Week: 04

UNIT - 1 [10 Hours]

Databases and Database Users: Introduction, An example, Characteristics of the Database Approach, Actors on the Scene, Workers behind the Scene, Advantages of Using DBMS Approach, A Brief History of Database Applications, When Not to Use a DBMS. Database System Concepts and Architecture: Data Models, Schemas, and Instances, Three-schema Architecture and Data Independence, Database Languages and Interfaces, The Database System Environment, Centralized and Client-Server Architectures, Classification of Database Management Systems.

UNIT - 2 [13 Hours]

Data Modeling Using Entity-Relationship Model: Using High-Level Conceptual Data Models for Database Design, An Example Database Application, Entity Types, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, Refining the ER Design Company Database Diagrams, Naming Conventions and Design. Issues, File organization and storage, secondary storage devices, operations in file, heap files and sorted files, hashing techniques, type of single level ordered index, multi-level indexes, indexes on multiple keys, other types of indexes.

UNIT – 3 [16 Hours]

Relational Model and Relational Algebra: Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update Operations, Transactions and Dealing with Constraint Violations, Unary Relational Operations: SELECT and PROJECT, Relational Algebra Operations from SET Theory, Binary Relational Operations: JOIN and DIVISION, Additional Relational Operations, Examples of Queries in Relational Algebra. Relational Database Design: Anomalies in a database, functional dependency, normal forms, lossless join and dependency, BCNF, normalization through synthesis, higher order normal forms. SQL- SQL Data Definition and Data Types, Specifying Constraints in SQL, Schema Change Statements in SQL, Basic Queries in SQL, More Complex SQL Queries, Insert, Delete and Update Statements in SQL, Specifying Constraints as Assertion and Trigger, Views(Virtual Tables) in SQL, Embedded SQL, Dynamic SQL,

UNIT – 4 [13 Hours]

Introduction to transaction processing, transaction and system concepts, desirable properties of transactions, transaction support in SQL. Concurrency control techniques: two-phase locking techniques, concurrency control based on timestamp ordering, multi-version concurrency control techniques, validation concurrency control techniques. Recovery techniques: recovery concepts, recovery in multi-database systems, database backup and recovery from catastrophic failures.

Text Books:

1. *Elmasri and Navathe: Fundamentals of Database Systems, 7th Edition, Addison -Wesley, 2016.*
2. *Silberschatz, Korth and Sudharshan Data base System Concepts, 7th Edition, Tata McGraw Hill, 2019.*

References:

1. *C.J. Date, A. Kannan, S. Swamynatham: An Introduction to Database Systems, 8th Edition, Pearson education, 2009*
2. *Database Management Systems :Raghu Ramakrishnan and Johannes Gehrke: , 3rd Edition, McGraw-Hill, 2003*

2MCA3: COMPUTER NETWORKS

Total Teaching Hours: 52 No. of Hours / Week: 04

UNIT - I [13 Hours]

Introduction: Data Communications, Networks, Network Types, Internet History, Network Models: Protocol Layering, The OSI Model, TCP/IP Protocol Suite, Introduction to Physical Layer: Transmission Impairments, Data Rate Limits, Performance, Introduction to Data-Link-Layer: Link-Layer Addressing, Error Detection and Correction: Block Coding, Cyclic Codes, Checksum

UNIT - II [13 Hours]

Data Link Control: Data-Link Layer Protocols, HDLC, Point-To-Point (PPP), Media Access Control (MAC): ALOHA, CSMA, CSMA/CD, CSMA/CA, Reservation, Polling, Token Passing, FDMA, TDMA, CDMA

UNIT - III [13 Hours]

Introduction to Network Layer: Network-Layer Services, Packet Switching, Network-Layer Performance, IPV4 Addresses, Network Layer Protocols: Internet Protocol (IP), ICMPv4, Mobile IP, Unicast Routing: Routing Algorithms, Unicast Routing Protocols, Next Generation IP: IPv6 Addressing, The IPv6 Protocol.

UNIT - IV [13 Hours]

Introduction to Transport Layer: Introduction, Transport-Layer Protocols, Transport-Layer Protocols: User Datagram Protocol, Transmission Control Protocol: TCP Services, TCP Features, Segment, A TCP Connection, TCP Congestion Control, Flow Control, Error Control, Application Layer: WWW, E-MAIL, Domain Name System (DNS), Quality of Service: Flow Control To Improves QoS, Integrated Services, Cryptography and Network Security: Introduction, Confidentiality, Other Aspects of Security.

Text Books:

1. Behrouz A. Forouzan, "Data Communications and Networking", 5th Edition, McGraw Hill Education, 2013.

Reference Books:

1. Andrew S. Tanenbaum, David J. Wetherall, "Computer Networks", 5th Edition, Prentice Hall, 2011.

2. Larry L. Peterson and Bruce S. Davie, "Computer Networks A System Approach", 5th Edition, MKP, 2012.

3. James F. Kurose , Keith W. Ross, " Computer Networking, A Top-Down Approach", 5th Edition, Pearson, 2012.

Web Resources:

1. *https://www.geeksforgeeks.org/computer-network-tutorials/*
2. *https://codescracker.com/networking/*
3. *https://youtube.com/playlist?list=PLxCzCOWd7aiGFBD2-2joCpWOLUrDLvVV_*

2MCA4: SOFTWARE ENGINEERING

Total Teaching Hours: 53 No. of Hours / Week: 04

UNIT - I [13 Hours]

Overview, Objectives ,Three Perspectives on Software Engineering , The Agile Manifesto , Individuals and Interactions over Processes and Tools, Working Software over Comprehensive Documentation, Customer Collaboration over Contract, Negotiation, Responding to Change over Following a Plan, Application of Agile Software Development , Data About Agile Software Development, Agile Software Development in Learning Environments University Course Structure, Teaching and Learning Principles, The Studio Environment, The Academic Coach Role ,Overview of the Studio Meetings. Teamwork: Overview, Objectives, A Role Scheme in Agile Teams, Remarks on the Implementation of the Role Scheme, Human Perspective on the Role Scheme, Using the Role Scheme to Scale Agile Projects, Dilemmas in Teamwork, Teamwork in Learning Environments, Teaching and Learning Principles, Role Activities, Student Evaluation. Customers and Users: Overview, Objectives, The Customer, Customer Role, Customer Collaboration, The User, Combining UCD with Agile Development, Customers and Users in Learning Environments, Teaching and Learning Principles, Customer Stories.

SOFTWARE DESIGN:

- Design Diagrams: Use Case Diagrams - Class Diagrams - Interaction Diagrams - State chart Diagrams - Activity Diagrams
- Design Process- Design concepts : Abstraction, Architecture, patterns, Separation of Concerns, Modularity, Information Hiding, Functional Independence, Refinement, Aspects, Refactoring.
- Object Oriented Design Concepts, Design Classes- Design Model: Data, Architectural, Interface, Component, Deployment Level Design Elements ,
- Code review Analysis.

UNIT - II [13 Hours]

Time:

Overview, Objectives, Time-Related Problems in Software Projects, List of Time-Related Problems of Software Projects. the Time Perspective, Tightness of Software Development Methods, Sustainable Pace, Time Management of Agile Projects, Time Measurements, Prioritizing Development Tasks, Time in Learning Environments, The Planning Activity, Teaching and Learning Principles, Students' Reflections on Time-Related Issues, The Academic Coach's Perspective. Measures: Overview, Objectives, Why Are Measures Needed, Who Decides What Is Measured? What Should Be Measured, When Are Measures Taken? How Are Measures Taken? Who Takes the Measures? How Are Measures Used? Case Study, Monitoring a Large-Scale Project by Measures, Measure Definition, Measure Illustration, Measures in Learning Environments, Teaching and Learning Principles, Measurement Activities.

Quality:

Overview, Objectives, The Agile Approach to Quality Assurance, Process Quality, Product Quality, Test-Driven Development, How Does TDD Help to Overcome Some of the Problems Inherent in Testing, Learning: Overview, Objectives, Study Questions, How Does Agile Software Development Support Learning Processes.

UNIT - III [13 Hours]

Quality- Continued

Agile Software Development from the Constructivist Perspective, The Role of Short Releases and Iterations in Learning Processes, Learning in Learning Environments, Gradual Learning Process

of Agile Software Engineering, Learning and Teaching Principle, The Studio Meeting, End of the First Iteration, Intermediate Course Review and Reflection, Abstraction: Overview, Objectives, Study Questions, Abstraction Levels in Agile Software Development, Roles in Agile Teams.

Planning:
The Stand-Up Meeting, Design and Refactoring, Abstraction in Learning Environments, Teaching and Learning Principles. Trust: Overview, Objectives, Software Intangibility and Process Transparency, Game Theory Perspective in Software Development, Ethics in Agile Teams, Diversity, Trust in Learning Environments, Teaching and Learning Principle. Globalization: Overview, Objectives, Study Questions, The Agile Approach in Global Software Development, Communication in Distributed Agile Teams, Planning in Distributed Agile Projects, Case Study, Tracking Agile Distributed Projects, Reflective Processes in Agile Distributed Teams, Organizational Culture and Agile Distributed Teams, Application of Agile Principles in Non-Software Projects.

UNIT – IV [13 Hours]

Overview, Objectives, Case Study, Reflection on Learning in Agile Software Development, Reflective Practitioner Perspective, Retrospective, The Retrospective Facilitator, Case Study, Guidelines for a Retrospective Session, Application of Agile Practices in Retrospective Sessions, End of the Release Retrospective, Reflection in Learning Environments. Change: Overview, Objectives, A Conceptual Framework for Change Introduction, Changes in Software Requirements, Organizational Changes, Transition to an Agile Software Development Environment. Leadership: Overview, Objectives, Leaders, Leadership Styles, Case Study, The Agile Change Leader, Coaches, Leadership in Learning Environments, Teaching and Learning Principles. Delivery and Cyclicality: Overview, Objectives, Delivery, Towards the End of the Release, Release Celebration, Reflective Session Between Releases, Cyclicality, Delivery and Cyclicality in Learning Environments, The Delivery in the Studio, Teaching and Learning Principles.

Text Books:

1. *Orit Hazzan and Yael Dubinsky, Agile Software Engineering, Springer, 2009*
2. *Bernd Bruegge, Alan H Dutoit, Object-Oriented Software Engineering, Pearson Education, 3rd edition, 2014.*
3. *David C. Kung, "Object oriented software engineering", Tata McGraw Hill,2015*

Reference books:

1. *Cockburn, Agile Software Development, Pearson Education India*
2. *Mike Cohn, Agile Estimating and Planning, Pearson Education, 2005*
3. *Michele Sliger, Stacia Broderick, The Software Project Manager's Bridge to Agility, Addison-Wesley Professional, 2008*

Web Resources:

1. *www.allaboutagile.com/what-is-agile-10-key-principles/*
2. *https://www.versionone.com/agile*
3. *https://www.youtube.com/watch?v=MTEl3LEI4EQ*
4. *https://azure.microsoft.com/en-in/cloud-adoption-framework/*

2MCA5: THE DESIGN AND ANALYSIS OF ALGORITHM

Total Teaching Hours: 52 No. of Hours / Week: 04

UNIT - I [13 Hours]

Introduction: Algorithm, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Fundamental Data Structures. Fundamentals of the Analysis of Algorithm Efficiency: The Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non-recursive and Recursive Algorithms, Empirical Analysis of Algorithms, Algorithm Visualization.

UNIT - II [13 Hours]

Brute Force Method: Selection Sort and Bubble Sort, Sequential Search, Brute-Force String Matching, Exhaustive Search, Depth-First Search and Breadth-First Search. Decrease and Conquer: Insertion Sort, Topological Sorting, Algorithms for Generating Combinatorial Objects, Decrease-by-a-Constant-Factor Algorithms. Divide and Conquer: Merge Sort, Quick Sort, Binary Tree Traversals and Related Properties, Strassen's Matrix Multiplication.

UNIT - III [13 Hours]

Space and Time Tradeoffs: Sorting by Counting, Input Enhancement in String Matching, Hashing. Dynamic programming: Binomial Coefficient, Principle of Optimality, Optimal Binary Search Trees, Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms. Greedy Technique: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees.

UNIT - IV [13 Hours]

Limitations of Algorithm Power: Lower-Bound Arguments, Decision Trees, P, NP and NP-Complete Problems. Coping with the Limitations of Algorithm Power: Back Tracking: n-Queens problem, Hamiltonian Circuit Problem, Subset-Sum Problem, Branch-and-Bound: Assignment Problem, Knapsack Problem, Traveling Salesman Problem.

Text Books:

1. *Anany Levitin, "Introduction to the Design and Analysis of Algorithms", 3rd Edition, Pearson, 2012.*
2. *Horowitz, Sahni, Rajasekaran, "Fundamentals of Computer Algorithms", 2/e, Universities Press, 2007.*

Reference Books:

1. *Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, The MIT Press, 2009.*
2. *A.V. Aho, J.E. Hopcroft, J.D. Ullmann, "The design and analysis of Computer Algorithms", Addison Wesley Boston, 1983.*
3. *Jon Kleinberg, Eva Tardos, "Algorithm Design", Pearson Education, 2006.*

Web Resources:

1. *https://onlinecourses.nptel.ac.in/noc20_cs27/preview*
2. *https://web.stanford.edu/class/archive/cs/cs161/cs161.1138/*

2MCA6: ARTIFICIAL INTELLIGENCE

Total Teaching Hours: 52 No. of Hours / Week: 04

UNIT - I [13 Hours]

Introduction to AI: What is AI? Intelligent Agents: Agents and environment; Rationality; the nature of environment; the structure of agents. Problem solving: Problem-solving agents; Example problems; Searching for solution; Uninformed search strategies. Informed Search, Exploration, Constraint Satisfaction, Adversial Search: Informed search strategies; Heuristic functions; On-line search agents and unknown environment. Constraint satisfaction problems; Backtracking search for CSPs. Adversial search: Games; Optimal decisions in games; Alpha-Beta pruning.

UNIT - II [13 Hours]

Knowledge-based agents; The Wumpus world as an example world; Logic; propositional logic Reasoning patterns in propositional logic; Effective propositional inference; Agents based on propositional logic. Representation revisited; Syntax and semantics of first-order logic; Using first-order logic; Knowledge engineering in first-order logic. Propositional versus first-order inference; Unification and lifting, Forward chaining; Backward chaining; Resolution, Truth maintenance systems.

UNIT - III [13 Hours]

Basic plan generation systems – Strips -Advanced plan generation systems – K strips - Strategic explanations -Why, Why not and how explanations. Learning: Forms of Learning; Inductive learning; Learning decision trees; Ensemble learning; Computational learning theory. Handling Uncertainties: Non-monotonic reasoning, Probabilistic reasoning, use of certainty factors, Fuzzy logic.

UNIT - IV [13 Hours]

Computer Vision, Natural Language Processing: Introduction, Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing, Spell Checking. Robotics: Fundamentals of Robotics, Robot Kinematics: Position Analysis, Dynamic Analysis and Forces. Expert Systems: Need and justification for expert systems, Architecture and role of expert systems, Case studies: MYCIN, DART and XOON. Neural Networks: Introduction - Features of Biological neural networks, Neuron models and Network Architectures – Basics of ANN, CNN, RNN and applications, Machine Learning, Deep Learning.

Text Books:

1. Stuart Russel, Peter Norvig, "Artificial Intelligence A Modern Approach", 4th Edition, Pearson Education, 2020.

2. Ela Kumar, "Artificial Intelligence", I.K.International Publishing House Pvt.Ltd, 2008.

Reference Books:

1. Elaine Rich, Kevin Knight, "Artificial Intelligence", 3rd Edition, Tata McGraw Hill, 2009.

2. Nils J. Nilsson," Principles of Artificial Intelligence", Elsevier, 1980.

3. Dan W. Patterson, "Introduction to AI and ES", Pearson Education, 2007. (Unit- 3).

4. Andries P. Engelbrecht, "Computational Intelligence: An Introduction", John Wiley & Sons, 2nd edition, 2007.

5. John J. Craig, "Introduction to Robotics", Addison Wesley publication.

Web Resources:

1. https://www.journals.elsevier.com/artificial-intelligence
2. https://nptel.ac.in/courses/106/105/106105078/
3. http://neuralnetworksanddeeplearning.com/
4. https://nptel.ac.in/courses/106/106/106106226/
5. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/lecture-videos/

2MCA 7: DATABASE MANAGEMENT SYSTEMS LAB

PART – A

1. **Draw E-R diagram** and convert entities and relationships to relation table for a given scenario.
a. Two assignments shall be carried out i.e. consider two different scenarios (eg. bank, college)

Consider the Company database with following Schema

EMPLOYEE (FNAME, MINIT, LNAME, SSN, BDATE,ADDRESS,SEX,SALARY, SUPERSSN, DNO)

DEPARTMENT (DNAME, DNUMBER,MGRSSN,MSRSTARTDATE)

DEPT_LOCATIONS (DNUMBER, DLOCATION)

PROJECT (PNAME, PNUMBER, PLOCATION, DNUM)

WORKS_ON (ESSN, PNO<HOURS)

DEPENDENT (ESSN, DEPENDENT_NAME, SEX, BDATE, RELATIONSHIP)

2. Perform the following:
   a. Viewing all databases, Creating a Database, Viewing all Tables in a Database, Creating Tables (With and Without Constraints), Inserting/Updating/Deleting Records in a Table, Saving (Commit) and Undoing (rollback)
3. Perform the following:
   a. Altering a Table, Dropping/Truncating/Renaming Tables, Backing up / Restoring a Database.
4. For a given set of relation schemes, create tables and perform the following Simple Queries, Simple Queries with Aggregate functions, Queries with Aggregate functions (group by and having clause).
5. Execute the fallowing queries
   a. How the resulting salaries if every employee working on the 'Research' Departments is given a 10% raise.
   b. Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary, and the average salary in this department
6. Execute the fallowing queries
   a. Retrieve the name of each employee Controlled by department number 5 (use EXISTS operator).
   b. Retrieve the name of each dept and number of employees working in each department which has at least 2 employees
7. Execute the fallowing queries
   a. For each project, retrieve the project number, the project name, and the number of employee who work on that project.(use GROUP BY)
   b. Retrieve the name of employees who born in the year 1990's

8. For each department that has more than five employees, retrieve the department number and number of employees who are making salary more than 40000.

9. For each project on which more than two employees work, retrieve the project number, project name and the number of employees who work on that project.

10. For a given set of relation tables perform the following
    a. Creating Views (with and without check option), Dropping views, Selecting from a view

PART B

Create the following tables with properly specifying Primary keys, Foreign keys and solve the following queries.

BRANCH (Branchid, Branchname, HOD)
STUDENT (USN, Name, Address, Branchid, sem)
BOOK (Bookid, Bookname, Authorid, Publisher, Branchid)
AUTHOR (Authorid, Authorname, Country, age)
BORROW (USN, Bookid, Borrowed_Date)

1. Perform the following:
   a. Viewing all databases, Creating a Database, Viewing all Tables in a Database, Creating Tables (With and Without Constraints), Inserting/Updating/Deleting Records in a Table, Saving (Commit) and Undoing (rollback)
   Execute the following Queries:

2. a. List the details of Students who are all studying in 2nd sem MCA.
   b. List the students who are not borrowed any books.

3. a. Display the USN, Student name, Branch_name, Book_name, Author_name, Books_Borrowed_ Date of 2nd sem MCA Students who borrowed books.
   b. Display the number of books written by each Author.

4. a. Display the student details who borrowed more than two books.
   b.Display the student details who borrowed books of more than one Author.

5. a. Display the Book names in descending order of their names.
   b. List the details of students who borrowed the books which are all published by the same publisher.

Consider the following schema:
STUDENT (USN, name, date_of_birth, branch, mark1, mark2, mark3, total, GPA)

6. Perform the following:
   a. Creating Tables (With and Without Constraints), Inserting/Updating/Deleting Records in a Table, Saving (Commit) and Undoing (rollback)

7. Execute the following queries:
   a. Find the GPA score of all the students.
   b. Find the students who born on a particular year of birth from the date_of_birth column.

8. a. List the students who are studying in a particular branch of study.
   b. Find the maximum GPA score of the student branch-wise.

9 . a. Find the students whose name starts with the alphabet “S”.
   b. Update the column total by adding the columns mark1, mark2, mark3.

10. Execute the following queries:

a. Find the students whose name ends with the alphabets “AR”.
b. Delete the student details whose USN is given as 1001.

2MCA 8: UNIX PROGRAMMING LAB
PART-A

1. Learn the use of basic UNIX commands –
    a. To access information using date, history, man, who, whoami, uptime, finger,cal.
    b. To display contents of files using cat, vi, more, head, tail, grep, cmp, wc
    c. To manage files using cat, cp, ls, mv,rm, chmod, find
    d. Process utilities using ps, pid, ppid, tty, time, kill, exit
    e. Directory handling utilities using cd, mkdir, rmdir, mv, pwd

2. Write a shell script that displays list of all the files in the current directory to which the user has read, write and execute permissions.

3. Write a shell script that accepts a list of file names as its arguments, count and reports the occurrence of each word that is present in the first argument file on other argument files.

4. Write a shell script that accepts one or more file name as arguments and converts all of them to uppercase, provided they exist in the current directory.

5. Write grep commands to the following:
    a. To select the lines from a file that has exactly 2 characters.
    b. To select the lines from a file that has more2 than one blank spaces.

6. Write a shell script which accepts two file names as arguments. Compare the contents. If they are same, then delete the second file.

7. Write a shell script
    a. to count number of lines in a file that do not contain vowels.
    b. to count number of characters, words, lines in a given file.

8. Write a shell script to list all the files in a given directory.

9. Write a shell script to display list of users currently logged in.

10. Write a shell script to read three text files in the current directory and merge them into a single file and returns a file descriptor for the new file.

PART-B

1. Write a program to copy a file into another using system calls.

2. Write a program using system call: create, open, write, close, stat, fstat, lseek.

3. Write a program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen.
4. Write a program to create a Zombie process.
5. Write a program to implement inter process communication using pipes.
6. Simulate the following CPU scheduling algorithms
    a. Round Robin
    b. SJF
7. Write a program that illustrates file locking using semaphores.
8. Write a program that implements a producer-consumer system with two processes (using semaphores).
9. Write a program that illustrates inter process communication using shared memory system calls.
10. Write a program that illustrates the following:
    a. Creating message queue.
    b. Writing to a message queue
    c. Reading from a message queue

Reference Books:

1. Sumitabha Das: “UNIX Concepts and Applications”, 4th Edition, Tata McGraw Hill, 2006.
2. Kenneth Roson et al ,“UNIX: The Complete Reference”, McGraw-Hill Osborne Media
3. M G Venkateshmurthy,“UNIX and Shell Programming”, Pearson Education Asia, 2005
4. Behrouz A. Forouzan, Richard F. Gilberg,“Unix and shell Programming.”, Brooks/Cole-Thomson Learning, 2003
5. Uresh Vahalia, “UNIX Internals”, Pearson Education, 2005.
6. Richard Stevens, Stephen Rago, “Advanced Programming in the UNIX Environment”, Pearson Education, 2/e.

BANGALORE UNIVERSITY
Regulations of Master of Computer applications (MCA) Course

1 **TITLE OF THE COURSE:** The course shall be called MCA – Master of Computer Applications.

2 **DURATION OF THE COURSE**: The course of study shall be two years.

3 **ELIGIBILITY FOR ADMISSION**: A candidate with any degree of a minimum of 3 years duration (10+2+3) of Bangalore university or of any other University equivalent there in to with a minimum of 50% of marks in the aggregate of all subjects including languages, if any, provided further, that the candidate has studied Mathematics / Computer science /Business Mathematics / Statistics / Computer Applications / Electronics as a subject at PUC level or equivalent HSC (XII Standard) or at Degree level is eligible for admission to MCA Course. Relaxation to SC/ST, Group I be extended as per University norms.

4 **ATTENDANCE**: In each Semester a candidate should be considered to have successfully undergone the prescribed Course of study if the candidate has attended at least 75% of the classes in each subject (Theory, Lab & Practical).

5 **SCHEME OF EXAMINATION**:

A. The Internal Assessment marks should be decided for each of the theory subjects by conducting 2 tests, each of 60 minutes duration, spread over the span of a Semester. A seminar should also be given by the student in the second year and the same to be assessed and evaluated for internal assessment along with two tests.

B. The Internal Assessment marks in Practical course is based on the performance in the Laboratory. The Internal Assessment marks for Project work of a candidate is based on the dissertation and seminar.

6 **ELIGIBILITY TO GO TO THE HIGHER SEMESTER**:

A. A Candidate is allowed to carry over all the previous uncleared (failed) theory papers and Practicals to subsequent semesters from the first to fourth semester.

B. The maximum period for the completion of the course shall be four years from the date of admission.

7 **MINIMUM FOR PASS AND DECLARATION OF RESULTS**

A. For a pass in a semester, a candidate shall secure a minimum of 40% of the marks prescribed for a subject in the University Examination (Theory, Practical, Project work) and 50% of the marks in the aggregate inclusive of the Internal Assessment marks obtained in all subjects put together.

B. The candidates who do not satisfy 7(A) shall be deemed to have failed and have to take exams in the subjects in which he has secured less than 40% at the University examination.

C. Provision is made for rejection of results of all the subjects of a Semester only once, if the candidate decides to reappear for all the subjects of that semester. Such rejection should be made within 30 days of announcement of result, by making a written application, through the Head of the Institution. If such rejection is in respect of the results of all the subjects of one semester and earn fresh Internal marks as well.

D. The results of any semester will be declared as pass or fail as the case may be in accordance with regulation 7(A).

E. To be eligible for the award of the MCA degree, a candidate shall have completed the scheme of training and passed in all subjects prescribed for the Course.

F. Further to regulation 7(A), the classification followed by the University for all PG courses shall be made applicable for the declaration of results of each Semester.

8 **CLASSIFICATION OF RESULT FOR THE MCA COURSE AND DECLARATION OF RANKS:**

Further to regulations 7(A) and 7(F), the names of all successful candidates securing First Class with Distinction and First Class in the First attempt shall be arranged in the order of Merit and only first FIVE Ranks shall be declared.

9 A candidate shall complete examinations of all Semesters of MCA Course within - FOUR years from the date of admission

SCHEME OF STUDY AND EXAMINATION FOR MASTER OF COMPUTER APPLICATIONS (MCA)

Sem	Paper Code	Title of the paper	Hours / Week	Marks			Credits	
				IA	Exam	Total	Subject	Sem
I	1MCA1	The Art of Programming	4	30	70	100	4	28
	1MCA2	Discrete Mathematics	4	30	70	100	4	
	1MCA3	Computer Organization and Architecture	4	30	70	100	4	
	1MCA4	Theory of Computation	4	30	70	100	4	
	1MCA5	Object Oriented Programming	4	30	70	100	4	
	1MCA6	Data Structures	4	30	70	100	4	
	1MCA7	Data Structures Lab	8	30	70	100	2	
	1MCA8	Object Oriented Lab	8	30	70	100	2	
II	2MCA1	Operating Systems	4	30	70	100	4	28
	2MCA2	Database Management Systems	4	30	70	100	4	
	2MCA3	Computer Networks	4	30	70	100	4	
	2MCA4	Software Engineering	4	30	70	100	4	
	2MCA5	The Design and Analysis of Algorithm	4	30	70	100	4	
	2MCA6	Artificial Intelligence	4	30	70	100	4	
	2MCA7	Database Management System Lab	8	30	70	100	2	
	2MCA8	Unix Programming Lab	8	30	70	100	2	
III	3MCA1	Open Elective	3	30	70	100	3	26
	3MCA2	Soft Core	3	30	70	100	3	
	3MCA3	Research Methodology	4	30	70	100	4	
	3MCA4	Elective I	4	30	70	100	4	
	3MCA5	Elective II	4	30	70	100	4	
	3MCA6	Elective III	4	30	70	100	4	
	3MCA7	MINI PROJECT	8	30	70	100	4	
IV		MAIN PROJECT		120	280	400	16	16

3MCA2: QUANTITATIVE, TEACHING AND RESEARCH APTITUDE

Total Teaching Hours: 36 No. of Hours / Week: 03

UNIT – I [8 Hours]

Numbers Property – Simplification – Divisibility – HCF and LCM – Decimal Fractions –Square roots and Cube Roots – Logarithms – Antilogarithms - Surds and indices - Permutation and Combination – Probability – Odd man out series - Number series - letter series – codes – Relationships – classification.

UNIT – II [7 Hours]

Time and work – Problems on Ages – Calendar – Clock – Pipes and Cistern – Time and Distance – Problems of Train – Boats and Streams. Area – Volume and surface Areas – Heights and Distances – Data Interpretation: Tabulation – Bar Graphs – Pie Charts – Line Graphs. Data Interpretation - Sources, acquisition and interpretation of data; Quantitative and qualitative data; Graphical representation and mapping of data.

UNIT – III [7 Hours]

Simple Interest – Compound Interest – Stocks and Shares – True Discount – Banker's discount. Averages – Percentage – Profit and Loss - Ratio and Proposition – Partnership – Allegation and mixture – Chain rule. Understanding the structure of arguments; Evaluating and distinguishing deductive and inductive reasoning; Verbal analogies: Word analogy Applied analogy; Verbal classification; Reasoning Logical Diagrams: Simple diagrammatic relationship, multi diagrammatic relationship; Venn diagram; Analytical Reasoning.

UNIT – IV [7 Hours]

Teaching: Nature, objectives, characteristics and basic requirements; Learner's characteristics; Factors affecting teaching; Methods of teaching; Teaching aids; Evaluation systems. Research Aptitude: Meaning, characteristics and types; Steps of research; Methods of research; Research Ethics; Paper, article, workshop, seminar, conference and symposium; Thesis writing: its characteristics and format. Reading Comprehension: A passage to be set with questions to be answered. Communication: Nature, characteristics, types, barriers and effective classroom communication.

UNIT – V [7 Hours]

Higher Education System: Governance, Polity and Administration; Structure of the institutions for higher learning and research in India; formal and distance education; professional/technical and general education; value education: governance, polity and administration; concept, institutions

Reference

1. R.S. Aggarwal, Quantitative Aptitude, S. Chand & Company, New Delhi, 2012
2. Govind Prasad Singh and Rakesh Kumar, Text Book of Quickest Mathematics (for all Competitive Examinations),
3. Kiran Prakashan, 2012.R.S. Aggarwal, Objective Arithmetic, S. Chand & Company, New Delhi, 2005.

4. Dr. Lal,Jain,Dr. K. C. Vashistha, "U.G.C.- NET/JRF/SET Teaching & Research Aptitude", Upkar Prakashan, 2010.
5. "UGC NET/SLET: Teaching & Research Aptitude", Bright Publications, 2010.

3MCA3: RESEARCH METHODOLOGY

Total Teaching Hours: 52 No. of Hours / Week: 04

UNIT – I [12 Hours]

Introduction: Definition and objectives of Research – Types of research, Various Steps in Research process, Mathematical tools for analysis, Developing a research questionChoice of a problem Literature review, Surveying, synthesizing, critical analysis, reading materials, reviewing, rethinking, critical evaluation, interpretation, Research Purposes, Ethics in research – APA Ethics code.

UNIT – II [10 Hours]

Quantitative Methods for problem solving: Statistical Modeling and Analysis, Time Series Analysis Probability Distributions, Fundamentals of Statistical Analysis and Inference, Multivariate methods, Concepts of Correlation andRegression, Fundamentals of Time Series Analysis and Spectral Analysis, Error Analysis, Applications of Spectral Analysis.

UNIT – III [10 Hours]

Tabular and graphical description of data: Tables and graphs of frequency data of one variable, Tables and graphs that show the relationship between two variables , Relation between frequency distributions and other graphs, preparing data for analysis

UNIT - IV [10 Hours]

Soft Computing: Computer and its role in research, Use of statistical software SPSS, GRETL etc in research. Introduction to evolutionary algorithms - Fundamentals of Genetic algorithms, Simulated Annealing, Neural Network based optimization, Optimization of fuzzy systems.

UNIT - V [10 Hours]

Structure and Components of Research Report, Types of Report, Layout of Research Report, Mechanism of writing a research report, referencing in academic writing.

Reference

1. C.R. Kothari, Research Methodology Methods and Techniques, 2/e, Vishwa Prakashan, 2006.
2. Donald H.McBurney, Research Methods, 5th Edition, Thomson Learning, ISBN:81-315-0047-0,2006.
3. Donald R. Cooper, Pamela S. Schindler, Business Research Methods, 8/e, Tata McGraw-Hill Co. Ltd., 2006.
4. Fuzzy Logic with Engg Applications, Timothy J.Ross, Wiley Publications, 2nd Edition, 2004.
5. Simulated Annealing: Theory and Applications (Mathematics and Its Applications, by P.J. van Laarhoven & E.H. Aarts[e], 19.
6. Genetic Algorithms in Search, Optimization, and Machine Learning by David E. publisher

MACHINE LEARNING (ELECTIVE)

Total Teaching Hours: 52 No. of Hours / Week: 04

Module 1 : Introduction to Machine Learning [12 Hours]

Introduction**,** Perspectives & Issues in ML**,** designing learning systems, Concepts of hypotheses, Version space, inductive bias, Performance metrics-accuracy, precision, recall, sensitivity, specificity, AUC, RoC, Bias Variance decomposition. Decision Trees Learning: Basic algorithm (ID3), Hypothesis search and Inductive bias, Issues in Decision Tree Learning – Overfitting, Solutions to overfitting, dealing with continuous values.

Module 2: Supervised Learning with KNN, ANN, SVM [10 Hours]

Instance-based learning: k-nearest neighbour learning, Artificial Neural networks: Introduction, Perceptrons, Multi-layer networks and back-propagation, Activation Units, Support Vector Machines – margin and maximization, SVM - The primal problem, the Lagrangian dual, SVM – Solution to the Lagrangian dual.

Module 3: Probabilistic and Stochastic Models: [10 Hours]

Bayesian Learning – Bayes theorem, Concept learning, Maximum likelihood, Bayes optimal classifier, Gibbs algorithm, Naive Bayes classifier, Expectation maximization and Gaussian Mixture Models, Hidden Markov models

Module 4: Unsupervised Learning and Association Mining [10 Hours]

Hierarchical vs non-hierarchical clustering, Agglomerative and divisive clustering, K-meansclustering, Bisecting k-means, K-Means as special case of Expectation, Maximization, K-medoid clustering, Association Mining: Apriori algorithm. Finding frequent itemsets, mining association rules, FP-growth – FP trees, Mining frequent items from an FP-Tree, Dimensionality reduction techniques – PCA, SVD.

Module 5: Genetic Algorithms [10 Hours]

Genetic Algorithms – Representing hypothesis, Genetic operators and Fitness function and selection, Simple applications of the Genetic Algorithm, application of GA in Decision tree, Genetic Algorithm based clustering, Single Objective and Bi-objective optimization problems using GA, using GA to emulate Gradient descent/ascent.

Reference Books:

1. Ethem Alpaydin,"Introduction to Machine Learning", MIT Press, Prentice Hall of India, Third Edition 2014
2. Jiawei Han and Micheline Kambers and Jian Pei, "Data Mining –Concepts and Techniques", 3rd edition, Morgan Kaufman Pub
3. Charu C. Aggarwal, "Data Classification Algorithms and Applications", CRC Press, 2014.

4. Charu C. Aggarwal, “DATA CLUSTERING Algorithms and Applications”, CRC Press, 2014.

5. “Machine Learning”, Tom Mitchell, McGraw Hill Education (India), 2013.

BIG DATA & ANALYTICS (ELECTIVE)

Total Teaching Hours: 52 No. of Hours / Week: 04

Unit I [10 Hours]

Introduction to Big Data, Big data definition, enterprise / structured data, social / unstructured data, unstructured data needs for analytics, what is Big Data, Big Deal about Big Data, Big Data Sources, Industries using Big Data, Big Data challenges.

Unit II [10 Hours]

Data Pre-processing, why to pre-process data? Data cleaning: Missing Values, Noisy Data, Data Integration and transformation, Data Reduction: Data cube aggregation, Dimensionality Reduction, Data Compression, Numerosity Reduction, Data Mining Primitives, Languages and System Architectures: Task relevant data, Kind of Knowledge to be mined, Discretization and Concept Hierarchy

Unit III [10 Hours]

Introduction to Classification and Prediction, Issues regarding Classification, Classification using Decision trees, Bayesian Classification, Classification by Backpropagation, Prediction Classification Accuracy, Introduction of Clustering, Spatial mining, Web mining, Text mining

Unit III [10 Hours]

Introduction of Big data programming-Hadoop, History of Hadoop, The ecosystem and stack, Components of Hadoop, Hadoop Distributed File System (HDFS), Design of HDFS, Java interfaces to HDFS, Architecture overview, Development Environment, Hadoop distribution and-basic commands, Eclipse development.

Unit V [12 Hours]

Pig: Introduction to PIG, Execution Modes of Pig, Comparison of Pig with Databases, Grunt, Pig Latin, User Defined Functions, Data Processing operators.

Hive: Hive Shell, Hive Services, Hive Metastore, Comparison with Traditional Databases, HiveQL, Tables, Querying Data and User Defined Functions.

Hbase: HBasics, Concepts, Clients, Example, Hbase Versus RDBMS. Big SQL

Data Analytics with R Machine Learning: Introduction, Supervised Learning, Unsupervised Learning, Collaborative Filtering.

Case Study: Implement your leanings to find sectors in which different companies ought to inves

Reference

1. Seema Acharya, Subhasini Chellappan, "Big Data Analytics" Wiley 2015.

2. Tom White " Hadoop: The Definitive Guide" Third Edit on, O'reily Media, 2012.
3. Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph. By David Loshin, Elsevier, August 23, 2013.
4. White, T. (2012). Hadoop: The definitive guide. " O'Reilly Media, Inc."Smolan, R. (2013). The human face of big data.
5. Tom Plunkett, Mark Hornick, "Using R to Unlock the Value of Big Data: Big Data Analytics with Oracle R Enterprise and Oracle R Connector for Hadoop", McGraw-Hill/Osborne Media (2013), Oracle press.
6. Mayer-Schönberger, V., & Cukier, K. (2013). Big data: A revolution that will transform how we live, work, and think. Houghton Mifflin Harcourt. Holmes, A. (2012). Hadoop in practice. Manning Publications Co..
7. Simon, P. (2013). Too big to ignore: the business case for big data (Vol. 72). John Wiley & Sons.
8. Robert D. Schneider , Hadoop for Dummies, Wiley India.

CRYPTOGRAPHY AND NETWORK SECURITY (ELECTIVE)

Total Teaching Hours: 52 No. of Hours / Week: 04

Unit I [12 Hours]

Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms, A Model for Network Security, Symmetric Ciphers, Classical Encryption Techniques, Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Steganography

Unit II [10 Hours]

Block Cipher Principles, The Data Encryption Standard, The Strength of DES, Differential and Linear Cryptanalysis, Block Cipher Design The AES Polynomials with Coefficients in GF(28), Simplified AES, Multiple Encryption and Triple DES, Block Cipher Modes of Operation, Stream Ciphers and RC4

Unit III [10 Hours]

Fermat's and Euler's Theorem, The Chinese Remainder Theorem, The RSA Algorithm, Key Management, Diffie-Hellman Key Exchange, Elliptic Curve Arithmetic, Elliptic Curve Cryptography, Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs

Unit IV [10 Hours]

Digital Signatures, Authentication Protocols, Digital Signature Standard, Kerberos, X.509 Authentication Service, Public-Key Infrastructure, IP Security Overview, IP Security Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management

Unit V [10 Hours]

Web Security, Secure Socket Layer and Transport Layer Security, Intruders, Intrusion Detection, Password Management, Malicious Software, Firewalls

References

1. William Stallings, Cryptography and Network Security: Principles and Practice, 7th Edition, Pearson

CLOUD COMPUTING (ELECTIVE)

Total Teaching Hours: 52 No. of Hours / Week: 04

Unit I [10 Hours]

Introduction of Cloud Computing: What is Cloud Computing, How it works, Types of Cloud, Goals & Challenges, Leveraging Cloud Computing, Cloud Economics and Total Cost of Ownership Cloud Service Models.

Unit II [12 Hours]

Software as a Service (SaaS): Overview of the Cloud application development lifecycle, Challenges in SaaS Model, SaaS Integration Services, Advantages and Disadvantages. Infrastructure as a Services (IaaS): Evolution of infrastructure migration approaches, Virtual Machines, VM Migration Services, Cloud Infrastructure services, Advantages and Disadvantages.

Unit III [10 Hours]

Platform as a service (PaaS): Evolution of computing paradigms and related components (distributed computing, utility computing, Cloud computing, grid computing, etc.), Cloud platform services, Integration of Private and Public Cloud, Advantages and Disadvantages.

Unit IV [10 Hours]

Programming Model: Parallel and Distributed Programming Paradigms, MapReduce, Twister and Iterative MapReduce, Hadoop Library from Apache, Mapping Applications, Programming Support Google App Engine, Amazon AWS, Cloud Software Environments, Eucalyptus, Open Nebula, OpenStack, Aneka, CloudSim.

Unit V [10 Hours]

Cloud Security Tools and technologies: Infrastructure Security, Network level security, Host level security, Application level security, Data privacy and security Issues, Access Control and Authentication in cloud computing, the data security in Private and Public Cloud Architecture, Legal issues and Aspects, Multi-tenancy issues

References:

1. George Reese, "Cloud Application Architectures: Building Applications and Infrastructure in the Cloud" O'Reilly Gautam Shroff, Enterprise Cloud Computing, Cambridge University Press,2011
2. Judith Hurwitz, R Bloor, M.Kanfman, F.Halper "Cloud Computing for Dummies", Wiley India Edition, First Edition
3. Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, "Cloud Computing : Principles and Paradigms", Wiley Publication,2011
4. Kai Hwang, Geoffrey C Fox, Jack G Dongarra, "Distributed and Cloud Computing, From ParallelProcessing to the Internet of Things", Morgan Kaufmann Publishers, 2012
5. RajkumarBuyya, Christian Vecchiola, S.ThamaraiSelvi, 'Mastering Cloud Computing", TMGH,2013

WEB PROGRAMMING (ELECTIVE)

Total Teaching Hours: 52 No. of Hours / Week: 04

Unit I [12 Hours]

HTML5 and JavaScript: Local Storage, Web Workers, Drag and Drop, Introduction to Client-Side Scripting, JavaScript Basics, Functions, Objects, Hoisting, Arrays, JavaScript Objects

Unit II [10 Hours]

DOM and DOM Events: Accessing and modifying DOM, Events and Event Handlers - Load, Mouse, Synthetic Events, Key and Form Related Events, Event Bubbling, Cookies

Unit III [10 Hours]

Apache: MIME, http, httpd Server, Request Response Formats Basics, Configuration, Debugging, .htaccess

Unit IV [10 Hours]

AJAX: File Handling and System Calls, Strings and Regular Expressions, Arrays, Cookies, Sessions, Functions, Classes, Database Access AJAX: Asynchronous GET/POST using XMLHttpRequest

Unit V [10 Hours]

AJAX Advanced: JS objects, prototype inheritance, Dynamic Script Loading, XMLHttpRequest, Image- Based AJAX, Cross-Domain Access (CORS), Introduction to XML, Parsers, Styling RSS / Atom Feeds, JSON and XML, JSON vs XML.

Reference Book(s):

1. “JavaScript Absolute Beginner's Guide”, Kirupa Chinnathambi, Que Publishing, 1st Edition, 2017.
2. “Programming the World Wide Web”, Robert W Sebesta, Pearson, 7th Edition, 2013. 3. “HTML5 Up and Running”, Mark Pilgrim, O’Reilly, 1st Edition, 2015
3. “AJAX: The Complete Reference”, Thomas A Powell, McGraw Hill, 2008.