Central University of Himachal Pradesh

Shahpur Parisar

Department of Computer Science & Informatics

School of Mathematics, Computers & Information Sciences

Program Outcomes

Program Specific Outcomes

Course Outcomes & Course Contents

of

Master of Computer Applications (MCA)

CENTRAL UNIVERSITY OF HIMACHAL PRADESH
TEMPORARY ACADEMIC BLOCK SHAHPUR

Department Vision

- To communicate quality technical education to students focusing on computer application awareness imbibed of Software Development and protection with entrepreneurship and real time usage of industry and societal needs.

Department Mission

- Communicate the knowledge of latest software tools and applications to meet the challenges in Competitive fields of Computer Science and Applications.
- Imparting computational skills to students.
- Encourage research and industrial activities among students with advanced software's and tools.

Program Educational Objectives (PEO's)

- Apply knowledge to solve real-world problems using Computer Applications.
- Get employment as Software Professional capable of undertaking software development matching with the current and future needs through technological innovations, and inter-disciplinary works.
- Functions effectively as individuals and in any team at the workplace demonstrating ethical behavior, quality conscious Software Professional with sensitivity to the impact of technology on society.
- Work with confidence, commitment, capabilities to grow technically, temperamentally as; highly technical professionals, project management team members and leaders etc., through their continued efforts for lifelong learning to remain up to date in their professional pursuits.

Programme Specific Outcomes (PSO's)

PSO^1- To apply standard practices and techniques in software development

PSO^2- To apply security mechanisms for computer applications

PSO^3- To design and develop effective algorithms for computer applications

Programme Outcomes (PO’s)

PO^1 - Apply knowledge of Mathematics, Science, and Computer Applications.

PO^2 - Design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.

PO^3 - Identify, formulate, and solve real-world problems using Computer Applications.

PO^4 - Get an understanding of professional and ethical responsibility.

PO^5 - Have abilities to use the techniques, skills, and modern engineering tools necessary for software engineering practice.

Course Code: MCA 501

Course Name: Data Structures

Course Outcomes: After the successful completion of this course, the student will be able to

CO1: Understand the basic terminology, data structure classification and algorithms

CO2: Learn about the linked list structure and its various types.

CO3: Understand stack, array and queues.

CO4: Understand trees and graphs and their types.

Course Contents:

UNIT- I

Introduction: Basic Terminology, Data structures and its classification, Algorithm, Complexity space & time complexity, complexity notations- big Oh, Omega, Theta. Array Definition, Representation and Analysis of Arrays, Single and Multidimensional Arrays, Address calculation, Linear Search, Binary Search of Array, Traversing, Insertion & deletion in array, Sparse Matrices, Strings. Internal and External sorting, Insertion Sort, Bubble Sort, selection sort, Quick Sort, Merge Sort, Radix sort.

UNIT- II

Linked List Introduction, Representation of linked list in to memory, Memory allocation - Garbage Collection, Traversing & Searching in Linked List, Insertion into linked list- at beginning of list & at given location, Deletion in linked list- from starting of list & given location of node, Header Linked List, two way List, Input & output restricted liked list, Circular Header Linked List, Representation of Polynomials using linked List.

UNIT- III

Stack, Array Implementation of stack, Linked Representation of Stack, Application of stack: Conversion of Infix to Prefix and Postfix Expressions and Expression evaluation. Queue, Array and linked implementation of queues, Circular queues, D-queues and Priority Queues.

UNIT- IV

Trees: Basic terminology, Binary Trees, algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees, Binary Search Tree (BST), AVL Trees, B-trees. Graphs: Introduction, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees. Searching & Hashing: Sequential search, binary search, Hash Table, Hash Functions, Collision Resolution Strategies.

Text Books:

1. Lipschuitz L. Seymour, “Data Structures”, Schaum Outline Series, TMH.

2. R. S. Salaria, “Data Structures & Algorithm Using C”, Khanna Book Publishing. Reference Books: 1. Horowitz and Sahani, “Fundamentals of data Structures”, Galgotia Publication Pvt. Ltd., N Delhi. 2. R. S. Salaria, “Data Structures & Algorithm Using C++”, Khanna Book Publishing

3. A.M. Tenenbaum, “Data Structures using C & C++”, Prentice-Hall of India Pvt. Ltd., New Delhi.

4. Trembley and Sorenson, “Data Structures”, TMH Publications

Course Articulation Matrix of MCA 501- Data Structures

Course Outcomes	Programme Outcomes 1	Programme Outcomes 2	Programme Outcomes 3	Programme Outcomes 4	Programme Outcomes 5		Programme Specific Outcomes 1	Programme Specific Outcomes 2	Programme Specific Outcomes 3
CO1	1	1	3	2	2		1	3	2
CO2	1	3	1	1	1		1	1	3
CO3	1	2	1	3	3		3	3	2
CO4	2	1	2	3	3		2	1	3

1. Partially Related

2. Moderately Related

3. Highly Related

Course Code: MCA 502

Course Name: Data Structures Lab

Course Outcomes: After the successful completion of this course, the student will be able to

CO1: Implement various searching algorithms

CO2: Practically understand the implementation of Traversing and searching in various linked lists.

CO3: Learn practically implementation of stack using array, representation and applications

CO4: Implement Binary trees, searching and hashing.

Course Contents:

UNIT- I

Introduction: Basic Terminology, Data structures and its classification, Algorithm, Complexityspace & time complexity, complexity notations- big Oh, Omega, Theta. Array Definition, Representation and Analysis of Arrays, Single and Multidimensional Arrays, Address calculation, Linear Search, Binary Search of Array, Traversing, Insertion & deletion in array, Sparse Matrices, Strings. Internal and External sorting, Insertion Sort, Bubble Sort, selection sort, Quick Sort, Merge Sort, Radix sort.

UNIT- II

Linked List Introduction, Representation of linked list in to memory, Memory allocation - Garbage Collection, Traversing & Searching in Linked List, Insertion into linked list- at beginning of list & at given location, Deletion in linked list- from starting of list & given location of node, Header Linked List, two way List, Input & output restricted liked list, Circular Header Linked List, Representation of Polynomials using linked List.

UNIT- III

Stack, Array Implementation of stack, Linked Representation of Stack, Application of stack: Conversion of Infix to Prefix and Postfix Expressions and Expression evaluation. Queue, Array and linked implementation of queues, Circular queues, D-queues and Priority Queues.

UNIT- IV

Trees: Basic terminology, Binary Trees, algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees, Binary Search Tree (BST), AVL Trees, B-trees. Graphs: Introduction, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees. Searching & Hashing: Sequential search, binary search, Hash Table, Hash Functions, Collision Resolution Strategies.

Text Books:

1. Lipschuitz L. Seymour, “Data Structures”, Schaum Outline Series, TMH.

2. R. S. Salaria, “Data Structures & Algorithm Using C”, Khanna Book Publishing. Reference Books: 1. Horowitz and Sahani, “Fundamentals of data Structures”, Galgotia Publication Pvt. Ltd., N Delhi. 2. R. S. Salaria, “Data Structures & Algorithm Using C++”, Khanna Book Publishing

3. A.M. Tenenbaum, “Data Structures using C & C++”, Prentice-Hall of India Pvt. Ltd., New Delhi.

4. Trembley and Sorenson, “Data Structures”, TMH Publications

Course Articulation Matrix of MCA 502- Data Structures Lab

Course Outcomes	Programme Outcomes 1	Programme Outcomes 2	Programme Outcomes 3	Programme Outcomes 4	Programme Outcomes 5		Programme Specific Outcomes 1	Programme Specific Outcomes 2	Programme Specific Outcomes 3
CO1	1	1	3	2	2		1	3	2
CO2	1	3	1	1	1		1	1	3
CO3	1	2	1	3	3		3	3	2
CO4	2	1	2	3	3		2	1	3

1. Partially Related

2. Moderately Related

3. Highly Related

Course Code: MCA 503

Course Name: Operating System

Course Outcomes: After the successful completion of this course, the student will be able to

CO1: Understand structure, functions and process management of Operating Systems.

CO2: Learn about Inter Process Synchronization and Communications

CO3: Learn about the Memory Management.

CO4: Understand the Storage Management of Operating Systems.

Course Contents:

UNIT-I

Operating System Introduction: function, characteristics, structures–simple batch, multi-programmed, timeshared, personal computer, parallel, distributed systems, real-time systems, system components, operating system services, system calls, virtual machines. Process and CPU Scheduling: Process concepts and scheduling, operation on processes, cooperating processes, threads and inter-process communication scheduling criteria, scheduling algorithm, multiple-processor scheduling, real time scheduling.

UNIT-II

Management and Virtual memory: logical versus physical address space, swapping, contiguous allocation, paging, segmentation, segmentation with paging. Demand paging, performance of denuding paging, page replacement, page replacement algorithm, allocation of frames, thrashing.

UNIT-III

File System Interface and Implementation: access methods, directory, structure, protection, file system structure, allocation methods, free space management, directory management, directory implementation, efficiency and performance. I/O Management: I/O software and its types, disk scheduling. Process Management and Synchronization: Critical section problem, synchronization, critical regions, monitors.

UNIT-IV

Deadlocks: system model, dead locks characterization, methods for handling deadlocks, deadlock prevention, deadlock avoidance, deadlock detection and recovery from deadlock. Shell

Programming: vi editor, shell variables, I/O in shell, control structures, loops, subprograms, creating shell scripts. Basic system administration in Linux/Unix.

Text Books:

1. Silberschart, Galvin, Gagne, "Operating System Concepts", Ninth Edition, WSE Wiley.

2. Das, S., Your UNIX: The Ultimate Guide, Fourth Edition, McGraw-Hill Inc. Reference Book: 1. D.M. Dhamdhere, "Operating Systems: A Concept Based Approach", Tata McGraw-Hill. 2. Milan Milenkovic, "Operating system-concepts and design", McGraw Hill International Edition

3. A. S. Godbole, "Operating systems", Tata McGraw hill

4. Deitel H. M., "Operating System", Pearson Publications

5. William Stallings, "Operating Systems: Internals and Design Principles", Prentice-Hall of India

6. Andrew. S. Tanenbaum, "Modern operating systems", Pearson Prentice Hall.

Course Articulation Matrix of MCA 503- Operating System

Course Outcomes	Programme Outcomes 1	Programme Outcomes 2	Programme Outcomes 3	Programme Outcomes 4	Programme Outcomes 5		Programme Specific Outcomes 1	Programme Specific Outcomes 2	Programme Specific Outcomes 3
CO1	1	1	3	2	2		1	3	2
CO2	1	3	1	1	1		1	1	3
CO3	1	2	1	3	3		3	3	2
CO4	2	1	2	3	3		2	1	3

1. Partially Related

2. Moderately Related

3. Highly Related

Course Code: MCA 521

Course Name: Software Engineering

Course Outcomes: After the successful completion of this course, the student will be able to

CO1: Understand basics of software engineering.

CO2: Learn about software requirement specification and system design.

CO3: Understand about software project management, reliability and quality assurance.

CO4: Learn about testing and software maintenance.

Course Contents:

UNIT-I

Introduction: Introduction to software Engineering, Software characteristics, Software components, Software applications, Software Engineering Principles, Software metrics and measurement, monitoring and control. Software development life-cycle, Water fall model, prototyping model, Incremental model, Iterative enhancement Model, Spiral model.

Unit-II

Software Requirement Specification: Requirements Elicitation Techniques, Requirements analysis, Models for Requirements analysis, requirements specification, requirements validation. System Design: Design Principles: Problem partitioning, abstraction, Top down and bottom up – design, structured approach. Functional versus object-oriented approach of design, design specification, Cohesiveness and Coupling.Overview of SA/SD Methodology, structured analysis, data flow diagrams, extending DFD to structure chart.

Unit-III

Software project Management: Project planning and Project scheduling. Software Metrics: Size Metrics like LOC, Token Count, Function Count. Cost estimation using models like COCOMO. Risk management activities. Software Reliability and Quality Assurance: Reliability issues, Reliability metrics, reliability models, Software quality, ISO 9000 certification for software industry, SEI capability maturity model.

Unit-IV

Testing: Verification and validation, code inspection, test plan, test case specification. Level of testing: Unit, Integration Testing, Top down and bottom up integration testing, Alpha and Beta testing, System testing and debugging. functional testing, structural testing, Software testing strategies. Software Maintenance: Structured Vs unstructured maintenance, Maintenance Models, Configuration Management, Reverse Engineering, Software Re-engineering.

Text Book:

1. Software Engineering, “K. K. Aggarwal&Yogesh Singh”, 2E, New Age International, 2005

2. PankajJalote’s, “Software Engineering”, Wiley India

Reference Book:

1. Roger S. Pressman, “Software Engineering- A Practitioner’s Approach”, Tata McGraw Hill

2. Rajib Mall, “Fundamentals of Software Engineering”, PHI Learning Pvt. Ltd.

Course Articulation Matrix of MCA 521- Software Engineering

Course Outcomes	Programme Outcomes 1	Programme Outcomes 2	Programme Outcomes 3	Programme Outcomes 4	Programme Outcomes 5		Programme Specific Outcomes 1	Programme Specific Outcomes 2	Programme Specific Outcomes 3
CO1	1	1	3	2	2		1	3	2
CO2	1	3	1	1	1		1	1	3
CO3	1	2	1	3	3		3	3	2
CO4	2	1	2	3	3		2	1	3

1. Partially Related

2. Moderately Related

3. Highly Related

Course Code: MCA 534

Course Name: Object Oriented Programming using C++

Course Outcomes: After the successful completion of this course, the student will be able to

CO1: Understand data types, variables, pointers, functions and control structures.

CO2: Learn about classes and objects.

CO3: Learn about Inheritance and Virtual Functions.

CO4: Understand exception Handling and work with files.

Course Contents:

UNIT-I:

Data Types, Identifiers, Variables Constants and Literals, Basic input/output statements, Operators, Expressions, Type conversion, Control structures, Arrays, Strings, Structures and Pointers. Functions: Basic, Recursive functions, Overloaded functions, inline functions, function with default arguments.

UNIT-II:

Introduction to classes and objects, Access specifiers, Constructor, Destructor, Function overloading, Operator overloading, friend functions.

UNIT-III:

Inheritance-Concept of derived and base class, accessing base class members, Single inheritance, multiple inheritance, hierarchical inheritance, multilevel inheritance, hybrid inheritance, constructor in derived classes. Virtual Functions-Functions accessed with pointers, virtual member functions accessing with pointers, late binding, pure virtual functions, abstract classes, virtual base classes.

UNIT-IV:

Exception handling. Working with files- classes for file stream operations, opening and closing a file, detecting end-offile, file modes, file pointers and their manipulations, sequential input and output operations, updating a file, error handling. Command line arguments.

Text Book:

1. Balagurusamy, E. “Object Oriented Programming with C++”, 8E , Tata McGraw Hill.

Reference Book:

1. Herbert Schildt, “ C++ The Complete Reference ” , 4E , TMH Publication.
2. 2. RobertLafore, “Object Oriented Programming in Turbo C++”, 4E ,Galgotia Publications Pvt. Ltd.

Course Articulation Matrix of MCA 534- Object Oriented Programming

Course Outcomes	Programme Outcomes 1	Programme Outcomes 2	Programme Outcomes 3	Programme Outcomes 4	Programme Outcomes 5		Programme Specific Outcomes 1	Programme Specific Outcomes 2	Programme Specific Outcomes 3
CO1	1	1	3	2	2		1	3	2
CO2	1	3	1	1	1		1	1	3
CO3	1	2	1	3	3		3	3	2
CO4	2	1	2	3	3		2	1	3

1. Partially Related

2. Moderately Related

3. Highly Related

Course Code: MCA 504

Course Name: Theory of Computations

Course Outcomes: After the successful completion of this course, the student will be able to

CO1: Learn basics of theory of computation like alphabets, strings, automata and machines

CO2: Understand Chomsky classification of Languages and introduction to Context free grammars.

CO3: Learn about Push Down Automata

CO4: Learn about Turing machine, NP complete and hard problems

Course Contents:

UNIT-I

Introduction of Theory of Computation, Alphabet, Strings and their properties, Definition of an automation, Description of a finite Automation, Transition graph, transition function, Acceptability of a string by a Finite Automation, Deterministic and nondeterministic FSM'S, Equivalence of DFA and NDFA, Mealy & Moore machines, Minimization of finite automata.

UNIT-II

Chomsky classification of Languages, Languages and their relation, Languages and Automata, Regular sets, regular expression, Regular Grammars, Finite state machine and regular expression, Pumping lemma for regular sets, Application of pumping lemma, closure properties of regular sets. Introduction to CFG, Context-free languages and Derivation Trees, Ambiguity in context-free Grammars, simplification of context-free Grammars, Normal forms for context-free Grammars – Chomsky normal form and Greiback normal form.

UNIT-III

Pushdown Automata: Definitions – Moves – Instantaneous descriptions – Deterministic pushdown automata – Pushdown automata and CFL, PDA corresponding to given CFG, CFG corresponding to a given PDA, Closure properties of CFL's.

UNIT-IV

Introduction, TM model Representation of Turing machines, languages acceptability of TM, Design of TM, Universal TM & Other modification, Church's hypothesis, Properties of recursive and Recursively enumerable languages. Tractable and Untractable Problems: P, NP, NP complete and NP hard problems

Text Books:

1. John E. Hopcroft, Jeffery Ullman,"Introduction to Automata theory, Langauges& computation" , Narosa Publishers.
2. John C Martin, "Introdution to languages and theory of computation", McGraw Hill

Reference Books:

1. K.L.P Mishra & N.Chandrasekaran,"Theory of Computer Science", PHI Learning
2. Daniel I.A. Cohen,"Introduction to Computer Theory",Wiley India.
3. Peter Linz, "An Introduction to Formal Languages and Automata", Jones• & Bartlett Learning

Course Articulation Matrix of MCA 504- Theory of Computation

Course Outcomes	Programme Outcomes 1	Programme Outcomes 2	Programme Outcomes 3	Programme Outcomes 4	Programme Outcomes 5		Programme Specific Outcomes 1	Programme Specific Outcomes 2	Programme Specific Outcomes 3
CO1	1	1	3	2	2		1	3	2
CO2	1	3	1	1	1		1	1	3
CO3	1	2	1	3	3		3	3	2
CO4	2	1	2	3	3		2	1	3

1. Partially Related

2. Moderately Related

3. Highly Related

Course Code: MCA 505

Course Name: Data Base Management System

Course Outcomes: After the successful completion of this course, the student will be able to

CO1: Understand the basic concepts like entity, relationship and file organization

CO2: Learn entity-relationship model and relational data model.

CO3: Understand and implement Structured Query Language.

CO4: Learn about the Transaction processing concepts, Concurrency control techniques and Security.

Course Contents:

UNIT-I

Basic Concepts: Entity, Relationship and its types, Components of a database, three level architecture of a DBMS, Database models. File Organization: Serial, Sequential, Index Sequential and Direct file organization.

UNIT-II

Entity-Relationship Model: Entity Types, Entity Sets, Attributes & keys, Relationships,Relationships Types, Roles and Structural Constraints, Design issues, E-R Diagrams,Design of an E-R Database Schema, Reduction of an E-R Schema to Tables. Relational Data Model: Relational model concepts, Integrity constraints over Relations,Relational Algebra – Basic Operations. SQL: DDL, DML, and DCL, views & Queries in SQL, Specifying Constraints & Indexesin SQL.

UNIT-III

Relational Database Design: Functional Dependencies, Decomposition, Normal formsbased on primary keys (1 NF, 2 NF, 3 NF, & BCNF) Transaction Processing Concepts: Introduction to Transaction Processing, Transaction &System Concepts, Properties of Transaction, Schedules and Recoverability, SerializabilityofSchedules. Concurrency Control Techniques: Locking Techniques, Timestamp ordering, MultiversionTechniques, Optimistic Techniques, Granularity of Data items.

UNIT-IV

Databases for Advanced Applications: Active database concepts, Temporal database concepts, Spatial databases, Deductive databases; Emerging Database Technologies: Mobile databases, Multimedia Databases, Geographic information systems (GIS); XML and Internet Databases: Structured, Semi-structured and Unstructured Data, Introduction to web databases and XML, Structure of XML data.

Text Books:

1. R. Elmasri and S. B. Navathe, “Fundamentals of Database Systems”, 7E, Addison Wesley.

2. Bayross, I., “SQL, PL/SQL: The Programming Language of Oracle”,4E , BPB Publications.

Reference Books:

1. R. Ramakrishnan and J. Gehrke, “Database Management Systems”, 3E, McGraw Hill.

2. A. Silberschatz, H. Korth and S. Sudarshan, “Database System Concepts”, 6E, McGraw Hill.

Course Articulation Matrix of MCA 505- Data Base Management System

Course Outcomes	Programme Outcomes 1	Programme Outcomes 2	Programme Outcomes 3	Programme Outcomes 4	Programme Outcomes 5		Programme Specific Outcomes 1	Programme Specific Outcomes 2	Programme Specific Outcomes 3
CO1	1	1	3	2	2		1	3	2
CO2	1	3	1	1	1		1	1	3
CO3	1	2	1	3	3		3	3	2
CO4	2	1	2	3	3		2	1	3

1. Partially Related

2. Moderately Related

3. Highly Related

Course Code: MCA 506

Course Name: Data Base Management System Lab

Course Outcomes: After the successful completion of this course, the student will be able to

CO1: Apply practically SQL data types, operators and commands

CO2: Learn and apply functions, joins and views

CO3: Understand and apply PL/SQL variables, procedures and functions.

CO4: Learn about cursor, triggers, exception handling and packages with practical.

Course Contents:

UNIT-I

SQL: Data Types, Operators, DDL, DML, DCL, and TCL Commands.

UNIT-II

SQL: Integrity Constraints, Functions, Join, Indexes, Subqueries, Views.

UNIT-III

PL/SQL: Variables, Constants, Control Statements, Procedure, Functions.

UNIT-IV

PL/SQL: Cursor, Triggers, Exception Handling, Packages.

Text Books:

Ivan Bayross, “SQL, PL/SQL: The Programming Language of Oracle”. Fourth Edition, BPB Publications.

Course Articulation Matrix of MCA 506- Data Base Management System Lab

Course Outcomes	Programme Outcomes 1	Programme Outcomes 2	Programme Outcomes 3	Programme Outcomes 4	Programme Outcomes 5		Programme Specific Outcomes 1	Programme Specific Outcomes 2	Programme Specific Outcomes 3
CO1	1	1	3	2	2		1	3	2
CO2	1	3	1	1	1		1	1	3
CO3	1	2	1	3	3		3	3	2
CO4	2	1	2	3	3		2	1	3

1. Partially Related

2. Moderately Related

3. Highly Related

Course Code: MCA 507

Course Name: Design & Analysis of Algorithm

Course Outcomes: After the successful completion of this course, the student will be able to

CO1: Learn about algorithms, their designing ,analyzing and various notations.

CO2: Study about greedy strategies and dynamic programming concepts.

CO3: Understand Backtracking concept with examples.

CO4: Learn about Various trees, traversal techniques and NP-completeness.

Course Contents:

UNIT- I

Algorithms, designing algorithms, analyzing algorithms, asymptotic notations, Introduction to divide and conquer technique, analysis, design and comparison of various algorithms based on this technique, example binary search, merge sort, quick sort, strassen's matrix multiplication.

UNIT- II

Study of Greedy strategy, examples of greedy method like optimal merge patterns, Huffman coding, minimum spanning trees, knapsack problem, job sequencing with deadlines, single source shortest path algorithm. Concept of dynamic programming, problems based on this approach such as 0/1 knapsack, multistage graph, reliability design, Floyd-Warshall algorithm.

UNIT- III

Backtracking concept and its examples like 8 queen's problem, Hamiltonian cycle, Graph coloring problem etc.Introduction to branch & bound method, examples of branch and bound method like travelling salesman problem etc.Meaning of lower bound theory and its use in solving algebraic problem.

UNIT- IV

Binary search trees, height balanced trees, 2-3 trees, B-trees, basic search and traversal techniques for trees and graphs (In order, preorder, postorder, DFS, BFS), NP-completeness.

Text Books:

1. Ellis Horowitz, SartajSahni and SanguthevarRajasekaran, “Computer Algorithms”, 2E, Universities Press, 2007.

2. Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., “Introduction to Algorithms”, 2E, Prentice Hall of India Pvt. Ltd, 2003.

Reference Books:

1. Aho, A.V., Hopcroft J.E. and Ullman, J.D., “The Design and Analysis of Computer Algorithms”, Pearson Education, 1999.

2. Sara Baase and Allen Van Gelder, “Computer Algorithms, Introduction to Design and Analysis”, 3E, Pearson Education, 2009.

3. Dasgupta; “Algorithms”; TMH

4. Michael T Goodrich, RobartoTamassia, “Algorithm Design”, Wiely India

Course Articulation Matrix of MCA 507- Design & Analysis of Algorithm

Course Outcomes	Programme Outcomes 1	Programme Outcomes 2	Programme Outcomes 3	Programme Outcomes 4	Programme Outcomes 5		Programme Specific Outcomes 1	Programme Specific Outcomes 2	Programme Specific Outcomes 3
CO1	1	1	3	2	2		1	3	2
CO2	1	3	1	1	1		1	1	3
CO3	1	2	1	3	3		3	3	2
CO4	2	1	2	3	3		2	1	3

1. Partially Related

2. Moderately Related

3. Highly Related

Course Code: MCA 522

Course Name: Software Testing

Course Outcomes: After the successful completion of this course, the student will be able to

CO1: **U**nderstand about Software project management.

CO2: Learn Software reliability and Quality Assurance

CO3: Understand Testing, case strategies, types and debugging.

CO4: Learn Software maintenance and reverse engineering.

Course Contents:

UNIT-I

Introduction: Software Failures, Testing Process, Program and Software, Verification and Validation, Fault, Error, Bug and Failure, Test, Test Case and Test Suite, Deliverables and Milestones, Alpha, Beta and Acceptance Testing, Quality and Reliability, Testing, Quality Assurance and Quality Control, Static and Dynamic Testing, Testing and Debugging, Limitations of Testing, The V Shaped Software Life Cycle Model, Graphical Representation, Relationship of Development and Testing Parts Functional Testing: Boundary Value Analysis – Robustness Testing, Worst-Case Testing, Robust Worst-Case Testing, Applicability; Equivalence Class Testing – Creation of Equivalence Classes, Applicability; Decision Table Based Testing – Parts of the Decision Table, Limited Entry and Extended Entry Decision Tables, 'Do Not Care' Conditions and Rule Count, Impossible Conditions, Applicability; Cause-Effect Graphing Technique – Identification of Causes and Effects, Design of Cause-Effect Graph, Use of Constraints in Cause-Effect Graph, Design of Limited Entry Decision Table, Writing of Test Cases, Applicability

UNIT-II

Structural Testing: Control Flow Testing – Statement Coverage, Branch Coverage, Condition Coverage, Path Coverage; Data Flow Testing – Define/Reference Anomalies, Definitions, Identification of du and dc Paths, Testing Strategies Using du-Paths, Generation of Test Cases; Slice Based Testing – Guidelines for Slicing, Creation of Program Slices, Generation of Test Cases; Mutation Testing – Mutation and Mutants, Mutation Operators, Mutation Score Software Verification: Verification Methods – Peer Reviews, Walkthroughs, Inspections, Applications; Software Requirements Specification (SRS) Document Verification – Nature of the SRS Document, Characteristics and Organization of the SRS Document, SRS Document Checklist;

Software Design Description (SDD) Document Verification – Organization of the SDD Document, SDD Document Checklist; Source Code Reviews – Issues Related to Source Code Reviews, Checklist of Source Code Reviews; User Documentation Verification – Review Process Issues, User Documentation Checklist; Software Project Audit – Relevance Scale, Theory and Practice Scale, Project Audit and Review Checklist

UNIT-III

Creating Test Cases from Requirements and Use Cases: Use Case Diagram and Use Cases – Identification of Actors, Identification of Use Cases, Drawing of Use Case Diagram, Writing of Use Case Description; Generation of Test Cases from Use Cases – Generation of Scenario Diagrams, Creation of Use Case Scenario Matrix, Identification of Variables in a Use Case, Identification of Different Input States of a Variable, Design of Test Case Matrix, Assigning Actual Values to Variables; Guidelines for generating validity checks – Data Type, Data Range, Special Data Conditions, Mandatory Data Inputs, Domain Specific Checks; Strategies for Data Validity – Accept Only Known Valid Data, Reject Known Bad Data, Sanitize All Data; Database Testing Selection, Minimization and Prioritization of Test Cases for Regression Testing: What is Regression Testing – Regression Testing Process, Selection of Test Cases; Regression Test Cases Selection – Select All Test Cases, Select Test Cases Randomly, Select Modification Traversing Test Cases; Reducing the Number of Test Cases – Minimization of Test Cases, Prioritization of Test Cases; Risk Analysis – What is Risk, Risk Matrix; Code Coverage Prioritization Technique – Test Cases Selection Criteria, Modification Algorithm, Deletion Algorithm Software Testing Activities: Levels of Testing – Unit Testing, Integration Testing, System Testing, Acceptance Testing; Debugging – Why Debugging is so Difficult, Debugging Process, Debugging Approaches, Debugging Tools; Software Testing Tools – Static Software Testing Tools, Dynamic Software Testing Tools, Process Management Tools; Software Test Plan

UNIT-IV

Object Oriented Testing: What is Object Orientation – Classes and Objects, Inheritance, Messages, Methods, Responsibility, Abstraction, Polymorphism, Encapsulation, What is Object Oriented Testing – What is a Unit, Levels of Testing; Path Testing, Activity Diagram, Calculation of Cyclomatic Complexity, Generation of Test Cases; State Based Testing – What is a State Machine, State Chart Diagram, State Transition Tables, Generation of Test Cases; Class Testing – How Should We Test a Class, Issues Related to Class Testing, Generating Test Cases Metrics and Models in Software Testing: Software Metrics – Measure, Measurement and Metrics, Applications, Categories of Metrics – Product Metrics for Testing, Process Metrics for Testing; Object Oriented Metrics Used in Testing – Coupling Metrics, Cohesion Metrics, Inheritance Metrics, Size Metrics; What Should We Measure During Testing – Time, Quality of Source Code, Source Code Coverage, Test Case Defect Density, Review Efficiency; Software Quality Attributes Prediction Models – Reliability Models, An Example of Fault Prediction Model in Practice, Maintenance Effort Prediction Model Automated Test Data Generation: What is Automated Test Data Generation – Test Adequacy Criteria, Static and Dynamic Test Data Generation; Approaches to Test Data Generation – Random Testing, Symbolic Execution, Dynamic Test Data Generation; Test Data Generation using Genetic Algorithm – Initial Population, Crossover and Mutation, Fitness Function, Selection, Algorithm for Generating Test Data; Test Data Generation Tools

Text Book:

1. YogeshSingh , “Software Testing” , Cambridge University Press.

Reference Books:

1. Paul C. Jorgensen, “Software Testing: A Craftsman’s Approach”, 4E, CRC Press.
2. Boris Beizer, “Software Testing Techniques”, 2E, Dreamtech Press.
3. A.P. Mathur, “Fundamentals of Software Testing”, Pearson.
4. Desikan& G. Ramesh, “Software Principals and Practices”, Pearson.
5. G.J. Myers, T. Badgett, C. Sandler, “The Art of Software Testing”, 3E, Wiley India.

Course Articulation Matrix of MCA 522- Software Testing

Course Outcomes	Programme Outcomes 1	Programme Outcomes 2	Programme Outcomes 3	Programme Outcomes 4	Programme Outcomes 5		Programme Specific Outcomes 1	Programme Specific Outcomes 2	Programme Specific Outcomes 3
CO1	1	1	3	2	2		1	3	2
CO2	1	3	1	1	1		1	1	3
CO3	1	2	1	3	3		3	3	2
CO4	2	1	2	3	3		2	1	3

1. Partially Related

2. Moderately Related

3. Highly Related