FOR QUALITY & EXCELLENCE
IN HIGHER EDUCATION
★★★★★
ACCREDITED BY NAAC

BANGALORE UNIVERSITY

REGULATIONS, SCHEME AND SYLLABUS

For the course

MASTER OF SCIENCE IN COMPUTER SCIENCE
(M Sc(CS))

I to IV Semesters

(Choice Based Credit System –Y2K14 Scheme)

Revised w.e.f.

Academic Year 2014-15 and onwards

MCA PROGRAMME
JNANABHARATHI CAMPUS
BANGALORE UNIVESITY, BANGALORE

SCHEME OF STUDY AND SCHEDULE OF EXAMINATION

1. Title of the course: Computer Science, M.Sc.
2. Duration of the course : 2 years (4 semesters)
3. Eligibility:
    a) B.Sc. (Computer Science) or BCA with Mathematics as one of the subject and atleast 50% aggregate marks of all optional subjects (throughout 3 years B.Sc. / BCA course), B.Sc. PCM with PG Diploma / Certificate in Computer Science of duration one year.

    b) The minimum requirement for SC / ST candidates are relaxed in accordance with University regulations.
4. Intake: 15 + supernumerary quota as per University regulations. Total number of students including payment seats not to exceed – 35. Payment seat fee is as per university guidelines.
5. Admission: A category wise merit list will be prepared with marks obtained in all optionals in all the three years.
6. Attendance: As per regulations of the University for P.G. courses.
7. Medium of instruction: English.
8. Scheme of study : Each semester is of 4 months duration I to III semester: Theory papers 4 , Practicals 2 in each semester. IV semester : Project, seminar and viva-voce, theory papers 4.
9. Scheme of examination: There shall be a University examination at the end of each semester.
    a) Appearance for the examination: As per regulations of the University for P.G. Courses.

    b) Provision for repeaters: As per regulations of the University for P.G. Courses.

    c) Dissertation and viva-voce examination: The period of dissertation work is on full semester (4th semester). A student has to select a guide from the department in consultation with the chairperson of the department.
10. Result declaration: As per regulations of the University for P.G. Courses.
11. Miscellaneous:
    a) It is recommended that tutorial work be provided for all theory and practical papers.

    b) Internal assessment:

Attendance	- 10
Seminars and Assignments	- 10
Mid-semester exam	- 10

    c) Lectures from experts in the field from R&D institutions are highly desirable.

Any other issue not envisaged above shall be resolved by the Vice-Chancellor in consultation with the appropriate bodies of the University, which shall be final and binding.

SCHEME OF STUDY AND EXAMINATION FOR MASTER OF SCIENCE IN COMPUTER SCIENCE (M Sc (CS))

Semester	Paper Code	Title of the paper	Hours / Week	Marks			Credits	
				IA	Exam	Total	Subject	Semester
I	MSC101T	File Structures	4	30	70	100	4	26
	MSC102T	Advanced Database Management Systems	4	30	70	100	4	
	MSC103T	Theory of Computation	4	30	70	100	4	
	MSC104T	Advanced Architecture	4	30	70	100	4	
	MSC105P	File Structures Lab	8	30	70	100	4	
	MSC106P	Advanced DBMS Lab	8	30	70	100	4	
	MSC107T	Soft Core – Quantitative, Teaching and Research Aptitude	3	30	70	100	2	
II	MSC201T	Object Oriented Analysis and Design using UML	4	30	70	100	4	26
	MSC202T	Advanced Java Programming	4	30	70	100	4	
	MSC203T	Artificial Intelligence	4	30	70	100	4	
	MSC204T	Quantitative Techniques	4	30	70	100	4	
	MSC205P	Object Oriented Design using UML Lab	8	30	70	100	4	
	MSC206P	Advanced Java Programming Lab	8	30	70	100	4	
	MSC207T	Soft Core – Soft Skill and Personality Development	3	30	70	100	2	
III	MSC301T	Advanced WEB Programming	4	30	70	100	4	24
	MSC302T	Advanced Algorithms	4	30	70	100	4	
	MSC303T	Cryptography and Network Security	4	30	70	100	4	
	MSC304T	Open Elective	4	30	70	100	4	
	MSC306P	Web Programming Lab	8	30	70	100	4	
	MSC307P	Advanced Algorithms Lab	8	30	70	100	4	
IV	MSC401T	Research Methodology	4	30	70	100	4	24
	MSC402T	Elective – 1	4	30	70	100	4	
	MSC403T	Elective – 2	4	30	70	100	4	
	MSC404T	Scilab Lb	4	30	70	100	4	
	MSC405P	Main Project	16	50	150	200	8	

FIRST SEMESTER MSc

MSC101T: FILE STRUCTURES

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

UNIT – I [12 Hours]

Introduction: File Structures, The Heart of the file structure Design, A Conceptual Toolkit; Fundamental File Operations: Physical Files and Logical Files, Opening Files, Closing Files, Reading and Writing, Seeking, Special Characters, The Unix Directory Structure, Physical devices and Logical Files, File-related Header Files, UNIX file System Commands; Buffer Management, Input /Output in UNIX. Fundamental File Structure Concepts, Managing Files of Records: Field and Record Organization, Using Classes to Manipulate Buffers, Using Inheritance for Record Buffer Classes, Managing Fixed Length, Fixed Field Buffers, An Object-Oriented Class for Record Files, Record Access, More about Record Structures, Encapsulating Record Operations in a Single Class, File Access and File Organization.

UNIT – II [10 Hours]

Organization of Files for Performance, Indexing: Data Compression, Reclaiming Space in files, Internal Sorting and Binary Searching, Key sorting; Index: Introduction, A Simple Index for Entry- Sequenced File, Object-Oriented support for Indexed, Entry-Sequenced Files of Data Objects, Indexes that are too large to hold in Memory, Indexing to provide access by Multiple keys, Retrieval Using Combinations of Secondary Keys. Consequential Processing and The Sorting of Large Files: A Model for Implementing Consequential Processes, Application of the Model to a General Ledger Program, Extension of the Model to include Multi-way Merging, A Second Look at Sorting in Memory, Merging as a Way of Sorting Large Files on Disk.

UNIT – III [10 Hours]

Multilevel indexing and B-Trees: The invention of B-Tree, Statement of the problem, Indexing with Binary Search Trees; Multi-Level Indexing, B-Trees, Example of Creating a B-Tree, An Object-Oriented Representation of B-Trees, B-Tree Methods; Nomenclature, Formal Definition of B-Tree Properties, Worst-case Search Depth, Deletion, Merging and Redistribution, Redistribution during insertion; B* Trees.

UNIT – IV [10 Hours]

Indexed Sequential File access and Prefix B+ Trees: Indexed Sequential Access, Maintaining a Sequence Set, Adding a Simple Index to the Sequence Set, The Content of the Index: Separators Instead of Keys, The Simple Prefix B+ Tree and its maintenance, Index Set Block Size, Internal Structure of Index Set Blocks: A Variable-order B- Tree, Loading a Simple Prefix B+ Trees, B-Trees, B+ Trees and Simple Prefix B+ Trees in Perspective.

UNIT – V [10 Hours]

HASHING: Introduction, A Simple Hashing Algorithm, Hashing Functions and Record Distribution, Collision resolution by progressive overflow, Buckets. How Extendible Hashing Works, Implementation, Deletion, Extendible Hashing Performance, Alternative Approaches.

Reference Books:

1. Michael J. Folk, Bill Zoellick, Greg Riccardi, “File Structures-An Object Oriented Approach with C++ “, 3rd Edition, Addison-Wesley.
2. Raghu Ramakrishan and Johannes Gehrke, “Database Management Systems”, 3rd Edition, McGraw Hill, 2003.
3. Robert L. Kruse, Bruce P. Leung, Clovis L.Tondo, “Data Structures and Program Design in C”, 2nd Edition, Prenctice Hall India, 2001.

MSC102T: THEORY OF COMPUTATION

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

UNIT – I [12 Hours]

Review of Mathematical Terms and Theory: Basic Mathematical Notations and Set Theory, Logic Functions and Relations, Language Definitions, Mathematical Inductions and Recursive Definitions. Finite Automata: Deterministic and Non Deterministic Finite Automata, U-Transitions, Conversion from NFA to DGA, Kleene's Theorem, Regular and Non Regular Languages.

UNIT – II [10 Hours]

Context Free Grammar: Introduction to CFG, CFG and Known Languages, Unions, Concatenations and *'s Notations and CFL, Derivatives of Trees and Ambiguity and Unambiguous CFG and Algebraic Expressions, Normal Forms and Simplified Forms.

Pushdown Automata, CFL and NFL: Introduction to PDA, Definition, DPDA, PDA Corresponding to CFG, CFG Corresponding to PDA, Introduction to CFL, Intersections and Complements of CFL, Decisions Problems and CFL.

UNIT – III [10 Hours]

Turing Machines, Recursive Language: Model of Computation and Church Turning Thesis, Definitions of Turing Machine, TM and Language Acceptors, Variations of TM, Non Deterministic TM, Universal TM, Enumerable and Language, Recursive and Non Recursive Enumerable.

UNIT – IV [10 Hours]

Computation Functions, Measuring, Classifications And Complexity: Primitive Recursive Functions, Halting Problem, Recursive Predicates and Some Bounded Operations, Unbounded Minimizations and μ-Recursive Functions, Godel Numbering, Computable Functions and μ-Recursive, Numerical Functions.

UNIT – V [10 Hours]

Tractable And Intractable Problems: Growth Rate and Functions, Time and Speed Complexity, Complexity Classes, Tractable and Possibly Intractable Problems, P and Np Completeness, Reduction of Time, Cook's Theorem, Np-Complete Problems.

Reference Books:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman, "Introduction to Automata Theory, Languages and Computation", 3rd Edition, Pearson Education, 2011.
2. John C Martin, "Introduction to Languages and Automata Theory", 3rd Edition, Tata McGraw-Hill, 2007.
3. Daniel I.A. Cohen, "Introduction to Computer Theory", 2nd Edition, John Wiley and Sons, 2009.
4. Thomas A. Sudkamp, "An Introduction to the Theory of Computer Science, Languages and Machines", 3rd Edition, Pearson Education, 2006.

MSC103T: ADVANCED DATABASES

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

UNIT – I [12 Hours]

Database Concepts: Characteristics of Database Approach – Data Models – Schemas-Three Schema Architecture and Data Independence; Database Design: ER Modelling – ER diagrams; Normalization; Relational Model and Query Processing

UNIT – II [10 Hours]

TRANSACTION PROCESSING AND CONCURRENCY CONTROL: Definition of Transaction and ACID properties;. Concurrency Control Techniques : Lock based Concurrency control -Optimistic Concurrency Control – Timestamp based Concurrency Control, Deadlocks ; Database Security: Security Issues – Control Measures-

Discretionary, mandatory and role based access control; Database Recovery Techniques: Recovery Concepts- Deferred Update and Immediate Update techniques – Shadow Paging – ARIES – Database backup and recovery

UNIT – III [10 Hours]

OBJECT ORIENTED, PARALLEL AND DISTRIBUTED DATABASES: Concept of Object Database: Object Definition Language ODL- Object Query Language; Object Database conceptual Design : Difference between ODB and RDB. Database System Architectures: Centralized and Client-Server Architectures - Parallel Systems- Distributed Systems ;

UNIT – IV [10 Hours]

Parallel Databases: I/O Parallelism – Inter and Intra Query Parallelism – Inter and Intra operation Parallelism – Design of Parallel Systems ;

Distributed Database Concepts - Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing.

UNIT – V [10 Hours]

EMERGING DATABASE TECHNOLOGIES: Multimedia Databases ; Spatial Databases ; XML and Web Databases ; Mobile Databases: Location and Handoff Management - Effect of Mobility on Data Management -Location Dependent Data Distribution - Mobile Transaction Models -Concurrency Control -Transaction Commit Protocols; Data Warehousing Data Mining; Text Mining.

Reference Books:

1. R. Elmasri, S.B. Navathe, "Fundamentals of Database Systems", Fifth Edition, Pearson Education/Addison Wesley, 2008.
2. Henry F Korth, Abraham Silberschatz, S. Sudharshan, "Database System Concepts", Sixth Edition, McGraw Hill, 2011.
3. C.J.Date, A.Kannan, S.Swamynathan, "An Introduction to Database Systems", Eighth Edition, Pearson Education, 2006.

MSC104T: ADVANCED COMPUTER ARCHITECTURE

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

UNIT – I [12 Hours]

Fundamentals of Computer design: Instruction set principles and examples- classifying instruction set - memory addressing- type and size of operands - addressing modes for signal processing-operations in the instruction set- instructions for control flow- encoding an instruction set. Overview of Parallel Processing and Pipelining Processing Necessity of high performance, Constraints of conventional architecture, Parallelism in uniprocessor system, Evolution of parallel processors, Architectural Classification, Applications of parallel processing

UNIT – II [10 Hours]

Parallel Computer methods: Multiprocessor and multi computers – Shared-Memory multiprocessors, Distributed-Memory Multiprocessors. Multi-vector and SIMD computers. PRAM and VLSI models - Architectural development tracks - Multiple-Processor Tracks, Multi-vector and SIMD Tracks, Multi-threaded and Dataflow Tracks. Program and Network properties: Conditions of parallelism - Program partitioning and scheduling - Program flow mechanism - System interconnect architecture. Principles of Scalable Performance: Performance metrics and measures - Speedup performance laws - Scalability analysis and approaches

UNIT – III [10 Hours]

Processors and Memory Hierarchy: Advanced processor technology - Super scalar and vector processors - Memory hierarchy technology - Virtual memory technology. Bus, Cache and Shared Memory: Bus System-Cache memory organizations-Shared memory organization-Sequential and weak consistency models.

UNIT – IV [10 Hours]

Instruction level Parallelism & Data Parallel Architectures: Instruction level parallelism (ILP)- over coming data hazards- reducing branch costs –high performance instruction delivery- hardware based speculation- limitation of ILP - ILP software approach-compiler techniques- static branch protection- VLIW approach- H.W support for more ILP at compile time- H.W verses S.W solutions - SIMD Architectures – Associative and Neural Architectures – Data-Parallel Pipelined and Systolic Architectures – Vector Architectures

UNIT – V [10 Hours]

Multiprocessors and Thread level parallelism: Multi-threaded Architectures, Distributed Memory MIMD Architectures, Shared Memory Architectures. Architecture of Multi-threaded processors, Latency hiding techniques, Principles of multithreading, Issues and solutions. Synchronization and Multiprocessing modes – Shared-Variable program structures, Message Passing program development, Mapping programs onto Multicomputers.

Reference Books:

1. Dezso Sima, Terence Fountain, Peter Kacsuk, “Advanced Computer Architectures – A Design Space approach”, Pearson Education, 2009
2. Kai Hwang, “Advanced Computer Architecture – Parallelism, Scalability, Programmability”, Tata McGraw-Hill, 2008.
3. John L. Hennessey and David A. Patterson, “Computer architecture – A quantitative approach”, Morgan Kaufmann / Elsevier Publishers, 5th Edition

MSC105P: FILE STRUCTURES LAB

1. Write a C++ Program to read series of names, one per line, from standard input and write these names spelled in reverse order to the standard output using I/O redirection and pipes. Repeat the exercise using an input file specified by the user instead of the standard input and using an output file specified by the user instead of the standard output.
2. Write a C++ program to read and write student object with fixed length records and the fields delimited by “|”. Implement pack(), unpack(), modify(), and search() methods.
3. Write a C++ program to read and write student objects with Variable-Length records using any suitable record structure. Implement pack(), unpack(), modify(), and search() methods.
4. Write a C++ program to read and write student objects with Variable-Length records using any suitable record structure and to read from this file a student record using RRN.
5. Write a C++ program to implement simple index on primary key for a file of student objects. Implement add(), search(), delete() using the index.
6. Write a C++ program to implement index on secondary key, the name, for a file of student objects. Implement add(), search(), delete() using the secondary index.
7. Write a C++ program to read two lists of names and then match the names in the two lists using sequential Match based on a single loop. Output the names common to both the lists.
8. Write a C++ program to read k Lists of names and merge them using k-way merge algorithm with k = 8.

9. Write a C++ program to implement B-Tree for a given set of integers and its operations insert() and search(). Display the tree.
10. Write a C++ program to implement B+ Tree for a given set of integers and its operations insert() and search(). Display the tree.
11. Write a C++ program to store and retrieve student data from file using hashing. Use any collision resolution techniques.
12. Write a C++ program to reclaim the free space resulting from the deletion of records using liked list.

MSC106P: ADVANCED DATA BASE MANAGEMENT SYSTEMS LAB

1. Database Customization
2. Creating Databases/Table spaces
3. Create Objects
4. Moving Data
5. Recovery
6. Locking
7. Preparing Applications for Execution using a front end tool
8. Application Performance Tool

The students are supposed to practice and develop a mini application for above mentioned lab. The students can do the activity in a group (team) consisting of not more than 2 students.

The entire application to be submitted by each team should be done with all the above activities. The examiner may ask to perform any of the above act

MCA107T: QUANTITATIVE, TEACHING AND REASERCH APTITUDE

UNIT – I

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

UNIT – II

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

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, multidiagrammatic relationship; Venn diagram; Analytical Reasoning.

UNIT – IV

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

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 and

Reference Books:

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

SECOND SEMESTER M Sc

MSC201T: OBJECT ORIENTED ANALYSIS AND DESIGN USING UML

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

UNIT – I [12 Hours]

Introduction: An overview - Object basics - Object state and properties, Behavior, Methods, Messages. Object Oriented system development life cycle, Benefits of OO Methodology. Overview of Prominent OO Methodologies: The Rumbaugh OMT, The Booch methodology, Jacobson's OOSE methodologies, Unified Process, Introduction to UML, Important views & diagram to be modelled for system by UML. Factional View (models): Use case diagram - Requirement Capture with Use case - Building blocks of Use Case diagram - actors, use case guidelines for use case models - Relationships between use cases - extend, include, generalize. Activity diagram - Elements of Activity Diagram - Action state, Activity state, Object, node, Control and Object flow, Transition (Fork, Merge, Join) - Guidelines for Creating Activity Diagrams - Activity Diagram - Action Decomposition (Rake) - Partition - Swim Lane.

UNIT – II [10 Hours]

Static structural view (Models): Classes, values and attributes, operations and methods, responsibilities for classes, abstract classes, access specification (visibility of attributes and operations). Relationships among classes: Associations, Dependencies., Inheritance - Generalizations, Aggregation. Adornments on Association: association names, association classes, qualified association, n-ary associations, ternary and reflexive association. Dependency relationships among classes, notations. Notes in class diagram, Extension mechanisms, Metadata, Refinements, Derived, data, constraint, stereotypes, Package & interface notation. Object diagram notations and modeling, relations among objects.

UNIT – III [10 Hours]

Class Modeling and Design Approaches: Three approaches for identifying classes - using Noun phrases, Abstraction, Use Case Diagram - Comparison of approaches - Using combination of approaches - Flexibility guidelines for class diagram: Cohesion, Coupling, Forms of coupling (identity, representational, subclass, inheritance), class Generalization, class specialization versus aggregation. Behavioral (Dynamic structural view): State diagram - State Diagram Notations, events (signal events, change events, Time events) - State Diagram states (composite states, parallel states, History states), transition and condition, state diagram behaviour(activity effect, do-activity, entry and exit activity), completion transition, sending signals.

UNIT – IV [10 Hours]

Interaction diagrams: Sequence diagram - Sequence diagram notations and examples, iterations, conditional messaging, branching, object creation and destruction, time constraints, origin of links, Activations in sequence diagram - Collaboration diagram - Collaboration diagram notations and examples, iterations, conditional messaging, branching, object creation and destruction, time constraints, origin of links, activations in sequence diagram. Approaches for developing dynamic systems: Top - down approach for dynamic systems - Bottom - up approach for dynamic systems - Flexibility Guidelines for Behavioral Design - guidelines for allocating and designing behaviors that lead to more flexible design.

UNIT – V [10 Hours]

Architectural view: Logical architecture: dependency, class visibility, sub systems - Hardware architecture: deployment diagram notations, nodes, object migration between node - Process architecture: what are process and threads and their notations in UML, object synchronization, invocation schemes for threads (UML notations for different types of invocations). Implementation architecture: component diagram notations and examples. Reuse: Libraries, Frame works components and Patterns: Reuse of classes, Reuse of components, Reuse of frameworks, black box framework, white box frame, Reuse of patterns: Architectural pattern and Design pattern.

Reference Books:

1. Charles Richter, “Designing Flexible Object Oriented systems with UML”
2. Jackson, Burd Thomson, “Object Oriented Analysis & Design”,
3. James Rumbaugh. Micheal Blaha, “Object oriented Modeling and Design with UML”.
4. Grady Booch, James Rumbaugh, Ivar Jacobson., “The Unified Modeling Language User Guide”, Pearson Education.
5. James Rumbaugh, “Object Oriented Modeling and Design”
6. Joseph Schmuilers, “Teach Yourself UML in 24 Hours”
7. Mike O'Docherty, “Object-Oriented Analysis and Design: using UML”, Wiley Publication

MSC202T: ADVANCED JAVA PROGRAMMING

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

UNIT – I [12 Hours]

Introduction: Data Types, Operators, Classes, Inheritance, Packages and Interfaces. Exception Handling, Concurrency and Multithreaded programming, Enumerations, Autoboxing, Annotations, I/O, Generics, String handling

UNIT – II [10 Hours]

JVM: Java Class file, Class Loader, Linking model, Garbage collection, Type conversion, Floating Point Arithmetic, Method Invocation and Return, Thread synchronization. Java I/O: Closeable, Flushable Interfaces, The Stream classes, Bytes Streams, Character Streams, Console Class, Serialization. Java Networking - Networking Classes and Interfaces, TCP/IP Sockets, Datagrams

UNIT – III [10 Hours]

Event Handling: Event Classes, Event Listener Interfaces, Adaptor Classes, Inner Classes. Comparable and Comparator. Java Sandbox security model, Applets. Server side programming - Java Servlets, JSP, Java XML library - JAXP, XML Parsing - DOM, SAX, Stax. Java Web Services - RESTful Web Services, SOAP Web Services

UNIT – IV [10 Hours]

Java Design patterns: Singleton, Observer, Adaptor, Proxy, Decorator, Factory, AbstractFactory, Fascade, Command, Template Method patterns, MVC .

UNIT – V [10 Hours]

Spring and Hibernate framework, Spring Flow, Hibernate Flow.

Reference Books:

1. Herbert Schildt, "Java The Complete Reference", 7th addition.
2. Ken Arnold, James Gosling, David Holmes, "The Java TM Programming Language", Addison-Wesley, 2006
3. Bill Venners, "Inside the Java 2 Virtual Machine", McGraw-Hill, 2nd edition, 2000.
4. Santhosh, “Spring and Hibernate”, Tata McGraw-Hill.

MSC203T: ARTIFICIAL INTELLIGENCE

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

UNIT-I [12 Hours]

Introduction to Artificial Intelligence: Definition. AI Applications, AI representation. Properties of internal Representation, Heuristic search techniques. Best first search, mean and end analysis, A* and AO* Algorithm. Minimize search procedure, Alpha beta cutoffs, waiting for Quiscence, Secondary search.

UNIT-II [10 Hours]

Knowledge representation using predicate logic: predicate calculus, Predicate and arguments, ISA hierarchy, frame notation, resolution, Natural deduction. Knowledge representation using non monotonic logic: TMS (Truth maintenance system), statistical and probabilistic reasoning, fuzzy logic, structure knowledge representation, semantic net, Frames, Script, Conceptual dependency.

UNIT-III [10 Hours]

Planning: block world, strips, Implementation using goal stack, Non linear planning with goal stacks, Hierarchical planning, list commitment strategy. Perception: Action, Robot Architecture, Vision, Texture and images, representing and recognizing scenes, waltz algorithm, Constraint determination, Trihedral and non trihedral figures labeling.

UNIT-IV [10 Hours]

Learning: Learning as induction matching algorithms. Failure driver learning, learning in general problem solving concept learning. Neural Networks: Introduction to neural networks and perception-qualitative Analysis only, neural net architecture and applications.

UNIT-V [10 Hours]

Natural language processing and understanding and pragmatic, syntactic, semantic, analysis, RTN, ATN, understanding sentences. Expert system: Utilization and functionality, architecture of expert system, knowledge representation, two case studies on expert systems.

Reference Books:

1. E. Charnaik and D. McDermott," Introduction to artificial Intelligence", Pearson Education, 1992.
2. Dan W. Patterson, “Introduction to Artificial Intelligence and Expert Systems”, PHI, 2003.
3. E. Rich and K. Knight," Artificial Intelligence", Tata McGraw Hill.2003.
4. Nils J. Nilson, “Principles of Artificial Intelligence”, Narosa Publishing Co. 2002.

MSC204T: OPTIMIZATION TECHNIQUES

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

UNIT - I [12 Hours]

Introduction to Operations Research: Basics definition, scope, objectives, phases, models and limitations of Operations Research. Linear Programming Problem – Formulation of LPP, Graphical solution of LPP. Simplex Method, Artificial variables, big-M method, two-phase method, degeneracy and unbound solutions.

UNIT - II [10 Hours]

Transportation Problem: Formulation, Solution, Unbalanced Transportation Problem. Finding Basic Feasible Solutions – Northwest corner rule, least cost method and Vogel's approximation method. Optimality test: the stepping stone method and MODI method. Assignment Model: Formulation. Hungarian method for optimal solution. Solving unbalanced problem. Traveling salesman problem and assignment problem.

UNIT - III [10 Hours]

Network Models: Definition, Minimum Spanning Tree algorithm, Shortest Route problem, Maximum flow problem. CPM & PERT: Network representation, Critical Path Computations, Linear Programming formulation of CPM, PERT Networks.

UNIT - IV [10 Hours]

Dynamic programming: Characteristics of dynamic programming. Dynamic Programming approach for Priority Management employment smoothening. Games Theory. Competitive games, rectangular game, saddle point, minimax (maximin) method of optimal strategies, value of the game. Solution of games with saddle points, dominance principle. Rectangular games without saddle point – mixed strategy for 2 X 2 games.

UNIT - V [10 Hours]

Queuing System: Elements of Queuing model, Pure birth and death models, Generalized Poission Queuing model, specialized poission. Queues: Steady-state Measure of performance, single sever models, Multiple server models, Matching serving model.

Reference Books:

1. J K Sharma., "Operations Research Theory & Applications , 3e", Macmillan India Ltd, 2007.
2. P. Sankara Iyer, "Operations Research", Tata McGraw-Hill, 2008.
3. P. K. Gupta and D. S. Hira, "Operations Research", S. Chand & co., 2007.
4. N.V.S. Raju, "Operations Research", HI-TECH, 2002.

MSC205P: OBJECT ORIENTED DESIGN USING UML LAB

1. The student should take up the case study of Unified Library application which is mentioned in the theory, and Model it in different views i.e. Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.
2. Student has to take up another case study of his/her own interest and do the same what ever mentioned in first problem. Some of the ideas regarding case studies are given in reference books, which were mentioned in theory syllabus, can be referred for some idea.

MSC206P: ADVANCED JAVA PROGRAMMING LAB

1. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result.

2. Write a Java program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.
3. Write a java program that simulates a traffic light. The program lets the user select one of three lights: red,yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time No light is on when the program starts
4. Write a Java Program to execute select query using JDBC
5. Write a Java Program to Create Thread using Interface and class.
6. Write a Java Program to Implement Producer and Consumer problem using Threads.
7. Write a Java Program to Implement DOM parser.
8. Write a Java Program to Implement SAX parser.
9. Write a Java Program to Implement Singleton design pattern using java.
10. Write a Java Program to Implement Factory and AbstractFactory design pattern using java.
11. Write a Java Program to Implement Observer Design pattern method using java.
12. Write a Java Program to Implement Adapter design design pattern using java
13. Write a Java Program to Implement proxy design pattern using java
14. Write a Java Program to Implement Helloworld program using servlets.
15. Write a JSP Program using Expression, Scriplet and Directive.

MCA207T: SOFT SKILLS AND PERSONALITY DEVELOPMENT

UNIT – I

Introduction to Soft Skills and Hard Skills, Break the ice berg –FEAR, Self Development - Etiquette and Manners. The Self Concept: Attitude, The process of attitude formation, positive attitude, How to build a success attitude, You are the chief architecture of yourself. Self Management Techniques. Believe in your self: Self Image and Self Esteem, Building Self Confidence, Environment we mix with, How to build self image?.

UNIT - II

Meaning and definition of personality, Personal Planning and Success Attitude: Prioritizing, Creating the master plan, Active positive visualization and Spot analysis. Self Motivation and Communication: Levels of motivation, power of irresistible enthusiasm, etiquettes and manners in a group, public speaking, Importance of listening and responding.

UNIT - III

Motivation Skills & Personality Development, Goal Setting, Career Planning, Resume Building, Psychometric Test, Priority Management & Time Management, Positive Attitude and Self Confidence. Verbal Communication includes Planning, Preparation Delivery, Feedback and assessment of activities like: Public speaking, Group Discussion, Oral Presentation skills, Perfect Interview, Listening and observation skills, body language and use of Presentation aids.

UNIT - IV

Written communication that includes project proposals, brochures, newsletters, articles. Etiquettes that include: etiquettes in social as well as office settings, email etiquettes, telephone etiquettes. Improving Personal Memory, study skills that include rapid reading, notes taking and creativity.

UNIT - V

Problem Solving and Decision Making Skills, Perceptive, Conceptual, Creative, Analytical and Decisive. Leadership as a process: co-ordination while working in a team, Leadership styles, Leader and Team player, Management of conflict, Profiles of great and successful personalities, Role of career planning in personality development, negotiation, Motivating.

Reference books:

1. Wallace : Personality Development 1st Edition, 2008 Cengage Learning India.
2. Succeed for your self -Richard Denny (3rd edition)- Kogan page India www.vivagroupindia.com.
3. Unleashing Leadership – John Hoover & Angelo Valenti – Jaico publishing House –WWW.JAICOBOOKS.COM
4. Kundu, C.l.- Personality development, Sterling Bangalore.
5. Listening and Responding – Sandra D.Collins-Cengage Learning India.
6. 1,001 ways to inspire your organization, your team and your self – David E. Rye-Jaico publishing house.