SIDDHARTH INSTITUTE OF ENGINEERING &TECHNOLOGY :: PUTTUR

(AUTONOMOUS)

Master of Computer Applications (MCA)

Course Structure

MCA I Year - I Semester

S. No.	Course Code	Subject	L	T	P	C
1	20HS0836	Discrete Mathematics	4	-	-	4
2	20MC9101	Computer Organization	3	1	-	4
3	20MC9102	Data Structures	3	1	-	4
4	20MC9103	Operating System	3	1	-	4
5	20MC9104	Object Oriented Programming through C++	3	1	-	4
6	20MC9105	Data Structures using C Lab	-	-	3	1.5
7	20MC9106	Object Oriented Programming through C++ Lab	-	-	3	1.5
8	20MC9107	P.C. Software Lab	-	-	4	2
		Contact periods/Week	16	4	10	25
			Total/Week 30			

MCA I Year - II Semester

S. No.	Course Code	Subject	L	T	P	C
1	20MC9108	Computer Networks	3	1	-	4
2	20MC9109	JAVA Programming	3	1	-	4
3	20MC9110	Computer Graphics	3	1	-	4
4	20MC9111	Database Management System	3	1	-	4
5	20MC9112	Software Engineering	3	1	-	4
6	20MC9113	JAVA Programming Lab	-	-	3	1.5
7	20MC9114	Computer Graphics Lab	-	-	3	1.5
8	20MC9115	Database Management System Lab	-	-	4	2
Credit Course						
9	COE-I	Comprehensive Online Examination – I	-	-	-	1
Audit Course – I						
10	20HS0863	Professional Communication Skills Lab	-	-	2	-
		Contact periods/Week	15	5	14	26
			Total/Week 34			

MCA II Year -I- Semester

S. No.	Course Code	Subject	L	T	P	C
1	20MC9116	Web Technologies	3	1	-	4
2	20MC9117	Mobile Application Development	3	1	-	4
3	20MC9118	Data Warehousing and Data Mining	3	1	-	4
4	20MC9119 20MC9120 20MC9121 20MC9122 20MC9123	**Programme Elective – I** Software Testing Artificial Intelligence Distributed Systems Linux Programming Network Security	3	1	-	4
5	20MC9124 20MC9125 20MC9126 20MC9127 20MC9128	**Programme Elective – II** Human Computer Interaction Advanced Programming(Python & R-Languages) Internet of Things E-Commerce Block Chain Technology	3	1	-	4
6	20MC9129	Web Technologies Lab	-	-	4	2
7	20MC9130	Mobile Application Development Lab	-	-	4	2
8	20MC9131	Data Warehousing and Data Mining Lab	-	-	4	2
Credit Course						
9	COE-II	Comprehensive Online Examination – II	-	-	-	1
Audit Course – II						
10	20HS0842	Aptitude Practices	3	-	-	-
		Contact periods/Week	18	5	12	27
			Total/Week 35			

MCA II Year-II- Semester

S. No.	Course Code	Subject	L	T	P	C
1	20MC9132	Cloud Computing	3	1	-	4
2	20MC9133 20MC9134 20MC9135 20MC9136 20MC9137	**Programme Elective – III** Cyber Security Machine Learning Software Project Management .Net Technologies Bio-informatics	3	1	-	4
3	20MC9138 20MC9139 20MC9140 20MC9141 20MC9142	**Programme Elective – IV** Neural Networks & Fuzzy logic Image Processing Design Patterns Big Data Analytics Enterprise Resource Planning	3	1	-	4
4	20MC9143	Seminar	-	-	4	2
5	20MC9144	Dissertation	-	-	20	10
		Contact periods/Week	09	3	24	24
			Total/Week 36			

Note: L: Lecture Hour, T: Tutorial, P: Practical, C: Credit
Total Credits: 102

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
4	-	-	4

(20HS0836) DISCRETE MATHEMATICS

COURSE OBJECTIVES

The objectives of this course:

1. *To train the students thoroughly in Mathematical concepts of Mathematical logic, Relations, Algebraic structures, Recurrence Relation, Graph Theory.*
2. *To prepare students for lifelong learning and successful careers using Mathematical concepts of Mathematical logic, Relations, Algebraic structures, Recurrence Relation, Graph Theory.*
3. *To develop the skill pertinent to the practice of the Mathematical concepts including the students' abilities to formulate and modeling the problems, to think creatively and to synthesize information.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Write an argument using logical notation and determine if the argument is or is not valid.*
2. *Demonstrate the ability to write and evaluate a proof or outline the basic structure of and give examples of each proof technique described.*
3. *Understand the basic principles of sets and operations in sets.*
4. *Demonstrate an understanding of relations and functions and be able to determine their properties.*
5. *Determine when a function is 1-1 and "onto".*
6. *Demonstrate different traversal methods for trees and graphs.*
7. *Model problems in Computer Science using graphs and trees.*

UNIT-I

Mathematical Logic:

Statements and notations, Connectives, Truth Tables, Well-formed formulae, tautology, equivalence implication, Rules of inference, Consistency, proof of Contradiction, Normal forms.

Predicates: Predicative logic, Free & Bound variables, Quantifiers, Rules of inference, Automatic Theorem Proving.

UNIT-II

Relations:

Properties of Binary Relations, equivalence, transitive closure, compatibility and partial ordering relations.

Functions: Inverse Function, Composition of functions, recursive Functions.

Algebraic structures:

Algebraic systems examples and general properties, Semi groups and monoids, groups, sub groups, homomorphism, Isomorphism.

UNIT-III

Elementary Combinatorics:

Basis of counting, Enumerating Combination & Permutation with repetition, Constrained repetitions, Binomial & Multinomial theorems (without proof), Binomial Coefficients, The principle of Inclusion & Exclusion - Pigeon hole principle and its application.

UNIT-IV

Recurrence Relation:

Generating Functions & Sequences, Calculating Coefficient of generating function, Recurrence relations, Solving Recurrence relation by substitution and Characteristic roots, solution of Inhomogeneous Recurrence Relation. Solving Recurrence relation by Generating function.

UNIT-V

Graph Theory:

Representation of Graphs, Sub graphs, Multi graphs, Graph Theory Applications: Basic Concepts Isomorphism. Euler circuits, planar Graphs, Hamiltonian graphs, Chromatic Numbers, Spanning Trees, DFS, BFS.

TEXT BOOKS

1. Thomas Koshy, *Discrete Mathematics with Applications*, Elsevier, 2004.
2. Chandra Sekharaiah D.S, *Mathematical Foundations of computer Science (Discrete Structures)*, Prism, 2012.

REFERENCES

1. Liu C.L & Mohapatra D.P, *Elements of Discrete Mathematics*- A Computer Oriented Approach, 3rd edition, McGraw Hill Education, 2012.
2. Mott J.L, Kandel A & Baker T.P, *Discrete Mathematics for Computer Scientists & Mathematicians*, 2nd edition, Prentice Hall of India , 2008.
3. Tremblay J.P and Manohar R, *Discrete Mathematical Structures with Application to Computer Science*, McGraw Hill Publication, 2017.
4. Satyanarayana Bh & Syam Prasad K, *Discrete Mathematics and Graph Theory*, Printice Hall of India, New Delhi, April 2014, ISBN: 978-81-203-4948-3.
5. Satyanarayana Bh, Venkata Pradeep Kumar T & Mohiddin Shaw S, *Mathematical Foundation of Computer Science*, BS Publications (A unit of BSP Books Pvt. Ltd.), Hyderabad, India, 2016, ISBN: 978-93-83635-81-8.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
3	1	-	4

(20MC9101) COMPUTER ORGANIZATION

COURSE OBJECTIVES

1. *To impart the knowledge in the field of digital electronics.*
2. *To impart knowledge about the various components of a computer and its internals.*
3. *To design and realize the functionality of the computer hardware with basic gates and other components using combinational and sequential logic.*

COURSE OUTCOMES

1. *Able to design digital circuits by simplifying the Boolean functions*
2. *Able to understand the organization and working principle of computer hardware components*
3. *Able to understand mapping between virtual and physical memory*
4. *Acquire knowledge about multiprocessor organization and parallel processing*
5. *Able to understand the importance of the hardware-software interface.*
6. *Able to trace the execution sequence of an instruction through the processor.*

UNIT-I

Number Systems And Computer Arithmetic: Number base Conversions - Addition and subtraction - multiplication - division - Floating point representation - logical operation - Gray code - BCD codes - Error detecting codes - Boolean algebra - Simplification of Boolean expressions - K-Maps.

Combinational And Sequential Circuits: Decoders - Encoders - Multiplexers - Adder - flip-flops.

UNIT-II

Memory Organization: Memory hierarchy - Main memory - RAM - ROM chips - Memory address map - memory contention to CPU - Cache Memory - Associative mapping - direct mapping - Set-associative mapping.

Micro Programmed Control: Control memory - Address sequencing - Micro program example - design of control unit -hard wired control - and Micro programmed control.

UNIT-III

Basic CPU Organization: Introduction to CPU - INTEL - 8086 CPU architecture - Instruction format- Zero, one, two, and three address instructions - Addressing modes - generation of physical address - code segment registers.

Intel 8086 Assembly Language Instructions: assembler directives - Data transfer instructions - input-output instructions - address transfer - Flag transfer – arithmetic, logical, shift and rotate instructions. Conditional and unconditional transfer - iteration control - interrupts and process control instructions - Programming with assembly language instructions.

UNIT-IV

Input Output Organization: Peripheral devices - input-output interface-I/O Bus and interface modules - I/O versus Memory bus - isolated versus memory mapped I/O - Modes of transfer-Programmed I/O - Interrupt-initiated I/O - priority interrupts-Daisy chaining - parallel priority - interrupt cycle.

DMA: DMA control - DMA transfer - Input output processor-CPU-IOP communication.

UNIT-V

Pipeline and Vector Processing: Parallel Processing - Pipelining - Arithmetic Pipeline - Instruction Pipeline - RISC Pipeline - Vector Processing - Array Processors.

Multiprocessors: Characteristics of Multiprocessors - Interconnection Structures - Interprocessor Arbitration. Interprocessor Communication and Synchronization - Cache Coherence - Shared Memory Multiprocessors.

TEXT BOOKS

1. *Computer System Architecture,* M. Morris Mano, 3rd Edition, PHI/Pearson Education.
2. *Microprocessors and Interfacing*, Douglas Hall, Tata McGraw-Hill.

REFERENCES

1. *Computer Organization, Carl Hamacher, ZvonkoVranesic, SafwatZaky, Vth Edition, McGraw Hill.*
2. *Fundamentals of Computer Organization and Design, Sivarama, P.Dandamudi, Springer Int. Edition.*
3. *Digital Logic & Computer Design, M. Morris Mano, PHI/Pearson Education*
4. *Computer Organization and Architecture, William Stallings, 7th Edition, Pearson/PHI, 2007.*

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
3	1	-	4

(20MC9102) DATA STRUCTURES

COURSE OBJECTIVES

1. *To understand the basic structure concept such as Abstract Data Types, Linear and Non Linear Data structures.*
2. *To understand the behavior of data structures such as stacks, queues, trees, directories, trees, Graph and their representations.*
3. *To understand and analyze various searching and sorting algorithms.*

COURSE OUTCOMES

1. *Learn how to use data structure concepts for realistic problems.*
2. *Ability to identify appropriate data structure for solving computing problems in C language.*
3. *Ability to solve problems independently and think critically.*
4. *Able to search and sort the elements in graphs and trees.*
5. *Ability to solve linked list problems.*
6. *Ability to solve queues and hash tables.*

UNIT - I

Basic Concepts: Data objects and Structures - Algorithm Specification-Introduction - Recursive algorithms - Data Abstraction - Performance analysis- time complexity and space complexity - Asymptotic Notation-Big O, Omega and Theta notations - Complexity Analysis Examples - Introduction to Linear and Non Linear data structures.

Linear Lists - Array Representation: Data Objects and Structures - The Linear List Data structure - Array Representation - Multiple Lists in a Single Array - Performance Measurement.

UNIT - II

Linear Lists - Linked Representation: Singly Linked Lists- Operations-Insertion, Deletion - Circularly linked lists-Operations for Circularly linked lists - Doubly Linked Lists-Operations- Insertion, Deletion.

Stack / Queue: ADT - definition - array and linked implementations - applications-infix to postfix conversion - Postfix expression evaluation - recursion implementation. Circular queues - Insertion and deletion operations.

UNIT - III

Trees: definition - terminology - Binary trees-definition - Implementing Binary Trees - Searching a Binary Tree - Tree traversals – Breadth-first Traversal - Depth-first Traversal - stackless Depth-first Traversal – Insertion, Deletion - Balancing a Tree - Heaps.

Multiway Trees: Family of B – Trees - B* Trees - B+ Trees - Prefix B+ Trees - Bit – Trees - R – Trees - 2 – 4 Trees - Sets and Multisets - Maps and Multimaps in the Standard Template Library - Tries

UNIT - IV

Searching: Linear Search - Binary Search - Hashing-Introduction - hash tables - hash functions - Overflow Handling - Comparison of Searching methods.

Sorting: Elementary Sorting Algorithm - Insertion Sort, Selection Sort and Bubble Sort. Efficient Sorting Algorithms - Shell Sort, Heap Sort, Quick sort, Merge sort and Radix Sort.

UNIT - V

Graphs: Definitions - Terminology - Operations - Applications and Properties - Graph Traversals.

Graph Algorithms: Minimum-Cost Spanning Trees- Prim's Algorithm -Kruskal's Algorithm Shortest Path Algorithms -Dijkstra's Algorithm.

TEXT BOOKS

1. *Data structures, Algorithms and Applications in C*, SartajSahni, 2nd Edition, Universities Press.
2. *Data structures and Algorithms in C,* Adam Drozdek, 4th edition, Cengage learning.

REFERENCES

1. *Data structures with C*, J. Hubbard, Schaum's outlines, TMH.
2. *Data structures and Algorithms in C*, M.T. Goodrich, R. Tamassia and D. Mount, Wiley India.
3. *Data structures and Algorithm Analysis in C++*, M. A. Weiss,3rd edition, Pearson.
4. *Classic Data Structures,* D. Samanta, 2nd edition, PHI.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
3	1	-	4

(20MC9103) OPERATING SYSTEMS

COURSE OBJECTIVES

1. *To be aware of the evolution and fundamental principles of operating system, processes and their communication*
2. *To understand the various operating system components like process management, memory management and to know about file management and the distributed file system concepts in operating systems*
3. *To be aware of components of operating system with relevant case study*

COURSE OUTCOMES

1. *Able to understand the operating system components and its services*
2. *Implement the algorithms in process management and solving the issues of IPC*
3. *Able to demonstrate the mapping between the physical memory and virtual memory*
4. *Able to understand file handling concepts in OS perspective*
5. *Able to understand the protection of system.*
6. *Able to understand the operating system components and services with the recent OS*

UNIT-I

Operating System Overview: Operating Systems objectives and functions, Computer System Architecture, OS Structure, OS Operations, Evolution of Operating Systems.

Operating System Structure: OS Services - System Calls - System Programs, Operating System Design and Implementation.

UNIT-II

Process Management, CPU Scheduling and Process Coordination: The Process, Process State, Process Control Block, Threads, Process Scheduling-Scheduling Queues, Schedulers, Context Switch, Preemptive Scheduling, Dispatcher, Scheduling Criteria, Scheduling algorithms, Multiple- Processor Scheduling, Real-Time Scheduling, Thread scheduling, Case studies: Linux, Windows.

Process Coordination – Process Synchronization, The Critical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, Case Studies: Linux, Windows.

UNIT-III

Memory Management: Logical & Physical Address Space, Swapping, Contiguous Allocation, Paging, Structure of Page Table, Segmentation, Segmentation with Paging.

Virtual Memory: Demand Paging, Performance of Demanding Paging, Page Replacement Page Replacement Algorithms, Allocation of Frames, Thrashing, Case Studies: Linux, Windows.

UNIT-IV

Mass Storage Structure: Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap space Management, RAID structure, stable-storage implementation, Tertiary storage structure.

File System Interface: The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection. File System Implementation - File System Structure, File System Implementation, Allocation methods, Free-space Management, Directory Implementation, Efficiency and Performance, Case Studies: Linux, Windows.

UNIT-V

Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.

Protection: System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection. Security**:** The Security problem, Program threats, System and network threats, Cryptography as a security tool, User authentication, Implementing security defenses, Firewalling to protect systems and networks, Computer –security classifications, Case Studies: Linux, Windows.

TEXT BOOKS

1. *Operating System Principles*, Abraham Silberschatz, Peter B. Galvin and Greg Gagne, 9th Edition, Wiley student Edition.
2. *Operating Systems – Internals and Design Principles*, W. Stallings, 6th Edition, Pearson Education.

REFERENCES

1. *"Operating Systems – Internals and Design Principles",*William Stallings, 7th Edition, Prentice Hall, 2011.
2. "*Modern Operating Systems*", Andrew S. Tanenbaum, Second Edition, Addison Wesley, 2001.
3. "*Operating Systems: A Concept-Based Approach*", D M Dhamdhere, Second Edition, Tata McGraw-Hill Education, 2007.
4. "*Operating Systems: A Design-Oriented Approach*", Charles Crowley, Tata McGraw Hill Education", 1996.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
3	1	-	4

(20MC9104) OBJECT ORIENTED PROGRAMMING THROUGH C++

COURSE OBJECTIVES

1. *Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.*
2. *Understand fundamentals of object-oriented programming in C++, including defining classes, invoking methods, using class libraries, etc.*
3. *Be able to use the Turbo C++ environment to create, debug and run simple C++ programs.*

COURSE OUTCOMES

1. *Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.*
2. *Understand fundamentals of object-oriented programming in C++, including defining classes, invoking methods, using class libraries, etc.*
3. *Debugging and running the program*
4. *Have the ability to write a computer program to solve specified problems.*
5. *Able to do the I/O Operations*
6. *Able to do the C++ Inheritance & Exception Handling concepts.*

UNIT-I

Different Paradigms for Problem Solving: need for OOP paradigm, classes and instances - fundamental characteristics of OOP (Alan key) - differences between OOP and Procedure Oriented Programming.

C++ Basics: Structure of a C++ program - Data types - Declaration of variables - Expressions - Operators - Operator Precedence - Evaluation of expressions - Type conversions - Pointers - Arrays - Pointers and Arrays - Strings - Structures - References. Flow control statements- if, switch, while, for, do, break, continue, goto statements.

UNIT-II

C++ Functions: Scope of variables - Parameter passing methods - Default arguments - inline functions - Recursive functions - Pointers to functions. C++ Classes and Data Abstraction: Class definition - Class objects - Class scope - this pointer - Friends to a class - Static class members - Constant member functions - Constructors and Destructors - Data abstraction - ADT and information hiding.

Dynamic Memory: allocation and de-allocation operators-new and delete - Dynamic creation and destruction of objects - Preprocessor directives - name spaces.

UNIT-III

Polymorphism: Function overloading - Operator overloading - generic programming-necessity of templates - Function templates and class templates

Inheritance: Defining a class hierarchy - Different forms of inheritance - Defining the Base and Derived classes - Access to the base class members - Base and Derived class construction - Destructors - Virtual base class.

UNIT-IV

Virtual Functions and Run Time Polymorphism: Overriding - Static and Dynamic bindings - Base and Derived class virtual functions.

Dynamic binding through virtual functions: Virtual function call mechanism - Pure virtual functions - Abstract classes.

UNIT-V

C++ I/O: I/O using C functions - C++ Stream classes hierarchy - Stream I/O - File streams and String streams - File Operations - Overloading << and >> operators - Error handling during file operations - Formatted I/O.

Exception Handling: Benefits of exception handling - Throwing an exception - The try block - Catching an exception - Exception objects - Exception specifications - Stack unwinding -Rethrowing an exception - Catching all exceptions.

TEXT BOOKS

1. *C++ The Complete Reference*, Herbert Schildt, TMH,4th Edition.
2. *Object Oriented Programming in C++*, R.Lafore, SAMS, Pearson Education, 4th Edition,
3. *An Introduction to OOP*, T. Budd,3rd Edition, Pearson Education,2008.

REFERENCES

1. *Programming Principles and Practice Using C++*, B.Stroutstrup, Addison- Wesley, Pearson Education.
2. *Problem solving with C++*, Walter Savitch, Pearson Education,6th Edition, 2007.
3. *The Art, Philosophy and Science of OOP with C++*, R.Miller,SPD.
4. *OOP in C++*, T.Gaddis, J.Walters and G.Muganda, 3rd Edition,Wiley DreamTech Press.
5. *An Introduction to OOP in C++ with applications in Computer Graphics*, G.M.Seed,2nd Edition, Springer.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
-	-	3	1.5

(20MC9105) DATA STRUCTURES USING C LAB

COURSE OBJECTIVES

1. *To understand the basic structure concept such as Abstract Data Types, Linear and Non Linear Data structures.*
2. *To understand the behavior of data structures such as stacks, queues, trees, directories, trees, Graph and their representations.*
3. *To understand and analyze various searching and sorting algorithms.*

COURSE OUTCOMES

1. *Learn how to use data structure concepts for realistic problems.*
2. *Ability to identify appropriate data structure for solving computing problems inC language.*
3. *Ability to solve problems independently and think critically.*
4. *Able to search and sort the elements in graphs and trees.*
5. *Ability to solve linked list problems.*
6. *Ability to solve queues and hash tables.*

List of Programs

1. Write a program in C for the following sorting methods.
   i) Bubble sort.
   ii) Selection sort.
   iii) Insertion sort.
   iv) Merge sort.
   v) Quick sort.
2. Write a program in C for the following searching methods.
   i) Linear search ii) Binary search
3. Write a menu driven program that implements singly linked list operations(Create, Display, intersection, deletion, Concatenate, merge, union, Count, Copy, Reverse, Sort).
4. Write a menu driven program that implements doubly linked list operations (Create, Display, Count, Insert, Delete, Search, Copy, Reverse and Sort).
5. Write a menu driven program that implements doubly linked list operations (Create, Display, Concatenate, merge, union and intersection).
6. Write a menu driven program that implements singly circular linked list operations (Create, Display, Count, Insert, Delete, Search, Copy, Reverse and Sort).
7. Write a program in C to implement simple Stack, Queue & Circular Queue using arrays and linked list concepts.
8. Write a menu driven program in C to
   a. Create a binary search tree.
   b. Traverse the tree in Inorder, Preorder and Post Order.
   c. Search the tree for a given node and delete the node.

9. Write a program in C to implement insertion and deletion in AVL tree.
10. Write a program in C to implement Heap Sort.
11. Write a program in C to implement Dijkstra's shortest path algorithm for a directed graph.
12. Write a program in C to implement BFS using linked representation of graph.
13. Write a program in C to implement DFS using linked representation of graph.
14. Write a program in C to create a minimum spanning tree using Kreskas's algorithm.
15. Write a program in C to create a minimum spanning tree using Prim's algorithm.

REFERENCES

1. *Data Structures using C*, D. S. Malik, 2nd edition, Cengage learning.
2. *Data Structures using C*, V. Patil, Oxford University Press.
3. *Fundamentals of Data structures in C*, E. Horowitz, S. Sahni and D. Mehta, Universities Press, 2nd edition.
4. *CPlus Data Structures*, Nell Dale, Jones and Bartlett,4th edition.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
-	-	3	1.5

(20MC9106) OBJECT ORIENTED PROGRAMMING THROUGH C++ LAB

COURSE OBJECTIVES

1. *Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.*
2. *Understand fundamentals of object-oriented programming in C++, including defining classes, invoking methods, using class libraries, etc.*
3. *Be able to use the Turbo C++ environment to create, debug and run simple C++ programs.*

COURSE OUTCOMES

1. *Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.*
2. *Understand fundamentals of object-oriented programming in C++, including defining classes, invoking methods, using class libraries, etc.*
3. *Debugging and running the program*
4. *Have the ability to write a computer program to solve specified problems.*
5. *Able to do the I/O Operations*
6. *Able to do the C++ Inheritance & Exception Handling concepts.*

List of Sample Problems/Experiments:

1. Write a **C++** program to find the sum of individual digits of a positive integer.
2. A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C++ program to generate the first n terms of the sequence
3. Write a **C++** program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
4. Write **C++** programs that use both recursive and non-recursive functions
   a) To find the factorial of a given integer.
   b) To find the nth Fibonacci number.
5. Write a **C++** program that uses functions
   a) To swap two integers.
   b) To swap two characters.
6. Write a **C++** program to find both the largest and smallest number in a list of integers.
7. Write a **C++** program to sort a list of numbers in ascending order.
8. Write a C++ program that overloads the + operator and relational operators (suitable) to perform the following operations:

   a) Concatenation of two strings.
   b) Comparison of two strings.

9. Write a template based **C++** program that determines if a particular value occurs in an array of values.
10. Write a **C++** program that uses functions to perform the following operations:
    a) Insert a sub-string into the given main string from a given position.
    b) Delete n characters from a given position in a given string.
11. Write a **C++** program that uses a function to reverse the given character string in place, without any duplication of characters.
12. Write a **C++** program to count the lines, words and characters in a given text.
13. Write a **C++** program to determine if the given string is a palindrome or not.
14. Write a **C++** program to construct of pyramid of numbers.
15. Write a **C++** program which copies one file to another.
16. Write a **C++** program that counts the characters, lines and words in the text file.
17. Write a **C++** program to change a specific character in a file.
    Note: Filename, number of the byte in the file to be changed and the new character are specified on the command line.
18. Write a **C++** program that uses a function to delete all duplicate characters in the given string.
19. Write a **C++** program that uses a function to convert a number to a character string.
20. Write a **C++** program
    a) To write an object to a file. b) To read an object from the file.
21. Write **C++** programs that illustrate how the following forms of inheritance are supported:
    a) Single inheritance b) Multiple inheritance
    c) Multi level inheritance d) Hierarchical inheritance
22. Write a **C++** program that illustrates the order of execution of constructors and destructors when new class is derived from more than one base class.
23. Write a **C++** program that illustrates how run time polymorphism is achieved using virtual functions.
24. Write a **C++** program that illustrates the role of virtual base class in building class hierarchy.
25. Write a **C++** program for Factorial using recursion.

REFERENCES

1. *Mastering C++*, K.R.VenuGopal, Raj Kumar and T.Ravi Shankar, TMH.
2. *C++ Programming*, D.S.Malik, Cengage Learning.
3. *Practical C++ Programming*, S.Qualline,SPD.
4. *Object Oriented Programming with C++*,E.Balaguruswamy, 4th Edition, TMH,2008.
5. *OOP with C++*,S.Sahay,Oxford Higher Education.
6. *C++ and OOP Paradigm*,D.Jana,2nd Edition, PHI

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – I Semester

L	T	P	C
-	-	4	2

(20MC9107) P.C. SOFTWARE LAB

COURSE OBJECTIVES

1. *To impart the knowledge Hardware and software components of PC.*
2. *To impart knowledge about the various components of a computer and its internals.*
3. *To disassemble and assemble the PC back to working condition.*
4. *To understand the installation of software's.*
5. *To impart the knowledge on Software troubleshooting and Hardware troubleshooting.*
6. *To impart knowledge on MS Office tools.*

COURSE OUTCOMES

1. *Able to disassemble and assemble the PC back to working condition.*
2. *Able to know installation of softwares.*
3. *Able to understand mapping between virtual and physical memory.*
4. *Able to know Software troubleshooting and Hardware Troubleshooting.*
5. *Able to work on MS Office tools.*

PC Hardware

Exercise 1: Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.

Exercise 2: Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva. Also students need to go through the video which shows the process of assembling a PC. A video would be given as part of the course content.

Exercise 3: Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.

Exercise 4: Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both windows and Linux. Lab instructors should verify the installation and follow it up with a Viva.

Exercise 5: Hardware Troubleshooting: Students have to be given a PC which does not boot due to improper assembly or defective peripherals. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva

Exercise 6: Software Troubleshooting: Students have to be given a malfunctioning CPU due to system software problems. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.

Internet & World Wide Web

Exercise 7: **Orientation & Connectivity Boot Camp:** Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should demonstrate, to the instructor, how to access the websites and email. If there is no internet connectivity preparations need to be made by the instructors to simulate the WWW on the LAN.

Exercise 8: **Search Engines & Netiquette:** Students should know what search engines are and how to use the search engines. A few topics would be given to the students for which they need to search on Google. This should be demonstrated to the instructors by the student.

MS-Word

Exercise 9: The mentor needs to give an overview of Microsoft (MS) word 2007: Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word. Give a task covering to create project certificate. Features to be covered:-Formatting Fonts in word, Drop Cap in word, Applying Text effects, Using Character Spacing, Borders and Colors, Inserting Header and Footer, Inserting table, using Drawing toolbar in word.

MS-Excel

Exercise 10: The mentor needs to tell the importance of MS office 2007 Excel as a Spreadsheet tool covering Accessing, overview of toolbars, saving excel files, Using help and resources., Also give a task that is covering the features like Gridlines, Format Cells, Summation, auto fill, Formatting Text.

MS-Power Point

Exercise 11: Students will be working on MS power point that helps them create basic power point presentation. Topics covered during this Exercise include :- PPT orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in Power point. Students shall be given a model power point presentation which needs to be replicated (exactly how it's asked).

REFERENCES

1. *Comdex Information Technology course tool kit,* Vikas Gupta, WILEY Dream tech
2. *The Complete Computer upgrade and repair book*, 3rd edition Cheryl A Schmidt, WILEY Dreamtech
3. *Introduction to Information Technology*, ITL Education Solutions limited, Pearson Education.
4. *PC Hardware and A+ Handbook* – Kate J. Chase PHI (Microsoft)
5. *LaTeX Companion* – Leslie Lamport, PHI/Pearson.
6. *IT Essentials PC Hardware and Software Companion Guide,* Third Edition by David Anfinson and Ken Quamme. – CISCO Press, Pearson Education.
7. *IT Essentials PC Hardware and Software Labs and Study Guide, T*hird Edition by Patrick Regan – CISCO Press, Pearson Education.
8. *Troubleshooting, Maintaining and Repairing PCs*, S.J.Bigelow,5thedition,TMH

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
3	1	-	4

(20MC9108) COMPUTER NETWORKS

COURSE OBJECTIVES

1. *To understand networking concepts and basic communication model*
2. *To understand network architectures and components required for data communication.*
3. *To analyze the function and design strategy of physical, data link, network layer and transport layer*

COURSE OUTCOMES

1. *Able to trace the flow of information from one node to another node in the network*
2. *Able to Identify the components required to build different types of networks*
3. *Able to understand the functionalities needed for data communication into layers*
4. *Able to choose the required functionality at each layer for given application*
5. *Able to understand the working principles of various application protocols*
6. *Acquire knowledge about security issues and services available*

UNIT-I

Introduction: Network hardware, Reference models, Transmission media, Narrow band and Broadband ISDN.

Data Link Layer Design Issues: Error detection and correction, Elementary Data link protocols, Sliding window protocols. Medium Access Control Sublayer: Channel allocation methods, TDM, FDM, ALOHA, Carrier sense Multiple access protocols, Collision free protocols, Ethernet, Wireless LAN.

UNIT-II

Network Layer: Routing Algorithms - Shortest path, Flooding, Flow based, Distance vector, Link state, Hierarchical, Broadcast Routing.

Congestion Control Algorithms: General principals of congestion control, Congestion prevention polices, choke packets and Load shedding.

UNIT-III

Internetworking: Tunneling, Fragmentation. Network layer in the Internet-IP protocols, IP address, Subnets, Internet control protocols, OSPF, BGP, Internet multicasting & Mobile IP.

ATM Networks: cell formats, connection setup routing & switching, service categories, and quality of service, ATM LANS.

UNIT-IV

The Transport Layer: Elements of transport protocols - addressing, establishing a connection, releasing connection, flow control & buffering & crash recovery.

Internet Transport protocol: TCP & UDP.

UNIT-V

Application Layer: Name service(DNS) Domain Hierarchy - Name servers - Name resolutions - Traditional applications - SMTP - MIME - WWW-HTTP - Network Management - SNMP.

Network Security: Cryptography – Substitution Ciphers - Transposition Ciphers - Symmetric and Public Key algorithms – DES and RSA.

TEXT BOOKS

1. *Computer Networks* - Andrew S Tanenbaum, 4th Edition.Pearson Education.
2. *Data Communications & Networking*, 4th Edition, Behrouz A. Forouzan, McGraw-Hill.

REFERENCES

1. *An Engineering Approach to Computer Networks*-S.Keshav, 2nd Edition, Pearson Education.
2. *Data and Computer Communication,* by William Stallings. 8e, 2003, PEA.
3. *Understanding communications and Networks*, 3rd Edition, W.A.Shay, Cengage Learning.
4. *Computer and Communication Networks,* Nader F. Mir, Person Education.
5. *Computer Networking: A Top-Down Approach Featuring the Internet*, James F.Kurose, K.W.Ross, 3rd Edition, Pearson Education.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
3	1	-	4

(20MC9109) JAVA PROGRAMMING

COURSE OBJECTIVES

1. *Understand fundamentals of programming such as variables, conditional and iterative execution, methods etc.*
2. *Understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.*
3. *Be able to use the Java SDK environment to create, debug and run simple Java programs.*

COURSE OUTCOMES

Students who have completed this course able to:

1. *Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.*
2. *Understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.*
3. *Have the ability to write a computer program to solve specified problems.*
4. *Able to do the java collection framework programs*
5. *Able to develop Applet programs.*
6. *Work with GUI, Event handling mechanism.*

UNIT-I

Java Basics: History of Java - Java buzzwords - comments - data types - variables - constants - scope and life time of variables - operators - operator hierarchy - expressions - type conversion and casting - enumerated types - control flow-block scope - conditional statements - loops, break and continue statements - simple java program - arrays - input and output - formatting output.

OOP concepts: encapsulation, inheritance, polymorphism, classes, objects, constructors, methods, parameter passing, static fields and methods, access control - this reference - overloading methods and constructors - recursion - garbage collection - building strings - exploring string class - Enumerations - auto boxing and unboxing - Generics.

UNIT-II

Inheritance: Inheritance concept - benefits of inheritance - Super classes and Sub classes - Member access rules - Inheritance hierarchies - super uses - preventing inheritance: final classes and methods - casting - polymorphism- dynamic binding - method overriding - abstract classes and methods - the Object class and its methods.

Interfaces: Interfaces vs. Abstract classes - defining an interface - implementing interfaces - accessing implementations through interface references - extending interface. Packages: Defining - Creating and Accessing a Package - Understanding CLASSPATH - importing packages.

UNIT-III

Data Structures Creation and Manipulation in Java: Introduction to Java Collections - Overview of Java Collection frame work - Commonly used Collection classes – ArrayList,

LinkedList, HashSet, HashMap andTreeMap, Collection Interfaces – Collection - Set, List and Map, Legacy Collection classes – Vector, Hashtable, Stack, Dictionary(abstract), Enumeration interface - Iteration over Collections – Iterator interface -ListIterator interface. Other Utility classes – StringTokenizer - Formatter - Random - Scanner - Observable - Using java.util.

Files: streams- byte streams - character streams - text Input/output - binary input/output - random access file operations - File management using File class - Using java.io.Networking in Java: Introduction - Manipulating URLs - Ex. Client/Server Interaction with Stream Socket Connections - Connectionless Client/Server Interaction with Datagrams - Using java.net.

UNIT-IV

Exception Handling: Dealing with errors - benefits of exception handling - the classification of exceptions- exception hierarchy - checked exceptions and unchecked exceptions - usage of try, catch, throw, throws and finally, rethrowing exceptions - exception specification - built in exceptions - creating own exception sub classes - Guide lines for proper use of exceptions.

Multithreading: Differences between multiple processes and multiple threads - thread states - creating threads - interrupting threads - thread priorities - synchronizing threads -interthread communication - thread groups - daemon threads.

UNIT-V

GUI Programming with Java: The AWT class hierarchy - Introduction to Swing - Swing vs. AWT -MVC architecture - Hierarchy for Swing components - Containers – Top-level containers – JFrame, JApplet, JWindow, JDialog, Light weight containers – JPanel, A simple swing application - Overview of several swing components- Java's Graphics capabilities – Introduction - Graphics contexts and Graphics objects - color control - Font control - Drawing lines, rectangles and ovals, Drawing arcs - Layout management - Layout manager types - border, grid, flow, box.

Event Handling: Events - Event sources - Event classes - Event Listeners - Relationship between Event sources and Listeners - Delegation event model - Semantic and Low-level events - examples: handling a button click - handling mouse and keyboard events - Adapter classes. Applets: Differences between applets and applications - life cycle of an applet - Four methods of an applet - Developing applets and testing - passing parameters to applets - applet security issues.

TEXT BOOKS

1. *Java: The complete reference,* Herbert Schildt, 7th edition,TMH.
2. *Core Java, Volume 1-Fundamentals,* Cay S.Horstmann and Gary Cornell, eighth edition,Pearson education

REFERENCES

1. *An introduction to Java programming and object oriented application development*, R.A. Johnson- Cengage Learning.
2. *An introduction to programming and OO design using Java*, J.Nino, F.A.Hosch, John Wiley & Sons.
3. *Java for Programmers,*P.J.Deitel and H.M.Deitel, Pearson education / Java: How to Program P.J.Deitel and H.M.Deitel ,8th edition, PHI
4. *Object Oriented Programming with Java*, B.Eswara Reddy, T.V.Suresh Kumar, P.Raghavan, Pearson-Sanguine

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
3	1	-	4

(20MC9110) COMPUTER GRAPHICS

COURSE OBJECTIVES

1. *To understand computational development of graphics with mathematics*
2. *To provide in-depth knowledge of transformation shape modeling of 2D and 3D application.*
3. *To provide in-depth knowledge of display systems, image synthesis, of 2D and 3D application.*

COURSE OUTCOMES

1. *Gain proficiency in 3D computer graphics API programming*
2. *Able to draw line, circle and ellipse using algorithms*
3. *Enhance the perspective of modern computer system with modeling, analysis and interpretation of 2D and 3D visual information.*
4. *Able to transform the 2D and 3D objects*
5. *Able to apply methods for detecting visual surface.*
6. *Able to design Animation for the objects.*

UNIT-I

Introduction: Application areas of Computer Graphics - overview of graphics systems - video-display devices - raster scan systems - random scan systems - graphics monitors and work stations and input devices.

Output Primitives: Points and lines - line drawing algorithms - mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm - boundary-fill and flood-fill algorithms.

UNIT-II

2-D Geometrical Transforms: Translation - scaling - rotation - reflection and shear transformations - matrix representations and homogeneous coordinates - composite transforms - transformations between coordinate systems.

3-D Geometric Transformations: Translation - rotation - scaling - reflection and shear transformations - composite transformations -

UNIT-III

2-D Viewing: The viewing pipeline - viewing coordinate reference frame - window to view-port coordinate transformation - viewing functions - Cohen-Sutherland and Liang -Barsky line clipping algorithms - Sutherland –Hodgeman polygon clipping algorithm

3-D Viewing: Viewing pipeline - viewing coordinates - view volume and general projection transforms and clipping3-D Object representation: Polygon surfaces - quadric surfaces -spline representation -Hermite curve - Bezier curve and B-spline curves - Bezier and B-spline surfaces. Basic illumination models - polygon rendering methods

UNIT-IV

Visible Surface Detection Methods: Classification - back-face detection - depth-buffer - A buffer method - scan-line - depth sorting - BSP-tree methods - area sub-division and Octree Methods - Ray-casting Method - Curved Surfaces - Curved- Surface Representations - Surface Contour Plots - Wireframe Methods - Visibility- Detection Functions.

Illumination Models and Surface-Rendering Methods: Light Source – Basic Illumination Models – Halftone Patterns and Dithering Techniques.

UNIT-V

Color Model and Color Applications: Properties of Light – Standard Primaries – Intuitive Color Concepts – RGB Color Model – YIQ Color Model – CMY Color Model – HSV Color Model.

Computer Animation: Design of animation sequence - general computer animation functions - raster animation - computer animation languages - key frame systems - motion specifications.

TEXT BOOKS

1. *Computer Graphics C version* - Donald Hearn and M. Pauline Baker - Pearson education.
2. *Computer Graphics Principles & practice* - Foley -VanDam -Feiner and Hughes - second edition - Pearson Education.

REFERENCES

1. *Computer Graphics* -Zhigandxiang - Roy Plastock -Schaum's outlines -*Second edition* - Tata McGraw hill edition.
2. *Procedural elements for Computer Graphics* - David F Rogers - Tata McGraw hill - 2nd edition.
3. *Principles of Interactive Computer Graphics* -Neuman and Sproul - TMH.
4. *Principles of Computer Graphics* -Shalini -Govil-Pai - Springer.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
3	1	-	4

(20MC9111) DATABASE MANAGEMENT SYSTEM

COURSE OBJECTIVES

1. *To understand the fundamentals of data models and conceptualize and depict a database system using ER diagram*
2. *To make a study of SQL and relational database design.*
3. *To know about data storage techniques a query processing.*

COURSE OUTCOMES

1. *Understand the basic concepts of the database and data models.*
2. *Design a database using ER diagrams and map ER into Relations and normalize the relations.*
3. *Acquire the knowledge of query evaluation to monitor the performance of the DBMS.*
4. *Ability to execute various SQL Commands.*
5. *Develop a simple database applications using normalization.*
6. *Acquire the knowledge about different special purpose databases and to critique how they differ from traditional database systems.*

UNIT-I

Introduction: Introduction to database systems; Database - DBMS distinction - approaches to building a database - data models - database management system - three- schema architecture of a database - challenges in building a DBMS - various components of a DBMS.

ER Model: Conceptual data modeling - motivation - entities - entity types - various types of attributes - relationships - relationship types - E/R diagram notation - examples.

UNIT-II

Relational Data Model - Concept of relations - schema-instance distinction - keys - referential integrity and foreign keys.

Relational Algebra Operators: selection - projection - cross product - various types of joins - division - example queries - tuple relation calculus - domain relational calculus - converting the database specification in E/R notation to the relational schema.

UNIT-III

SQL: Introduction - data definition in SQL – table - key and foreign key definitions -update behaviors.

Querying in SQL: Basic select-from-where block and its semantics. nested queries - correlated and uncorrelated - notion of aggregation - aggregation functions group by and having clauses - embedded SQL.

UNIT-IV

Dependencies and Normal forms: Importance of a good schema design - problems encountered with bad schema designs - motivation for normal forms - dependency theory - functional dependencies - Armstrong's axioms for FD's - closure of a set of FD's - minimal covers.

Normal Forms: definitions of 1NF, 2NF, 3NF and BCNF, decompositions and desirable properties of them - algorithms for 3NF and BCNF normalization - multi-valued dependencies and 4NF - join dependencies and definition of 5NF.

UNIT-V

Data Storage and Indexes: file organizations - primary - secondary index structures - various index structures - hash-based - dynamic hashing techniques - multi-level indexes - B+ trees.

Transaction processing and Error recovery: concepts of transaction processing - ACID properties - concurrency control - locking based protocols for CC - error recovery and logging - undo - redo - undo-redo logging and recovery methods.

TEXT BOOKS

1. *Database management Systems*, Peter Rob, A.AnandaRao and Carlos Coronel, Cengage Learning.
2. *Fundamentals of Database Systems*, Elmasri, Navate, Pearson Education.

REFERENCES

1. *Introduction to Database Systems*, C.J. Date Pearson Education.
2. *Oracle for Professionals*, The X Team, S.Shah and V.Shah, SPD.
3. *Database Management Systems*, Raghurama Krishnan, Johannes Gehrke, TATA McGraw Hill 3rd Edition.
4. *Database System Concepts*, Silberschatz, Korth, McGraw Hill, V Edition.
5. *Database Systems using Oracle*: A simplified guide to SQL and PL/SQL, Shah, PHI.
6. *Fundamentals of Database Management Systems*, M.L. Gillenson, Wiley Student Edition.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
3	1	-	4

(20MC9112) SOFTWARE ENGINEERING

COURSE OBJECTIVES

1. *To provide an insight into the processes of software development*
2. *To understand and practice the various fields such as analysis, design, development, testing of Software Engineering*
3. *To develop skills to construct software of high quality with high reliability*
4. *To apply metrics and testing techniques to evaluate the software*

COURSE OUTCOMES

1. *Get an insight into the processes of software development*
2. *Able to understand the problem domain for developing SRS and various models of software engineering*
3. *Able to Model software projects into high level design using DFD diagrams*
4. *Able to Measure the product and process performance using various metrics*
5. *Able to Evaluate the system with various testing techniques and strategies*
6. *Able to apply metrics to evaluate the software*

UNIT-I

Software, Software Engineering, and Process: The nature of Software - The changing nature of software - Software engineering- software process and software engineering practices and principles - Generic process model (framework) - Process patterns - Process assessment and improvement - CMMI - Software myths.

Process Models: Prescriptive process models: The waterfall model - Incremental process models - Evolutionary process models; Personal and Team process models.The Unified process - Aspect oriented software development - Agile development: Agile process - Extreme programming.

UNIT-II

Software Requirements: Introduction to functional and non-functional requirements - Requirements engineering - groundwork analysis - Eliciting requirements - developing usecases. Requirements modeling - Requirements validation - Software requirements specification(SRS) - Requirements management.

Requirements Modeling: Scenario based - class based - Web/Mobile App based modeling. Software Project Estimation: Empirical estimation models.

UNIT-III

Design Concepts: Software design quality guidelines and attributes - Design concepts - Design model. Architecture Design: Architecture and its importance - Architectural Styles - Data design - Architectural design. Component-level Design: Component - Designing Class based components - Component-level design for web and mobile apps - component based-development.

User Interface Design: Golden rules - User interface analysis and design - interface analysis - interface design steps. Pattern Based Design: Design patterns - Pattern based software design - Architectural patterns - Component level design patterns - User interface design patterns. Web app design: Interface design - Content design - Navigation design. Mobile app Design: Developing mobile app - design practices.

UNIT-IV

Testing: Software testing strategies: A strategic approach to software testing - Test strategies for conventional - object oriented software - web app - mobile app; Validation testing - System testing - The art of debugging.

Testing Conventional Applications: Software testing fundamentals - White-Box testing: Basis path testing - condition (predicate) testing - data flow testing - loop testing - Black box testing: Equivalence partitioning - Boundary value analysis - Graph based testing methods. Testing Object Oriented Applications: OO testing methods - Testing methods applicable at class level - Interclass test case design. Testing Web Applications and Mobile Applications - Security Engineering and risk analysis - Security assurance.

UNIT-V

Umbrella Activities: Software Quality and achieving it - Measurement and metrics: Size oriented metrics - Function oriented metrics - Metrics for software quality - Product metrics: Metrics for the requirements model - Metrics for the design model - Metrics for source code - Metrics for testing - Metrics for maintenance.

Software Reengineering: A software reengineering process model - Software reengineering activities - Reverse engineering.

TEXT BOOKS

1. *Software Engineering: A practioner's approach* by Roger S. Pressman and Bruce R. Maxim.
2. *Software Engineering*- Sommerville, 8th edition, Pearson education.

REFERENCES

1. *Software Engineering*- K.K. Agarwal&YogeshSingh,New Age International Publishers.
2. *Software Engineering, an Engineering approach*- James F. Peters, WitoldPedrycz, John Wiely.
3. *Systems Analysis and Design*- ShelyCashman Rosenblatt, Thomson Publications.
4. *Software Engineering principles and practice*- Waman S Jawadekar, The McGraw- Hill Companies.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
-	-	3	1.5

(20MC9113) JAVA PROGRAMMING LAB

COURSE OBJECTIVES

1. *To understand Java and object-oriented concepts.*
2. *To understand decision and iteration control structures to implement algorithms*
3. *To implement Java collection framework as programming techniques.*

COURSE OUTCOMES

After completion of this course, the students would be able to

1. *Understand programming language concepts, particularly Java and object-oriented concepts, data types.*
2. *Effectively create and use objects from predefined class libraries*
3. *Apply decision and iteration control structures to implement algorithms*
4. *Implement interfaces, inheritance, and polymorphism as programming techniques.*
5. *Able to write recursive algorithms.*
6. *Implement Java collection frame work as programming techniques.*

List of Sample Programs/Experiments

1. a) Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer
   b) Write a Java program to multiply two given matrices.
   c) Write a Java program to find the roots of quadratic equation.
   d) Write a Java program to generate first *n* Fibonacci numbers.
2. Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use StringTokenizer class of java.util)
3. Write a Java program to find both the largest and smallest number in a list of integers.
4. Write a Java program to illustrate method overloading.
5. Write a Java program to sort a list of names in ascending order.
6. Write a Java program to implement the matrix ADT using a class. The operations supported by this ADT are:
   a) Reading a matrix. b) Printing a matrix.
   c) Addition of matrices. d) Subtraction of matrices.
   e) Multiplication of matrices.
7. Write a Java Program to solve Towers of Hanoi problem .
8. Write a Java Program that uses a recursive function to compute ncr. (Note: n and r values are given.)
9. Write a Java program to perform the following operations:
   a) Concatenation of two strings.
   b) Comparison of two strings.
10. Implement the complex number ADT in Java using a class. The complex ADT is used to represent complex numbers of the form c=a+b, where a and b are real numbers. The operations supported by this ADT are:
    a) Reading a complex number. d) Subtraction of complex numbers.

b) Writing a complex number. e) Multiplication of complex numbers.
c) Addition of Complex numbers. f) Division of complex numbers.

11. Write a Java program that makes frequency count of letters in a given text.
12. Write a Java program that uses functions to perform the following operations :
    a) Inserting a sub-string in to the given main string from a given position.
    b) Deleting n characters from a given position in a given string.
13. Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome.
14. Write a Java program to make frequency count of words in a given text.
15. Write a Java program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
16. Write a Java program that reads a file and displays the file on the screen, with a line number before each line.
17. Write a Java program that displays the number of characters, lines and words in a text file.
18. Write a Java program to change a specific character in a file.
    Note: Filename, number of the byte in the file to be changed and the new character are specified on the command line.
19. Write a Java program that:
    i) Implements stack ADT.
    ii) Converts infix expression into Postfix form
    iii) Evaluates the postfix expression.
20. Develop an applet in Java that displays a simple message.
21. Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked.
22. 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.
23. Write a Java program for handling mouse events.
24. 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.
25. Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
26. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.
27. Write a Java program that implements a simple client/server application. The client sends data to a server. The server receives the data, uses it to produce a result, and then sends the result back to the client. The client displays the result on the console. For ex: The data sent from the client is the radius of a circle, and the result produced by the server is the area of the circle. (Use java.net)
28. 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.

29. Write a Java program that allows the user to draw lines, rectangles and ovals.
30. Write a Java program to create an abstract class named Shape that contains an empty method named numberOfSides ( ).Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides ( ) that shows the number of sides in the given geometrical figures.
31. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Jtable component.
32. Write a Java program that illustrates the following
    a) Creation of simple package.b) Accessing a package. c) Implementing interfaces.
33. Write Java programs that illustrates the following
    a) Handling predefined exceptions
    b) Handling user defined exceptions
34. Write Java programs that use both recursive and non-recursive functions for implementing the following searching methods:
    a) Linear search b) Binary search
35. Write Java programs to implement the List ADT using arrays and linked lists
36. Write Java program to implement the Queue ADT using arrays and linked lists.
37. Write a Java program for handling Key events.
38. Write a Java program that uses both stack and queue to test whether the given string is a palindrome.
39. Write Java programs to implement the following using a singly linked list.
    a) Stack ADT b) Queue ADT
40. Write Java programs for implementing the following sorting methods:
    a) Bubble sort d) Quick sort b) Selection sort c) Insertion sort

REFERENCES

1. *An introduction to Java programming and object oriented application development*, R.A. Johnson- Cengage Learning.
2. *An introduction to programming and OO design using Java*, J.Nino, F.A.Hosch, John Wiley & Sons.
3. *Java for Programmers,*P.J.Deitel and H.M.Deitel, Pearson education / Java: How to Program P.J.Deitel and H.M.Deitel ,8th edition, PHI
4. *Object Oriented Programming with Java*, B.Eswara Reddy, T.V.Suresh Kumar, P.Raghavan, Pearson-Sanguine

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
-	-	3	1.5

(20MC9114) COMPUTER GRAPHICS LAB

COURSE OBJECTIVES

1. *To understand computational development of graphics with mathematics*
2. *To provide in-depth knowledge of transformation shape modeling of 2D and 3D application.*
3. *To provide in-depth knowledge of display systems, image synthesis,of 2D and 3D application.*

COURSE OUTCOMES

1. *Gain proficiency in 3D computer graphics API programming*
2. *Able to draw line, circle and ellipse using algorithms*
3. *Enhance the perspective of modern computer system with modeling, analysis and interpretation of 2D and 3D visual information.*
4. *Able to transform the 2D and 3D objects*
5. *Able to apply methods for detecting visual surface.*
6. *Able to design Animation for the objects.*

List of Programs

1. Write a program to demonstrate Line Digital Differential Analyzer Algorithm
2. Write a program to demonstrate Bresenham's Line Drawing Algorithm
3. Write a program to demonstrate Midpoint Circle Generation Algorithm
4. Write a program to demonstrate Ellipse Generation Algorithm
5. Write a program to rotate the text in various angles?
6. Write a program to Draw Rectangle from (100,200) pixel to (400,500) pixel.
7. Write a program to implement Basic 2D Transformations for translation on a rectangle.
8. Write a program to implement Basic 2D Transformations for Rotation on a rectangle.
9. Write a program to implement Basic 2D Transformations for Scaling on a rectangle.
10. Write a program to implement Boundary Fill algorithm.
11. Write a program to implement Flood Fill algorithm.
12. Write a program for Line Clipping.

REFERENCES

1. *Computer Graphics* -Zhigandxiang - Roy Plastock -Schaum's outlines -*Second edition* - Tata McGraw hill edition.
2. *Procedural elements for Computer Graphics* - David F Rogers - Tata McGraw hill - 2nd edition.
3. *Principles of Interactive Computer Graphics* -Neuman and Sproul - TMH.
4. *Principles of Computer Graphics* -Shalini -Govil-Pai - Springer.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
-	-	4	2

(20MC9115) DATABASE MANAGEMENT SYSTEM LAB

COURSE OBJECTIVES

1. *The objective of the course is to enable students to understand and use a relational database system.*
2. *Introduction to Databases, Conceptual design using ERD, Functional dependencies and Normalization, Relational Algebra are covered in detail.*
3. *Students learn how to design and create a good database and use various SQL operations.*

COURSE OUTCOMES

1. *Able to master the basic concepts and understand the applications of database systems.*
2. *Able to construct an Entity-Relationship (E-R) model andRelational Algebra.*
3. *Understand and apply database normalization principles.*
4. *Able to construct SQL queries to perform CRUD operations on database. (Create, Retrieve, Update, Delete).*
5. *Understand the usage of triggers.*
6. *Able to execute the pl/sql programs.*

List of Sample Programs/Experiments

1. Create table using various data types, modifying structure of the table (adding, removing columns and changing column name), inserting rows into the table and display rows using SELECT command.
2. Create table using necessary constraints, - adding, altering, dropping of constraints and inserting rows into a table - examples using SELECT command.
3. Queries using Relational, Logical Operators, IN, NOT IN, LIKE, NOT LIKE, BETWEEN, NOT BETWEEN etc.
4. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints.
   Example:- Select the roll number and name of the student who secured fourth rank in the class.
5. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
6. Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)
7. i) Creation of simple PL/SQL program which includes declaration section, executable section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
   ii) Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.

8. Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.
9. Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE-APPLICATION ERROR.
10. Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.
11. Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.
12. Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.
13. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
14. Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

REFERENCES

1. *SQL & PL/SQL for Oracle 10g*, Black Book, Dr.P.S. Deshpande.
2. *Introduction to SQL*, Rick F.VanderLans, Pearson Education.
3. *Oracle PL/SQL Programming*,Steven Feuerstein,SPD.
4. *The Database Book*, N.Gehani, Universities Press.
5. *Database Systems using Oracle: A Simplified Guide to SQL and PL/SQL*, Shah, PHI.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA I Year – II Semester

L	T	P	C
-	-	2	-

(20HS0863) PROFESSIONAL COMMUNICATION SKILLS LAB

COURSE OBJECTIVES

The objectives of this course:

1. *To improve the students' fluency in English, through a well-developed vocabulary*
2. *To enable them listening spoken English at normal conversational speed by English speakers*
3. *To respond appropriately in different social-cultural and professional contexts*
4. *To develop drafting skills among the students.*
5. *To develop Inter-personal and Intra-personal Skills*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Flair in Writing and felicity in written expression*
2. *Enhancing job required skills for getting success in their professions*
3. *Improving Effective Speaking Abilities for their business or professional correspondence*
4. *prepare effective Interview techniques to get job in the present scenario*
5. *Using the appropriate skills in all kinds of professional activities*
6. *Use effective communicative approaches by preparing job application, report and other kinds of spoken and written correspondences.*

UNIT I
COMMUNICATIVE COMPETENCY

1. Reading Comprehension
2. Listening Comprehension
3. Vocabulary for competitive purpose
4. Spotting Errors

UNIT II
TECHNICAL WRITING

1. Report writing
2. Curriculum vitae
3. Cover Letter
4. E-mail writing

UNIT III
PRESENTATIONAL SKILLS

1. Oral presentation
2. Power point presentation
3. Poster presentation

4. Stage Dynamics

UNIT IV
CORPORATE SKILLS

1. Dress code
2. Telephonic skills
3. Net-etiquettes

UNIT V
GETTING READY FOR JOB

1. Group Discussion
2. Interview skills
3. Psychometric test

Minimum Requirements for English For Corporate Communications:

1. Computer Assisted Language Learning (CALL) Lab: The Computer Aided Language Lab for 60 Students with 60 systems one Master Console, LAN facility and English Language Software for self-study by learners.
2. Professional Communication Skills Lab with movable chairs and audio visual aids with a P. A. system, Projector, a Digital stereo audio & video system and Camcorder etc.

System Requirement (Hardware component):

Computer network with: LAN with minimum 60 multimedia systems with the following.

Specifications:

i)
   a) P- IV Processor
   b) Speed 3.10 GHZ
   c) RAM – 4 GB
   d) Hard Disk – 320 GB

ii) Headphones of High quality

Software:

1. Walden Info Tech Software.

References:

1. Effective Tech Communication, Rizvi, Tata McGraw – Hill Education, 2007.
2. Communication skills, Sanjay Kumar &Pushpalatha, Oxford University Press, 2012.
3. Writing Tutor. Advanced English Learners' Dictionary, 9th Edition, Oxford Universty Press, 2015.
4. Powerful Vocabulary Builder, AnjanaAgarwal, New Age International Publishers, 2011.
5. Listening Extra, Miles Craven, Cambridge University Press, 2008.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9116) WEB TECHNOLOGIES

COURSE OBJECTIVES

1. *To understand the basics of HTML5 and CSS*
2. *To understand the web server and the server-side programming*
3. *To understand the concepts of PHP and AJAX*

COURSE OUTCOMES

Student is able to:

1. *Design and work with HTML5 and CSS applications.*
2. *Usage of javascript functions and objects.*
3. *Do the server side programming, maintain sessions.*
4. *Establish the DB connections and access the data.*
5. *Ability to work on Java Server Page*
6. *Design pages using PHP and AJAX.*

UNIT-I

HTML5: Essential HTML –Working with Text, List and Links, Images, Tables, and Frames. Introduction to Forms and HTML controls.

CSS: Introduction – working with Style Sheets – Formatting text with styles – Layout with styles.

UNIT-II

JavaScript: Features – Using JavaScript in an HTML Document – JavaScript Data types – Variables – Operators – Control Flow statements.

JavaScript Functions and Objects: Working with functions – Working with events – JavaScript Objects – Built-in objects String, Number, Array, Math and Date

UNIT-III

Servlets: Introduction to Servlets – Lifecycle of a Servlet – A simple servlet - The javax.servlet Package - Reading Servlet parameters -The javax.servlet.httppackage - Handling Http Request & Responses - Using Cookies-Session Tracking.

Java Server Page: Understanding JSP – JSP Life Cycle – Creating simple JSP pages – Working with JSP basic tags and Implicit Objects.

UNIT-IV

PHP – Introduction - Data types - Flow of Control statements.

PHP Functions: Functions - Arrays - Forms - Handling Files - Error handling.

UNIT-V

AJAX – Introduction to AJAX -XMLHttp - Request - Response – Events.

AJAX Database: Database Connection - XML - PHP using AJAX.

TEXT BOOKS

1. *The complete Reference Java*, Herbert Schildt. TMH.
2. *HTML5 Black Book*, DT Editorial Services, 2nd edition.
3. *HTML5 and CSS3*, Elizabeth Castro and Bruce Hyslop, 7th edition.
4. *Ajax: A beginner's guide*,stevenHolzner.

REFERENCES

1. *Learn JavaScript and Ajax with w3schools*, Jan Egil, Stale, Kai Jim, and Hege
2. *Beginning Web Programming*, Jon Duckett, WROX.
3. *http://www.w3schools.com/php/default.asp*
4. *Beginning PHP*, Matt Doyle

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9117) MOBILE APPLICATION DEVELOPMENT

COURSE OBJECTIVES

1. *Describe those aspects of mobile programming that make it unique from programming for other platforms*
2. *Critique mobile applications on their design pros and cons*
3. *Utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces*

COURSE OUTCOMES

1. *Students understood the aspects of mobile programming that make it unique from programming for other platforms.*
2. *Students can design and develop sophisticated mobile interfaces.*
3. *Students program mobile applications for the Android operating system by use basic features.*
4. *Able to work on advanced phone features.*
5. *Students understood sending and receiving messages.*
6. *Ability to deploy applications to the Android marketplace for distribution.*

UNIT-I

The Android Platform: Introduction to the Android platform - Required tools - Setting up environment - creating Hello world Application;

Understanding Activities: Linking activities using Intents - Fragments.

UNIT-II

Getting to Know the Android User Interface: Understanding the Components of a Screen - Views and ViewGroups, LinearLayout, AbsoluteLayout, TableLayout, RelativeLayout, FrameLayout and ScrollView.

Other Screen Orientation: Adapting to Display Orientation - Managing Changes to Screen Orientation - Utilizing the Action Bar.

UNIT-III

Designing User Interface with Views: Basic views - Picker Views - List views;

Pictures: Pictures and menus with views.

UNIT-IV

Data Persistence: Saving and loading user preferences - Persisting data to files - creating and using databases.

Content Providers: Sharing Data in Android - Using a Content Provider -Creating Own Content Providers.

UNIT-V

Messaging: Sending SMS Messages Programmatically - Getting Feedback after Sending a message - Sending SMS Messages Using Intent - Receiving SMS Messages - Caveats and Warnings.

Developing and Publishing Android Applications: Creating Your Own Services - Establishing Communication between a Service and an Activity - Binding Activities to Services - Understanding Threading**;** Preparing for Publishing - Deploying APK Files.

TEXTBOOKS

1. *Beginning Android 4 Application Development* by Wei-Meng Lee -Wrox Publications.
2. *Android: A programmer's guide* by Jerome Dimarzio -McGrawHill

REFERENCES

1. *Introduction to android application development* -Addision Wesley - Joseph Annuzzi Jr. - Lauren Darcey - Shane Conder - 4th edition.
2. *Lean Mobile App Development Paperback,*Mike van Drongelen, Aravind Krishnaswamy.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9118) DATA WAREHOUSING AND DATA MINING

COURSE OBJECTIVES

1. *To expose the students to the concepts of Data warehousing Architecture and implementation*
2. *To learn to use association rule mining for handling large data*
3. *To understand the concept of classification for the retrieval purposes*

COURSE OUTCOMES

Upon Completion of the course, the students will be able to

1. *Preprocess the data for mining applications*
2. *Apply the association rules for mining the data*
3. *Cluster the high dimensional data for better organization of the data*
4. *Able to apply the data mining for text and multimedia.*
5. *Evolve Multidimensional Intelligent model from typical system*
6. *Evaluate various mining techniques on complex data objects*

UNIT-I

Introduction: Fundamentals of data mining - Data Mining Functionalities - Classification of Data Mining systems - Data Mining Task Primitives - Integration of a Data Mining System with a Database or a Data Warehouse System - Major issues in Data Mining.

Data Preprocessing: Need for Preprocessing the Data - Data Cleaning - Data Integration and Transformation - Data Reduction - Discretization and Concept Hierarchy Generation.

UNIT-II

Data Warehouse and OLAP Technology for Data Mining: Data Warehouse - Multidimensional Data Model - Data Warehouse Architecture - Data Warehouse Implementation - Further Development of Data Cube Technology - From Data Warehousing to Data Mining.

Data Cube Computation and Data Generalization: Efficient Methods for Data Cube Computation - Further Development of Data Cube and OLAP Technology - Attribute-Oriented Induction.

UNIT-III

Mining Frequent Patterns, Associations and Correlations: Basic Concepts - Efficient and Scalable Frequent Itemset Mining Methods - Mining various kinds of Association Rules - From Association Mining to Correlation Analysis - Constraint-Based Association Mining

Classification and Prediction: Issues Regarding Classification and Prediction - Classification by Decision Tree Induction - Bayesian Classification - Rule-Based Classification - Classification by Back propagation - Support Vector Machines – Associative Classification - Lazy Learners - Other Classification Methods - Prediction - Accuracy and Error measures - Evaluating the accuracy of a Classifier or a Predictor - Ensemble Methods.

UNIT-IV

Cluster Analysis Introduction: Types of Data in Cluster Analysis - A Categorization of Major Clustering Methods - Partitioning Methods - Hierarchical Methods - Density-BasedMethods - Grid-Based Methods - Model-Based Clustering Methods - Clustering High-Dimensional Data - Constraint-Based Cluster Analysis - Outlier Analysis

Mining Streams: Time Series and Sequence Data: Mining Data Streams - Mining Time – Series Data - Mining Sequence Patterns in Transactional Databases - Mining Sequence Patterns in Biological Data - Graph Mining - Social Network Analysis and Multirelational Data Mining.

UNIT-V

Mining Object, Spatial, Multimedia, Text and Web Data: Multidimensional Analysis and Descriptive Mining of Complex Data Objects - Spatial Data Mining - Multimedia Data Mining - Text Mining - Mining the World Wide Web.

Applications and Trends in Data Mining: Data Mining Applications - Data Mining System Products and Research Prototypes - Additional Themes on Data Mining and Social Impacts of Data Mining.

TEXT BOOKS

1. *Data Mining – Concepts and Techniques*, Jiawei Han & MichelineKamber, Morgan Kaufmann Publishers, 2nd Edition, 2006.
2. *Introduction to Data Mining*, Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Pearson education.

REFERENCES

1. *Data Warehousing in the Real World* – Sam Aanhory& Dennis Murray Pearson Edn Asia.
2. *Insight into Data Mining*,K.P.Soman,S.Diwakar, V.Ajay,PHI,2008.
3. *Data Warehousing Fundamentals* – PaulrajPonnaiah Wiley student Edition
4. *The Data Warehouse Life cycle Tool kit* – Ralph Kimball Wiley student edition
5. *Building the Data Warehouse*By William H Inmon, John Wiley & Sons Inc, 2005.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9119) SOFTWARE TESTING
(PROGRAMME ELECTIVE – I)

COURSE OBJECTIVES

1. *To know the behavior of the testing techniques to detect the errors in the software*
2. *To understand standard principles to check the occurrence of defects and its removal.*
3. *To understand the models of software reliability.*

COURSE OUTCOMES

1. *Test the software by applying testing techniques to deliver a product free from bugs*
2. *Evaluate the web applications using bug tracking tools.*
3. *Investigate the scenario and the able to select the proper testing technique*
4. *Explore the test automation concepts and tools*
5. *Deliver quality product to the clients by way of applying standards such as TQM, Six Sigma*
6. *Evaluate the estimation of cost, schedule based on standard metrics*

UNIT-I

Introduction: Purpose of testing - Dichotomies - model for testing - consequences of bugs - taxonomy of bugs.

Flow Graphs and Path Testing: Basics concepts of path testing - predicates, path predicates and Achievable paths - path sensitizing - path instrumentation - application of path testing.

UNIT-II

Transaction Flow Testing: Transaction flows - transaction flow testing techniques.

Dataflow Testing: Basics of dataflow testing - strategies in dataflow testing - application of dataflow testing.

UNIT-III

Domain Testing: Domains and paths - Nice & ugly domains - domain testing.

Interfaces Testing: Interfaces testing - domain and interface testing - domains and testability.

UNIT-IV

Paths, Path Products and Regular Expressions: Path products & path expression – reduction procedure - applications - regular expressions & flow anomaly detection. Logic.

Based Testing: Overview - decision tables - path expressions -kv charts - specifications.

UNIT-V

State, State Graphs and Transition Testing: State graphs - good & bad state graphs - state testing - Testability tips.

Graph Matrices and Application: Motivational overview - matrix of graph - relations - power of a matrix - node reduction algorithm - building tools. (Student should be given an exposure to a tool like JMeter or Win-runner).

TEXT BOOKS

1. *Software testing techniques* - BarisBeizer, DreamTech, second edition.
2. *Software Testing Tools* – Dr.K.V.K.K.Prasad, DreamTech.

REFERENCES

1. *The craft of software testing* - Brian Marick, Pearson Education.
2. *Software Testing*, 3rd edition, P.C.Jorgensen, Aurbach Publications (Dist.by SPD).
3. *Software Testing in the Real World* – Edward Kit, Pearson.
4. *Effective methods of Software Testing*, Perry, John Wiley, 2nd Edition, 1999.
5. *Art of Software Testing* – Meyers, John Wiley

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9120) ARTIFICIAL INTELLIGENCE
(PROGRAMME ELECTIVE – I)

COURSE OBJECTIVES

1. *To understand the searching strategies in AI.*
2. *To understand the Game Playing, Knowledge Based techniques.*
3. *To understand what Expert System*

COURSE OUTCOMES

At the end of this course:

1. *Student should have a knowledge and understanding of the basic conepts of AI including Search.*
2. *Student can able to solve optimization problems.*
3. *Student can solve the Game Playing problems.*
4. *Student can able to use to planning and learning techniques*
5. *Student should be able to use this knowledge and understanding of appropriate principles and guidelines to synthesise solutions to tasks in AI and to critically evaluate alternatives.*
6. *Student can have ability to use the expert system*

UNIT-I

Introduction: AI-Acting and thinking humanly - rationally - Searching: Searching for solutions - Uniformed Search Strategies - Informed Search Strategies - Heuristic Functions.

Local Search Algorithms and Optimization Problems: Hill-climbing - Simulated annealing - Local beam - Genetic algorithms - Constraint Satisfaction Problems - Backtracking Search for CSPs.

UNIT-II

Adversial Search: Games - Optimal Decision in Games - Alpha-Beta Pruning - Evaluation Functions - Cutting off search - Games that include an Element of chance - Game programs. Knowledge and reasoning-I: Logical Agents.

Knowledge and reasoning-II: First-Order Logic: Syntax and Semantics - Using First Order Logic - Knowledge Engineering - Inference in First-Order Logic: Propositional vs. First-Order Inference - Unification and Lifting - Resolution - Forward and Backward Chaining.

UNIT-III

Planning: Classical planning problem - Language of planning problems - Expressiveness and extension - planning with state-space search - Partial-Order planning - Planning Graphs - Planning with Propositional Logic.

Learning: Forms of learning - Introduction learning - Learning Decision Tree - Statistical learning methods - learning with complete data - learning with hidden variables-EM Algorithms - Instance based learning - Neural networks.

UNIT-IV

Expert Systems: Introduction - Advantages - Characteristics - General concepts - Applications and Domains - Languages - Shells and Tools – Elements.

Production Systems: Procedural and Nonprocedural Paradigms - Artificial Neural Systems - Connectionist Expert Systems and Inductive Learning.

UNIT-V

Design of Expert Systems: Selecting the Appropriate Problem - Stages in the Development of an Expert System - The Expert System Life Cycle.

Detailed life cycle model: Expert system design examples-Certainty factors - Decision trees - backward chaining.

TEXT BOOKS

1. *Artificical Intelligence-A Modern Approach*, Russell, Norvig, 2e, 2004, PEA
2. *Expert Systems-Principles and Programming*, Giarratano, Riley, 3e,2003, Thomson

REFERENCES

1. *Artificial Intelligence-Structures and strategies for Complex problem Solving*, George F Luger, 4e, 2004, PEA.
2. *Artificial Intelligence*, Rich, Knight, Nair, 3e, TMH.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9121) DISTRIBUTED SYSTEMS
(PROGRAMME ELECTIVE – I)

COURSE OBJECTIVES

1. *To recognize how the principles are applied in contemporary distributed systems, explain how they affect the software design, and be able to identify features and design decisions that may cause problems;*
2. *To design a distributed system that fulfills requirements with regards to key distributed systems properties (such as scalability, transparency, etc.), be able to recognize when this is not possible, and explain why;*
3. *To build distributed system software using basic OS mechanisms as well as higher - level middleware and languages*

COURSE OUTCOMES

After completion of this course, the student is:

1. *Able to explain what a distributed system is, why you would design a system as a distributed system, and what the desired properties of such systems are;*
2. *Able to solve inter process communication problems.*
3. *Able to list the principles underlying the functioning of distributed systems, describe the problems and challenges associated with these principles, and evaluate the effectiveness and shortcomings of their solutions;*
4. *Able to recognize how the principles are applied in contemporary distributed systems, explain how they affect the software design, and be able to identify features and design decisions that may cause problems;*
5. *Able to design a distributed system that fulfills requirements with regards to key distributed systems properties (such as scalability, transparency, etc.), be able to recognize when this is not possible, and explain why*
6. *Able to build distributed software and higher – level middleware and languages.*

UNIT-I

Introduction: Characterization of Distributed Systems - System models.

Networking and Internetworking: Introduction – Types of network - Network principles - Internet protocols - Case study (Ethernet).

UNIT-II

Inter Process Communication: The API for Internet protocols - External data representation and marshaling - Multicast communication - overlay networks. Remote Invocation – Request replay protocols - Remote Procedure call - Remote method Invocation.

Case Study: Java Remote Method Invocation.

UNIT-III

Operating System Support: The operating system layer - Protection - Process and threads - Communication and invocation - Operating system architecture - Distributed Objects – Distributed Objects.

Case Study: CORBA

UNIT-IV

Security: Overview of security techniques - Cryptographic algorithms - Digital signatures.

Distributed File Systems: File service architecture - Sun Network File System - The Andrew File System.Name Services: Name services and the Domain Name System - Directory services.

UNIT-V

Time and Global states: Clock - events and process states - Synchronizing physical clocks - Logical time and logical clocks - Transactions and Concurrency control - Transactions - Nested Transactions - Locks.

Distributed Transactions: Flat and nested distributed transactions - Atomic commit protocols - Concurrency control in distributed transactions - distributed Deadlocks. Overview of Distributed Operating systems - Mach - Chorus.

TEXT BOOKS

1. *Distributed Systems Concepts and Design*, G Coulouris, J. Dollimore and T. Kindberg, Fourth Edition, Pearson Education.
2. *Distributed systems: principles and paradigms*, I Andrew S.Tanenbaum, Maarten Van Steen, 2nd Edition.

REFERENCES

1. *Distributed Operating Systems*, PradeepK.Sinha, PHI.
2. *Advanced Concepts in Operating Systems*, M Singhal, N G Shivarathri, and Tata McGraw-Hill Edition.
3. *Distributed Systems*, S.Ghosh, Chapman&Hall/CRC, Taylor&Francis Group, 2010.
4. *Distributed Systems* – Principles and Paradigms, A.S. Tanenbaum and M.V. Steen, Pearson Education.
5. *Distributed Algorithms*, N.A.Lynch, and Elsevier

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9122) LINUX PROGRAMMING
(PROGRAMME ELECTIVE – I)

COURSE OBJECTIVES

1. *To understand the LINUX system structure.*
2. *To understand and use command line shell.*
3. *To develop the basic skills required to write network programs using Sockets.*

COURSE OUTCOMES

1. *Work confidently in Linux environment.*
2. *Work with different text processing commands*
3. *Work with files and directories in linux.*
4. *Work with shell script to automate different tasks as Linux administration.*
5. *Able to communicate with processes and memory.*
6. *Work confidently with the socket.*

UNIT-I

Linux Utilities: File handling utilities - Security by file permissions - Process utilities - Disk utilities - Networking commands - Filters - Text processing utilities and Backup utilities -SEd - AWK.

Working with the BASH: Introduction - shell responsibilities - pipes and input Redirection - output redirection - here documents - running a shell script - the shell as a programming language - shell meta characters - file name substitution - shell variables - command substitution - shell commands - the environment - quoting - test command - control structures - arithmetic in shell - shell script examples - interrupt processing - functions - debugging shell scripts.

UNIT-II

Linux Files and Directories: File Concept - File System Structure - Inodes - File types - The standard I/O (fopen, fclose, fflush, fseek, fgetc, getc, getchar, fputc, putc, putchar, fgets, gets etc.), formatted I/O, stream errors - kernel support for files - System calls - library functions - file descriptors - low level file access - usage of open, creat, read, write, close, lseek, stat family, umask, dup, dup2, fcntl - file and record locking.

File And Directory Management: chmod, chown, links(soft links & hard links - unlink, link, symlink), mkdir, rmdir, chdir, getcwd - Scanning Directories-opendir, readdir, closedir,rewinddir, seekdir, telldir functions.

UNIT-III

Linux Process: Process concept - Kernel support for process - process attributes - process hierarchy - processstates - process composition - process control - process creation - waiting for a process - process termination - zombie process -orphan process - system call interface for process management-fork, vfork, exit, wait, waitpid - exec family, system.

Linux Signals: Introduction to signals - Signal generation and handling - Kernel support for signals - Signal function - unreliable signals - reliable signals, kill, raise, alarm, pause, abort, sleep functions.

UNIT-IV

Interprocess Communication: Introduction to IPC - IPC between processes on a single computer system - IPC between processes on different systems - pipes - FIFOs - Introduction to three types of IPC (Linux)-message queues, semaphores and shared memory.

Message Queues: Kernel support for messages - Linux APIs for messages - client/server example. Semaphores: Kernel support for semaphores - Linux APIs for semaphores - file locking with semaphores. Shared Memory: Kernel support for shared memory - Linux APIs for shared memory - semaphore and shared memory example.

UNIT-V

Multithreaded Programming: Differences between threads and processes - Thread structure and uses - Threads and Lightweight Processes - POSIX Thread APIs - Creating Threads - Thread Attributes - Thread Synchronization with semaphores and with Mutexes - Example programs.

Sockets: Introduction to Linux Sockets - Socket system calls for connection oriented protocol and connectionless protocol -example-client/server programs.

TEXT BOOKS

1. *Unix Concepts and Applications*, Sumitabha Das, TMH,2006.
2. *Beginning Linux Programming* by Neil Matthew, Richard Stones

REFERENCES

1. *A Linux and UNIX System Programming Handbook* by Michael Kerrisk
2. *Linux System Programming*, Robert Love, O'Reilly, SPD.
3. *Advanced Programming in the Unix environment*, 2nd Edition, W.R.Stevens, Pearson Education.
4. *System Programming with C and Unix*,A.Hoover,Pearson.
5. *Unix and Shell programming*, B.A.Forouzan and R.F.Gilberg, Cengage Learning.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9123) NETWORK SECURITY
(PROGRAMME ELECTIVE – I)

COURSE OBJECTIVES

1. *This Course focuses towards the introduction of network security using various cryptographic algorithms.*
2. *Underlying network security applications.*
3. *It also focuses on the practical applications that have been implemented and are in use to provide email and web security.*

COURSE OUTCOMES

1. *Understand the most common type of cryptographic algorithm*
2. *Understand the Public-Key Infrastructure*
3. *Understand network security in transport layer.*
4. *Understand security protocols for protecting data on networks*
5. *Understand vulnerability assessments and the weakness of using passwords for authentication*
6. *Be able to configure simple firewall architectures*

UNIT – I

Introduction: Computer Security Concepts - the OSI Security Architecture - Security Attacks - Security Services - Security Mechanisms - A Model for Network Security.

Symmetric Encryption and Message Confidentiality: Symmetric Encryption Principles - Symmetric Block Encryption Algorithms - Random and Pseudorandom Numbers - Stream Ciphers and RC4 - Cipher Block Modes of Operation.

UNIT – II

Public-Key Cryptography and Message Authentication: Approaches to Message Authentication - Secure Hash Functions - Message Authentication Codes - Public-Key Cryptography Principles - Public-Key Cryptography Algorithms - Digital Signatures.

Key Distribution and User Authentication: Symmetric Key Distribution Using Symmetric Encryption - Kerberos - Key Distribution Using Asymmetric Encryption - X.509 Certificates - Public-Key Infrastructure - Federated Identity Management.

UNIT – III

Transport-Level Security: Web Security Considerations - Secure Socket Layer and Transport Layer Security - Transport Layer Security - HTTPS - Secure Shell (SSH).

Wireless Network Security: IEEE 802.11 Wireless LAN Overview - IEEE 802.11i Wireless LAN Security - Wireless Application Protocol Overview - Wireless Transport Layer Security - WAP End-to-End Security.

UNIT – IV

Electronic Mail Security: Pretty Good Privacy - S/MIME -DomainKeys Identified Mail.

IP Security: IP Security Overview -IP Security Policy -Encapsulating Security Payload - Combining Security Associations - Internet Key Exchange - Cryptographic Suites.

UNIT – V

System Security Intruders: Intruders - Intrusion Detection - Password Management.

Malicious Software: Types of Malicious Software -Viruses -Virus Countermeasures - Worms -Distributed Denial of Service Attacks. Firewalls:The Need for Firewalls - Firewall Characteristics - Types of Firewalls -Firewall Basing -Firewall Location and Configurations.

TEXT BOOKS

1. *Network Security Essentials: Applications and Standards*,William Stallings, Fourth Edition.
2. *Introduction to Cryptography with coding theory*, Wade Trappe, Lawrence C Washington, Pearson.

REFERENCES

1. *Cryptography and Network Security - Principles and Practice,* William Stallings.
2. *Network Security,*M. V. Arun Kumar.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9124) HUMAN COMPUTER INTERACTION
(PROGRAMME ELECTIVE – II)

COURSE OBJECTIVES

1. *Gain an overview of Human-Computer Interaction (HCI), with an understanding of user interface design in general, and alternatives to traditional "keyboard and mouse" computing*
2. *Become familiar with the vocabulary associated with sensory and cognitive systems as relevant to task performance by humans*
3. *Be familiar with a variety of both conventional and non-traditional user interface paradigms*

COURSE OUTCOMES

1. *Find innovative ways of interacting with computers*
2. *Help the disabled by designing non-traditional ways of interacting*
3. Able to use system menus and windows.
4. *Use cognitive psychology in the design of devices for interaction*
5. *Able to apply models from cognitive psychology.*
6. *Able to perform effectively on menus and windows*

UNIT- I

Introduction: Importance of user Interface – definition - importance of good design - Benefits of good design - A brief history of Screen design.

The graphical user interface – popularity of graphics - the concept of direct manipulation - graphical system – Characteristics. Web user Interface - popularity - characteristics- Principles of user interface.

UNIT- II

Design process – Understanding how people interact with computers - importance of human characteristics human consideration - Human interaction speeds - understanding business functions.

Screen Designing: Design goals – Screen meaning and purpose - organizing screen elements - ordering of screen data and content – screen navigation and flow – Visually pleasing composition – amount of information – focus and emphasis – presentation information simply and meaningfully – information retrieval on web – statistical graphics – Technological consideration in interface design.

UNIT- III

System Menus: Structures of Menus - Functions of Menus - Content of Menus - Kinds of Graphical menus

Windows: Window characteristics - Components of a window - Window presentation styles - Types of windows - Windom management

UNIT- IV

Select the Proper Interaction Devices: Input Device - Characteristics of Input Device - Selecting the proper input device – Output Devices.

Choose the Proper Screen-Based Control: Operable controls - Text Entry/Read-only controls - Selection controls - Combination Entry/selection controls - Selecting the proper controls.

UNIT- V

Graphics: Icons - Multimedia - Color-what is it - Color uses - Color and Human vision - Choosing colors.

Testing: The purpose and importance of usability testing - Scope of testing - Prototypes - Kinds of Tests - Developing and conducting the test

TEXT BOOKS

1. *The essential guide to user interface design*, Wilbert O Galitz, Wiley, 2nd edition, 2013.
2. *Human –Computer Interaction - D.R.Olsen - Cengage Learning.*

REFERENCES

1. *Designing the user interface* - Ben Shneidermann - 3rd Edition - Pearson Education Asia.
2. *Human – Computer Interaction* -I.Scott Mackenzie - Elsevier Publishers.
3. *Interaction Design* -Prece - Rogers - Sharps - Wiley Dreamtech.
4. *User Interface Design* -SorenLauesen - Pearson Education.
5. *Human –Computer Interaction* - Smith - Atakan -Cengage Learning.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9125) ADVANCED PROGRAMMING (PYTHON & R LANGUAGES)
(PROGRAMME ELECTIVE – II)

COURSE OBJECTIVES
1. *Exposure to various problem solving approaches of computer science*
2. *Understand the fundamentals of 'R' programming*
3. *Explore data-sets to create testable hypotheses and identify appropriate statistical tests.*

COURSE OUTCOMES
At the end of the course the student will be able to:
1. *Making Software easily right out of the box.*
2. *Experience with an interpreted Language.*
3. *Prior Introduction to testing software*
4. *Ability to Work on a real life Project, implementing R Analytics to create Business Insights.*
5. *Ability to analyze the data and results using R, a flexible and completely Cross - platform.*
6. *Ability to use a wide range of analytical methods and produce presentation quality graphics*.

UNIT – I
Introduction: History of Python - Need of Python Programming - Applications - Basics of Python - Programming Using the REPL(Shell) - Running Python Scripts - Variables - Assignment - Keywords - Input-Output - Indentation.

Types, Operators and Expressions: Types - Integers - Strings - Booleans; Operators- Arithmetic Operators - Comparison (Relational) Operators - Assignment Operators - Logical Operators - Bitwise Operators - Membership Operators - Identity Operators, Expressions and order of evaluations - Control Flow- if, if-elif-else, for, while, break, continue and pass

UNIT – II
Data Structures: Lists - Operations, Slicing - Methods - Tuples - Sets - Dictionaries - Sequences - Comprehensions.

Functions: Defining Functions - Calling Functions - Passing Arguments - Keyword Arguments - Default Arguments - Variable-length arguments - Anonymous Functions – Fruitful Functions (Function Returning Values) - Scope of the Variables in a Function - Global and Local Variables. Modules: Creating modules - import statement - from Import statement - name spacing. Python packages: Introduction to PIP - Installing Packages via PIP - Using Python Packages

UNIT – III
Object Oriented Programming OOP in Python: Classes - 'self variable' - Methods - Constructor Method - Inheritance - Overriding Methods - Data hiding.

Error and Exceptions: Difference between an error and Exception - Handling Exception - try except block - Raising Exceptions - User Defined Exceptions.

UNIT – IV

Introducing R: Getting the Hand of R - Running the R Program - Finding Your Way with R - Command Packages. BECOMING FAMILIAR WITH R: Reading and Getting Data into R - Viewing Named Objects - Types of Data Items - The Structure of Data Items - Examining Data Structure Working with History Commands - Saving your Work in R.

Working with Objects: Manipulating Objects - Viewing Objects within Objects - Constructing Data Objects - Forms of Data Objects: Testing and Converting.

UNIT – V

Data: Descriptive statistics and tabulation. Distribution - Looking at the Distribution of Data. Simple Hypothesis Testing - Using the Student's t-test, The Wilcoxon U-Test (Mann-Whitney), Paired t- and U-Tests, Correlation and Covariance - Tests for Association.

Introduction To Graphical Analysis: Box-whisker Plots, Scatter Plots, Pairs Plots(Multiple Correlation Plots) Line Charts, Pie Charts, Clevel and Dot Charts, Bar Charts, Copy Graphics to Other Applications. Adding elements to existing plots - Matrix plots - multiple plots in one window - exporting graphs. Writing Your Own Scripts: Beginning to program Copy and Paste Scripts - Creating Simple Functions - Making Source Code.

TEXT BOOKS

1. *Python Programming: A Modern Approach*, VamsiKurama, Pearson
2. *Learning Python*, Mark Lutz, Orielly
3. *"Beginning R the statistical programming language"*, Dr. Mark Gardener, Wiley Publications, 2015.

REFERENCES

1. *Think Python*, Allen Downey, Green Tea Press
2. *Core Python Programming*, W.Chun, Pearson.
3. *Introduction to Python*, Kenneth A. Lambert, Cengage
4. *Hands-On Programming with R Paperback*,Grolemund (Author), Garrett (Author),SPD, 2014. 2. The R Book, Michael J. Crawley, WILEY, 2012.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9126) INTERNET OF THINGS
(PROGRAMME ELECTIVE – II)

COURSE OBJECTIVES

1. *Makes clear view over physical computing, ubiquitous computing, or the Internet of Things, it's a hot topic in technology.*
2. *It discusses Prototyping on Embedded and Physical Devices.*
3. *It discusses design concepts that will make IOT products eye-catching and appealing.*

COURSE OUTCOMES

1. *Ability to combine sensors, servos, robotics, Arduino chips, and more with various or the Internet,*
2. *Ability to create interactive, cutting-edge devices.*
3. *Better idea of the overview of necessary steps to take the idea of IOT concept through production*
4. *Ability to apply techniques for writing Embedded Code*
5. *Ability to manage memory and performance of battery life*
6. *Ability to design a printed circuit boards.*

UNIT – I

Introduction - Internet of Things – Design Principles for Connected Devices – Web Thinking for Connected Devices.

Internet Principles: IP – TCP – IP Protocol Suite – UDP – IP Address – MAC Address – TCP and UDP Ports – Application Layer Protocols.

UNIT – II

Prototyping – Prototypes and Production – Cloud – Open Source vs Closed Source – Tapping into the Community.

Prototyping Embedded Devices: Electronics – Embedded Computing Basics – Ardunio – Raspberry Pi – Beagle Bone Black – Electronic Imp.

UNIT – III

Prototyping the Physical Design: Laser Cutting – 3D Printing – CNC Milling – Repurposing and Recycling.

Prototyping Online Components: New API – Real Time Reactions – Other Protocols.

UNIT – IV

Techniques for Writing Embedded Code – Memory Management – Performance and Battery life – Libraries – Debugging.

Business Models: Models – Funding an Internet of Things Startup.

UNIT – V

Moving to Manufacture: Designing Kits – Designing Printed Circuit Boards – Manufacturing Printed Circuit Boards – Mass Producing the case and other Fixtures – Scaling up Software.

Ethics: Characterizing the Internet of Things – Control – Environment – Solutions

TEXT BOOKS

1. *Designing The Internet of Things*, Adrian Mcewen and HakinCassimally, Wiley Publications , 2015
2. *Internet of Things (A Hands-on-Approach)*, Vijay Madisetti and ArshdeepBahga, 1st Edition, VPT, 2014.

REFERENCES

1. *Rethinking the Internet of Things: A Scalable Approach to Connecting Everything*, Francis daCosta, 1st Edition, Apress Publications, 2013
2. *Getting Started with the Internet of Things*,CunoPfister, O‟Reilly Media, 2011

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9127) E-COMMERCE
(PROGRAMME ELECTIVE – II)

COURSE OBJECTIVES

1. *The fundamental principles of e-Business and e- Commerce and the role of Management,*
2. *The underlying used technologies with emphasis on Internet Technologies*
3. *The application of tools and services to the development of small scale e-Commerce applications.*

COURSE OUTCOMES

1. *Recognize the impact of Information and Communication technologies, especially of the Internet in business operations*
2. *Recognize the fundamental principles of e-Business and e-Commerce*
3. *Distinguish the role of Management in the context of e-Business and e-Commerce*
4. *Able to manage electronic payments*
5. *They can know the added value, risks and barriers in the adoption of e-Business and e-Commerce*
6. *Examine applications of e-Commerce in relation to the applied strategic.*

UNIT – I

Electronic Commerce: Electronic Commerce Framework - Electronic Commerce and Media Convergence - The Anatomy of E-Commerce Application - Electronic Commerce Organization Applications

The Network Infrastructure for Electronic Commerce: Market Forces Influencing the I-Way - Components of the I Way - Network Access Equipment - the Last Mlle: Local Roads and Access Ramps - Global Information Distribution: Networks: Public Policy Issues Shaping the I-Way.

UNIT – II

The Internet as a Network Infrastructure: The Internet Terminology - Chronological History of the Internet NSFNET: Architecture and Components: Globalization of the Academic Internet -

Internet Governance: The Internet Society –An Overview of Internet Applications – Electronic Commerce - World Wide Web(WWW) as the Architecture: Web Background: Hypertext Publishing - Technology behind the Web: Security and the Web-Consumer-Oriented Electronic Commerce: Oriented Applications - Mercantile Process Models Mercantile Models from the Consumer's Perspective - Mercantile Models from the Merchant's Perspective. Case study: E-Commerce/High Security (Pci)

UNIT – III

Electronic Payment Systems: Types of Electronic Payment Systems - Smart Cards and Electronic Payment Systems - Credit Card-Based Electronic Payment systems: Risk and

Electronic Payment Systems Designing Electronic Payment systems – Inter organizational Commerce

EDI: Legal - security, and Privacy Issues:EDI and Electronic Commerce – EDI Implementation - MIME, and Value- Added Networks : Standardization and EDI -EDI Software Implementation: EDI Envelope for Message Transport: Value- Added Networks (VANs) - Internet – Based EDI.Case study: Social Media Marketing

UNIT – IV

Intra organization Electronic Commerce: Internal Information System: Macro forces and Internal Commerce - Work-Flow Automation and Coordination - Customization and Internal Commerce -

Supply Chain Management (SCM): The Corporate Digital Library: Dimensions of Internal Electronic Commerce Systems - Making a Business Case for a Document Library - Types of Digital Document Library - Types of Digital Documents - Issues behind Document Infrastructure - Corporate Data Warehouses.Case study: Email Marketing - Email Personalization

UNIT – V

M-Commerce: Introduction to Mobile Commerce - Limitations - history - applications - architecture - transaction models - payment methods - advantages – disadvantages.

Mobile app marketing Case Study: O2 Priority Moments gets small businesses on side.

TEXT BOOKS

1. *Frontiers of Electronic Commerce*,Ravi Kalakota and Andrew B.Whinston.Pearson Education.
2. *ECommerce*,Henry Chan, Raymond Lee. TharanDillan and E.Chany,Wiley,2003.

REFERENCES

1. *Web Commrece Technology*, DanjelMinoli and EmunaMimoli, Tata MicGraw Hill, 1999.
2. *A Electronic Commerce*, Marilyn Greenstein and Todd M Feinman, TaraMcGraw Hill Edition.
3. *M-Commerce: Book Your Business with the Power of Mobile Commerce*, Paul

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	1	-	4

(20MC9128) BLOCK CHAIN TECHNOLOGY
(PROGRAMME ELECTIVE – II)

COURSE OBJECTIVE

By the end of the course, students will be able to

1. *Understand how blockchain systems (mainly Bitcoin and Ethereum) work,*
2. *To securely interact with them,*
3. *Design, build, and deploy smart contracts and distributed applications,*
4. *Integrate ideas from blockchain technology into their own projects.*

COURSE OUTCOMES

1. *Explain design principles of Bitcoin and Ethereum.*
2. *Explain the Simplified Payment Verification protocol.*
3. *List and describe differences between proof-of-work and proof-of-stake consensus.*
4. *Interact with a blockchain system by sending and reading transactions.*
5. *Design, build, and deploy a distributed application.*
6. *Evaluate security, privacy, and efficiency of a given blockchain system.*

UNIT – I

Basics: Distributed Database, Two General Problem, Byzantine General problem and Fault Tolerance, Hadoop Distributed File System, Distributed Hash Table, ASIC resistance, Turing Complete.

Cryptography: Hash function, Digital Signature - ECDSA, Memory Hard Algorithm, Zero Knowledge Proof.

UNIT – II

Blockchain: Introduction, Peer to Peer, Cryptographically Secure, Append Only, Updateable via Consensus, blockchain working, accumulation of blocks, pros and cons of blockchain, tiers of blockchain technology, features of blockchain.

Types of Blockchain: Distributed Ledger, Public Blockchains, Private Blockchains, Semiprivate Blockchains, Sidechains, Permissioned Ledger, Shared Ledger, Fully Private and Proprietary Blockchains, Tokenized Blockchains, Tokenless Blockchains.

UNIT – III

Consensus: Mechanism, Types of Consensus Mechanism, Consensus in Blockchain.

Decentralization: Disintermediation and Contest Driven Decentralization, Routes to Decentralization, Full Ecosystem Decentralization, Smart Contracts, Decentralized Organizations, Platforms for Decentralization.

UNIT-IV

Symmetric Cryptography: Symmetric Cryptography, Security Characteristics.

Cryptographic Primitives: Stream Ciphers, Block Ciphers, Data Encryption Standards, Advanced Encryption Standards, Public and Private Key.

UNIT-V

Asymmetric Cryptography: Digital Signature and Digital Certification, Encryption and Decryption using RSA Algorithm.

General Attributes: General Attributes, Economics, Sales, Counterparty, Trade Life Cycle, Order Anticipators, Market Manipulation.

TEXT BOOKS:

1. *Mastering Blockchain*, Imran Bashir, 2nd Edition, Packt Publishing, 2018.
2. *Blockchain*, Alan Wright Google Books, 2017.Daniel Dresher.
3. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Princeton University Press (July 19, 2016).
4. *Blockchain Basics*, Apress, 2017.

REFERENCES:

1. *Data Mining Applications for Empowering Knowledge Societies*, Hakikur Rahman, Information Science Reference, 2009.
2. *Principles of Data Mining*, David Hand, Heikki Mannila and Padhraic Smith, MIT Press, 2001.
3. *Data Mining : A Knowledge Discovery Approach*, Krzysztof J Cios, Witold Pedrycz, Roman W Swiniarski and Lukasz A Kurgan Springer, 2007
4. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*, Antonopoulos,

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
-	-	4	2

(20MC9129) WEB TECHNOLOGIES LAB

COURSE OBJECTIVES

1. *To understand the basics of HTML5 and CSS*
2. *To understand the web server and the server-side programming*
3. *To understand the concepts of PHP and AJAX*

COURSE OUTCOMES

Student is able to:

1. *Design and work with HTML5 and CSS applications.*
2. *Usage of javascript functions and objects.*
3. *Do the server side programming, maintain sessions.*
4. *Establish the DB connections and access the data.*
5. *Ability to work on Java Server Page*
6. *Design pages using PHP and AJAX.*

List of Programs/Experiments

1. Develop static pages (using only HTML) of an online Book store. The pages should resemble: www.amazon.com. The website should consist the following pages.
    - Home page
    - Registration and user Login
    - User profile page
    - Books catalog
    - Shopping cart
    - Payment by credit card Order Conformation
2. Install TOMCAT web server and APACHE. While installation assign valid port number. Make sure that these ports are available i.e., no other process is using this port.
3. Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).
4. Write a servlet program to display “hello world”.
5. A servlet program to read the parameters and the initialization parameters.
6. Write a Javascript program to Calculate squares and cubes of the numbers from 0 to 10.
7. Validate the Registration and user login pages using JavaScript.
8. Install a database. A servlet program to access the details of the table in the database. (Insert, Display, Update)
9. A JSP program to display the current system date and time.
10. A JSP program to access the details of the table in the database. (Insert, Display, Update)
11. A PHP program to display the form on the page.
12. A PHP program to access the database.

13. AJAX program to Change the content of the web page
14. AJAX program to retrieve data from XML
15. AJAX program to retrieve from the database.

REFERENCES

1. Learn JavaScript and Ajax with w3schools by Jan Egil, Stale, Kai Jim, and Hege
2. http://www.w3schools.com/php/default.asp
3. Beginning PHP by Matt Doyle

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
-	-	4	2

(20MC9130) MOBILE APPLICATION DEVELOPMENT LAB

COURSE OBJECTIVES

1. *Describe those aspects of mobile programming that make it unique from programming for other platforms*
2. *Critique mobile applications on their design pros and cons*
3. *Utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces*

COURSE OUTCOMES

1. *Students understood the aspects of mobile programming that make it unique from programming for other platforms.*
2. *Students can design and develop sophisticated mobile interfaces.*
3. *Students program mobile applications for the Android operating system by use basic features.*
4. *Able to work on advanced phone features.*
5. *Students understood sending and receiving messages.*
6. *Ability to deploy applications to the Android marketplace for distribution.*

List of Programs/Experiments

1. Set up the Android Application development environment.
2. Develop the ‘Hello world’ application.
3. Create an application with layouts.
4. Create an application using form controls.
5. Create an application to add two numbers and display the result.
6. Create an application that displays image on the screen.
7. Create an application with animation
8. Create an application with multiple activities and a simple menu using list view.
9. Create activities for menu items and parsing XML files.

REFRENCES

1. *Beginning Android 4 Application Development by Wei-Meng Lee, Wrox Publications.*
2. *Android: A programmer’s guide* by Jerome Dimarzio, McGrawHill.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
-	-	4	2

(20MC9131) DATA WAREHOUSING AND DATA MINING LAB

COURSE OBJECTIVES

This Lab course is intended to

1. *Introduce data mining techniques including predictive, descriptive and visualization modeling and their effective use in discovering interesting hidden patterns in large volume of data generated by businesses, science, web, and other sources.*
2. *Focus is on the main process of data mining such as data preparation, classification, clustering, association analysis, and pattern evaluation*

COURSE OUTCOMES

After undergoing the course students will be able to

1. *Synthesize the data mining fundamental concepts and techniques from multiple perspectives.*
2. *Develop skills and apply data mining tools for solving practical problems*
3. *Advance relevant programming skills.*
4. *Gain experience and develop research skills by reading the data mining literature.*

Task 1: Credit Risk Assessment
Description:

The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide whether the credit of a customer is good, or bad. A bank's business rules regarding loans must consider two opposing factors. On the one hand, a bank wants to make as many loans as possible. Interest on these loans is the banks profit source. On the other hand, a bank cannot afford to make too many bad loans. Too many bad loans could lead to the collapse of the bank. The bank's loan policy must involve a compromise: not too strict, and not too lenient.

To do the assignment, you first and foremost need some knowledge about the world of credit.

You can acquire such knowledge in a number of ways.

1. Knowledge Engineering. Find a loan officer who is willing to talk. Interview her and try to represent her knowledge in the form of production rules.
2. Books. Find some training manuals for loan officers or perhaps a suitable textbook on finance. Translate this knowledge from text form to production rule form.
3. Common sense. Imagine yourself as a loan officer and make up reasonable rules which can be used to judge the credit worthiness of a loan applicant.
4. Case histories. Find records of actual cases where competent loan officers correctly judged when, and when not to, approve a loan application.

The German Credit Data:

Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such dataset, consisting of 1000 actual cases collected in Germany. Credit dataset (original) Excel spreadsheet version of the German credit data.
In spite of the fact that the data is German, you should probably make use of it for this assignment. (Unless you really can consult a real loan officer !)

A few notes on the German dataset

- DM stands for Deutsche Mark, the unit of currency, worth about 90 cents Canadian (but looks and acts like a quarter).
- owns_telephone. German phone rates are much higher than in Canada so fewer people own telephones.
- Foreign_worker. There are millions of these in Germany (many from Turrkey). It is very hard to get German citizenship if you were not born of German parents.
- There are 20 attributes used in judging a loan applicant. The goal is the classify the applicant into one of two categories, good or bad.

Subtasks: (Turn in your answers to the following tasks)

1. List all the categorical (or nominal) attributes and the real-valued attributes seperately.
2. What attributes do you think might be crucial in making the credit assessement ? Come up with some simple rules in plain English using your selected attributes.
3. One type of model that you can create is a Decision Tree - train a Decision Tree using the complete dataset as the training data. Report the model obtained after training.
4. Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly? (This is also called testing on the training set) Why do you think you cannot get 100 % training accuracy?
5. Is testing on the training set as you did above a good idea? Why or Why not?
6. One approach for solving the problem encountered in the previous question is using cross-validation? Describe what cross-validation is briefly. Train a Decision Tree again using cross-validation and report your results. Does your accuracy increase/decrease? Why?
7. Check to see if the data shows a bias against "foreign workers" (attribute 20), or "personal-status" (attribute 9). One way to do this (perhaps rather simple minded) is to remove these attributes from the dataset and see if the decision tree created in those cases is significantly different from the full dataset case which you have already done. To remove an attribute you can use the preprocess tab in Weka's GUI Explorer. Did removing these attributes have any significant effect? Discuss.
8. Another question might be, do you really need to input so many attributes to get good results? Maybe only a few would do. For example, you could try just having attributes 2, 3, 5, 7, 10, 17 (and 21, the class attribute (naturally)). Try out some combinations. (You had removed two attributes in problem 7. Remember to reload the arff data file to get all the attributes initially before you start selecting the ones you want.)
9. Sometimes, the cost of rejecting an applicant who actually has a good credit (case 1) might be higher than accepting an applicant who has bad credit (case 2). Instead of counting the misclassifications equally in both cases, give a higher cost to the first case (say cost 5) and lower cost to the second case. You can do this by using a cost matrix in Weka. Train your Decision Tree again and report the Decision Tree and cross-validation results. Are they significantly different from results obtained in problem 6 (using equal cost)?

10. Do you think it is a good idea to prefer simple decision trees instead of having long complex decision trees? How does the complexity of a Decision Tree relate to the bias of the model?
11. You can make your Decision Trees simpler by pruning the nodes. One approach is to use Reduced Error Pruning - Explain this idea briefly. Try reduced error pruning for training your Decision Trees using cross-validation (you can do this in Weka) and report the Decision Tree you obtain? Also, report your accuracy using the pruned model. Does your accuracy increase?
12. (Extra Credit): How can you convert a Decision Trees into "if-then-else rules". Make up your own small Decision Tree consisting of 2-3 levels and convert it into a set of rules. There also exist different classifiers that output the model in the form of rules - one such classifier in Weka is rules. PART, train this model and report the set of rules obtained. Sometimes just one attribute can be good enough in making the decision, yes, just one! Can you predict what attribute that might be in this dataset? OneR classifier uses a single attribute to make decisions (it chooses the attribute based on minimum error). Report the rule obtained by training a one R classifier. Rank the performance of j48, PART and oneR.

Task Resources:

- Mentor lecture on Decision Trees
- Andrew Moore's Data Mining Tutorials (See tutorials on Decision Trees and Cross Validation)
- Decision Trees (Source: Tan, MSU)
- Tom Mitchell's book slides (See slides on Concept Learning and Decision Trees)
- Weka resources:
    - Introduction to Weka
    - Download Weka
    - Weka Tutorial
    - ARFF format
    - Using Weka from command line

Task 2: Hospital Management System

Data Warehouse consists Dimension Table and Fact Table.

Remember the followingDimension

The dimension object (Dimension):

_ Name

_ Attributes (Levels), with one primary key

_ Hierarchies

One time dimension is must.

About Levels and HierarchiesDimension objects (dimension) consist of a set of levels and a set of hierarchies defined over those levels. The levels represent levels of aggregation. Hierarchies describe parent-child relationships among a set of levels.

For example, a typical calendar dimension could contain five levels. Two hierarchies can be defined on these levels:

H1: YearL>QuarterL>MonthL>WeekL>DayL

H2: YearL>WeekL>DayL

The hierarchies are described from parent to child, so that Year is the parent of Quarter, Quarter the parent of Month, and so forth. About Unique Key Constraints When you create a definition for a hierarchy, Warehouse Builder creates an identifier key for each level of the hierarchy and a unique key constraint on the lowest level (Base Level). Design a Hospital

Management system data warehouse (TARGET) consists of Dimensions Patient, Medicine, Supplier, Time. Where measures are ' NO UNITS', UNIT PRICE.

Assume the Relational database (SOURCE) table schemas as follows

TIME (day, month, year),

PATIENT (patient_name, Age, Address, etc.,)

MEDICINE (Medicine_Brand_name, Drug_name, Supplier, no_units, Uinit_Price, etc.,)

SUPPLIER(Supplier_name, Medicine_Brand_name, Address, etc.,)

If each Dimension has 6 levels, decide the levels and hierarchies, Assume the level names suitably.

Design the Hospital Management system data warehouse using all schemas. Give the example 4-D cube with assumption names.

REFERENCES

1. *Data Mining: Practical Machine Learning Tools and Techniques*, Ian H. Witten, Eibe Frank, Mark A. Hall, 3rd Edition.
2. *Data Warehousing Fundamentals* – PaulrajPonnaiah Wiley student Edition
3. *The Data Warehouse Life cycle Tool kit* – Ralph Kimball Wiley student edition
4. *Building the Data Warehouse*By William H Inmon, John Wiley & Sons Inc, 2005.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – I Semester

L	T	P	C
3	-	-	-

(20HS0842) APTITUDE PRACTICES

COURSE OBJECTIVES

The objectives of this course:

1. *To evaluate various real life situations by resorting to Analysis of key issues and factors.*
2. *To read between the lines and understand various language structures.*
3. *To demonstrate various principles involved in solving Mathematical problems and thereby reducing the time taken for performing job functions.*

COURSE OUTCOMES

On successful completion of this course, the student will be able to

1. *Develop the subtle way of approaching in the candidate.*
2. *Acquired the decision making with in no time.*
3. *Implement logical thinking during professional tenure.*
4. *Improve knowledge on problem solving.*
5. *Understand problems on coding and decoding.*
6. *Apply the knowledge on the concept of reasoning in real life.*

UNIT – I

Percentages, Partnership, Ratio and Proportion**,** Time and Distance, Time and Work.

UNIT – II

Clocks, Calendars, Blood relations, Profit and Loss**,** Simple Interest, Compound Interest.

UNIT – III

Permutations and Combinations, Probability**.**

Menstruation: Areas, Volumes of different solids, Problems on Areas, Volumes and Surface Areas, Cubes.

UNIT – IV

Number and letter series:

Difference series, Product series, Squares series, Cubes series, Alternate series, Combination series, miscellaneous series, Place values of letters.

Number and Letter Analogies:

Definition of Analogy, Problems on number analogy.

Odd man out:

Problems on number Odd man out, Problems on letter Odd man out, Problems on verbal Odd man out.

UNIT – V

Coding and decoding, Directions.

Critical Reasoning:

Problems on assumption, Problems on conclusions, Problems on inferences, Problems on strengthening and weakening of arguments, Problems on principle, Problems on paradox.

TEXTBOOKS

1. Barrons GL, *Thorpe's verbal reasoning*, McGraw Hills, Lsat Materials, 2010.
2. Agarwal R S, *A modern approach to Logical reasoning*, S.Chand, 2017.

REFERENCES

1. Agarwal R S, *Quantitative Aptitude*, S Chand, 2017.
2. Barrons G L, *Quantitative Aptitude*, 2010.
3. Abhijit Guha, *Quantitative Aptitude*, PHI Learning PVT. LTD, 2019.
4. Tyra, *Magical Book on Quicker Maths*, BSC publishing company, 2018.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	1	-	4

(20MC9132) CLOUD COMPUTING

COURSE OBJECTIVES

1. *To introduce the broad perceptive of cloud architecture and model*
2. *To understand the concept of Virtualization and design of cloud Services*
3. *To be familiar with the lead players in cloud.*
4. *To apply different cloud programming model as per need.*
5. *To learn to design the trusted cloud Computing system*

COURSE OUTCOMES

1. *Compare the strengths and limitations of cloud computing*
2. *Identify the architecture, infrastructure and delivery models of cloud computing*
3. *Choose the appropriate cloud player, Programming Models and approach.*
4. *Able to work on various cloud services.*
5. *Address the core issues of cloud computing such as security, privacy and Interoperability*
6. *Design Cloud Services and Set a private cloud*

UNIT-I

The Evolution of Cloud Computing: Hardware Evolution - Internet Software Evolution - Server Virtualization

Web Services Delivered from the Cloud: Communication-as-a-Service - Infrastructure-as-a-Service - Monitoring-as-a-Service - Platform-as-a-Service - Software-as-a-Service.

UNIT-II

Building Cloud Networks: The Evolution from the MSP Model to Cloud Computing and Software-as-a-Service – Collaboration - Basic Approach to a Data Center-Based SOA - The Role of Open Source Software in Data Centers - Where Open Source Software Is Used.

Virtualization Practicum: Downloading and Installing Sun xVM VirtualBox.

UNIT-III

Federation, Presence, Identity, and Privacy in the Cloud: Federation in the Cloud - Presence in the Cloud - Privacy and Its Relation to Cloud-Based Information Systems.

Security in the Cloud: Cloud Security Challenges - Software-as-a-Service Security.

UNIT-IV

Common Standards in Cloud Computing: The Open Cloud Consortium - The Distributed Management Task Force - Standards for Application Developers - Standards for Messaging - Standards for Security.

End-User Access to Cloud Computing: YouTube API Overview – Zimbra – Zoho – DimDim Collaboration.

UNIT-V

Mobile Internet Devices and the Cloud: What Is a Smartphone - Mobile Operating Systems for Smartphones - Mobile Platform Virtualization - Collaboration Applications for Mobile Platforms.

Massively Multiplayer Online Game Hosting on Cloud Resources: Introduction – Background – Model – Experiments.

TEXT BOOKS

1. John W. Rittinghouse and James F. Ransome, "Cloud Computing Implementation, Management, and Security
2. Raj kumar Buyya, James Broberg and Andrzej M. Goscinski, "*Cloud computing: Principles and Paradigms"*, September 2010, John Wiley & Sons.
3. Michael Miller*," Cloud Computing: Web -Based Applications That change the way You Work and Collaborate Online"*, First Edition, 2008, Pearson Education.

REFERENCES

1. Haley beard, "*Cloud Computing best practices for managing and measuring processes for on-demand computing, applications and Data centers in the cloud".*
2. *The Definitive Guide to Cloud Computing,* Dan Sullivan.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	1	-	4

(20MC9133) CYBER SECURITY
(PROGRAMME ELECTIVE – III)

COURSE OBJECTIVES

1. *To understand the cyber-attack*
2. *To understand the types of cybercrimes*
3. *To understand and how to protect them self and ultimately society from such attacks*

COURSE OUTCOMES

After learning the course the students should be able to:

1. *Understand cyber-attack and System Vulnerability Scanning.*
2. *Understand the tools in network defense.*
3. *Knowledge on cyber laws.*
4. *Able to protect themself and ultimately society from cyber-attacks.*
5. *Knowledge on Web application tools.*
6. *Understand Cyber Crimes and types.*

UNIT-I

Systems Vulnerability Scanning: Overview of vulnerability scanning - Open Port / Service Identification - Banner / Version Check - Traffic Probe - Vulnerability Probe - Vulnerability Examples -OpenVAS -Metasploit.

Networks Vulnerability Scanning: Netcat -Socat - understanding Port and Services tools - Datapipe -Fpipe -WinRelay - Network Reconnaissance – Nmap - THC-Amap and System tools. Network Sniffers and Injection tools – Tcpdump and Windump -Wireshark -Ettercap - Hping Kismet

UNIT-II

Network Defense Tools: Firewalls and Packet Filters: Firewall Basics - Packet Filter Vs Firewall - How a Firewall Protects a Network - Packet Characteristic to Filter - Stateless Vs Stateful Firewalls.

Network Address Translation (NAT) and Port Forwarding: The basic of Virtual Private Networks - Linux Firewall - Windows Firewall - Snort: Introduction Detection System

UNIT-III

Web Application Tools: Scanning for web vulnerabilities tools: Nikto - W3af - HTTP utilities - Curl -OpenSSL and Stunnel.

Application Inspection Tools: Zed Attack Proxy -Sqlmap. DVWA -Webgoat - Password Cracking and Brute-Force Tools – John the Ripper - L0htcrack -Pwdump - HTC-Hydra

UNIT-IV

Introduction to Cyber Crime and Law: Cyber Crimes - Types of Cybercrime - Hacking - Attack vectors - Cyberspace and Criminal Behavior - Clarification of Terms - Traditional

Problems Associated with Computer Crime - Introduction to Incident Response - Digital Forensics - Computer Language - Network Language - Realms of the Cyber world.

Internet: A Brief History of the Internet - Recognizing and Defining Computer Crime - Contemporary Crimes - Computers as Targets - Contaminants and Destruction of Data - Indian IT ACT 2000

UNIT-V

Introduction to Cyber Crime Investigation: Firewalls and Packet Filters - password Cracking - Key loggers and Spyware - Virus and Warms - Trojan and backdoors.

Steganography: DOS and DDOS attack - SQL injection - Buffer Overflow - Attack on wireless Networks

TEXT BOOKS

1. *Cyber Security: Analytics, Technology and Automation* by MarttiLehto
2. *Cyber Law & Cyber Crimes Simokified* by Adv. Prashant Mali

REFERENCES

1. *Anti-Hacker Tool Kit (Indian Edition*) by Mike Shema, Publication McGraw Hill.
2. *Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives* by Nina Godbole and SunitBelpure, Publication Wiley

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	1	-	4

(20MC9134) MACHINE LEARNING
(PROGRAMME ELECTIVE – III)

COURSE OBJECTIVES

1. *To develop a broad perspective about the applicability of ML algorithms in different fields.*
2. *To understand the major ML algorithms, the problem settings, and assumptions that underlies them.*
3. *To understand linear models for Regression*

COURSE OUTCOMES

The student will be able to:

1. *Identify the machine learning algorithms which are more appropriate for various types of learning tasks in various domains*
2. *Implement machine learning algorithms on real datasets*
3. *Implement decision tree learning.*
4. *Demonstrate knowledge on Bayesian Learning.*
5. *Able to apply models of hidden markov.*
6. *Demonstrate linear models for Regression*

UNIT-I

Introduction: Well-posed learning problems - Designing a Learning System - Perspectives and Issues in Machine learning.

Concept Learning and General-to-specific Ordering: A concept learning task - Concept learning as Search - Finding a maximally specific hypothesis - Version Spaces and Candidate elimination algorithm - Inductive Bias.

UNIT-II

Decision Tree Learning: Decision tree learning algorithm - Hypothesis space search in decision tree.

Evaluating Hypothesis: Estimating Hypothesis accuracy - Basics of sampling theory - Deriving confidence intervals - Hypothesis testing - comparing learning algorithms.

UNIT-III

Bayesian Learning: Bayes theorem and concept learning - Maximum likelihood and least square error hypotheses - Minimum description length principle - Bayes optimal classifier - Gibbs algorithm - Naive Bayes classifier.

Computational Learning Theory: Probably learning an approximately correct hypothesis - PAC learnability - The VC dimension - the mistake bound model for learning.

UNIT-IV

Linear Models for Regression: Linear basis function models - The Bias-Variance decomposition - Bayesian Linear Regression - Bayesian Model comparison.

Kernel Methods: Constructing kernels - Radial basis function networks - Gaussian Processes.

UNIT-V

Approximate Inferencing: Variational inference -Variational mixture of Gaussians - Variational linear regression -Variational logistic regression.

Hidden Markov Models: Learning algorithms for HMM - The Viterbi algorithm - Linear Dynamical Systems. Reinforcement Learning: The learning task - Q learning - Non-deterministic rewards and action - Temporal difference learning - Generalizing from examples

TEXT BOOKS

1. Mitchell, T. M., (1997), Machine Learning, McGraw-Hill.
2. EthernAlpaydin, Introduction to Machine Learning. Eastern Economy Edition, Prentice Hall of India, 2005.

REFERENCES

1. Bishop, C. M., (2007), Pattern Recognition and Machine Learning, Springer.
2. T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning, 2e, 2008

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	1	-	4

(20MC9135) SOFTWARE PROJECT MANAGEMENT
(PROGRAMME ELECTIVE – III)

COURSE OBJECTIVES

1. *To know of how to do project planning for the software process.*
2. *To learn the cost estimation techniques during the analysis of the project.*
3. *To understand the quality concepts for ensuring the functionality of the software*

COURSE OUTCOMES

1. *Understand the activities during the project scheduling of any software application.*
2. *Learn the risk management activities and the resource allocation for the projects.*
3. *Can apply the software estimation and recent quality standards for evaluation of the software projects.*
4. *Understand the flow and interactive process.*
5. *Acquire knowledge and skills needed for the construction of highly reliable software project.*
6. *Able to create reliable, replicable cost estimation that links to the requirements of project planning and managing*

UNIT-I

Conventional Software Management: The Waterfall Model - Conventional software Management Performance.

Evolution of Software Economics: Software Economics - Pragmatic Software Cost Estimation. Improving Software Economics: Reducing Software Product Size - Improving software Processes - Improving Team Effectiveness - Improving Automation - Achieving Required Quality - Peer Inspections.

UNIT-II

Conventional and Modern Software Management: Principles of Conventional Software Engineering - Principles of Modern Software Management - Transitioning to an Iterative Process. Life Cycle Phases: Engineering and Production Stages - Inception. Elaboration - Construction - Transition Phases.

Artifacts of the Process: The Artifact Sets. Management Artifacts - Engineering Artifacts - Programmatic Artifacts. Model Based Software Architectures: A Management Perspective and Technical Perspective.

UNIT-III

Flows of the Process: Software Process Workflows. Inter Trans Workflows. Checkpoints of the Process - Major Mile Stones - Minor Milestones - Periodic Status Assessments.

Interactive Process Planning: Work Breakdown Structures - Planning Guidelines - Cost and Schedule Estimating. Interaction Planning Process. Pragmatic Planning.

UNIT-IV

Project Organizations and Responsibilities: Line-of-Business Organizations - Project Organizations - and Evolution of Organizations.

Process Automation: Automation Building Blocks - the Project Environment. Project Control and Process Instrumentation: Server Care Metrics - Management Indicators - Quality Indicators - Life Cycle Expectations Pragmatic Software Metrics - Metrics Automation. Tailoring the process: Process Discriminates - Example.

UNIT-V

Future Software Project Management: Modern Project Profiles Next Generation Software economics - Modern Process Transitions.

Case Study: The Command Center Processing and Display System –Replacement (CCPDS-R)

TEXT BOOKS

1. Walker Rayce - "*Software Project Management*" - 1998 - PEA.
2. Henrey - "*Software Project Management*"' Pearson.

REFERENCES

1. *Software Engineering Project Management* -Richard H. Thayer -1997 - IEEE Computer Society.
2. *Software Engineering and Management* -Shere K. D - 1998 - PHI.
3. *Software Project Management: A Concise Study* - S. A. Kelkar - PHI.
4. *Software Project Management* - Hughes Cotterell - 2e - TMH.
5. *Software Project Management from Concept to Development*" -Kaeron Conway - Dream Tech.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	1	-	4

(20MC9136) .NET TECHNOLOGIES
(PROGRAMME ELECTIVE – III)

COURSE OBJECTIVES

1. *Introduce to .Net IDE Component Framework*
2. *Programming concepts in .Net*
3. *Creating Websites using ASP.Net Controls*

COURSE OUTCOMES

1. *Aware of .net framework components.*
2. *Creating simple data binding applications in VB using ADO.Net connectivity.*
3. *Able to create a web form application using c#.*
4. *Performing Database operations for windows form.*
5. *Able to create a web applications.*
6. *Creating user interactive web pages.*

UNIT-I

Introduction To .Net Framework: .NET Overview - Behind Microsoft .NET- The .NET Platform - .NET Framework Design Goals -.NET Framework.

The Common Language Runtime: CLR Environment - CLR Executable – Metadata - JIT Compilation - Automatic Memory Management -Assemblies and Manifests - Intermediate Language (IL) - The CTS and CLS - CLR Execution..Net Programming **–** Common Programming Model – Core Features and Languages

UNIT-II

Introduction To Visual basic .Net : Visual Basic .NET variables – Data types constants – building projects – Displaying Output – Operators – Conditional Statements – If – Then, Select Case – Looping – Do, For Next, Nested Loops.

Interfacing with the End User: Windows Forms – Threading – MDI Applications – Components and Controls – Menus and Toolbars –Collecting User Input

UNIT-III

C -Sharp Language (C#): Introduction – Data Types – Identifiers – Variables – Constants – Literals – Array and Strings.

C# OOPs: Object and Classes – Inheritance and Polymorphism – Operator Overloading – Interfaces – Delegates and Events – Type conversion.

UNIT-IV

Application Development Using Ado .Net: Features of ADO.NET. Architecture of ADO.NET – ADO.NET providers –Accessing Data bases Using ADO.NET- Connection opening and closing– Command object – Data Adapter – Dataset – DataTables - Controlling table views with DataViews and DataRelation Objects – Data-binding.

Database Connection - Windows Forms using C# and VB.NET.

UNIT-V

ASP .Net: Anatomy of ASP .NET Page – ASP.Net Features – Introduction to Web Forms Server Controls – label, dropdown list box, Button, AdRotator , Textbox, Checkbox etc., – Validation controls. ASP.NET Web Services – State Management – Caching – Authentication (window,.Net Passport, Forms Based) – Securing ASP.NET Applications.

Crystal Reports: Creating - Setup - Deployment.

TEXT BOOKS

1. *.NET Framework Essentials*, Thuan L. Thai, Hoang Lam Publisher: O'Reilly. Third Edition, 2003
2. *Programming Microsoft® Visual C#® 2008: The Language*, Donis Marshall Microsoft Press,2008.

REFERENCES

1. *Programming Microsoft® Visual Basic® .NET (Core Reference)*, Francesco Balena, Microsoft Press, 2006.
2. *Microsoft® ADO.NET Step by Step*,Rebecca M. Riordan, Microsoft Press, 2002
3. *Professional ASP.NET 2.0 XML*, ThiruThangarathinam ,Wiley Publishing, Inc. 2006
4. *Building Microsoft® ASP.NET Applications for Mobile Devices*, Andy Wigley, Peter Roxburgh , Second Edition, Microsoft Press,2003

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	1	-	4

(20MC9137) BIO – INFORMATICS
(PROGRAMME ELECTIVE – III)

COURSE OBJECTIVES

The basic objective is to give students

1. *Introduction to the basic practical techniques of bioinformatics.*
2. *Emphasis will be given to the application of bioinformatics and biological databases to problem solving in real research problems.*
3. *The students will become familiar with the use of a wide variety of internet applications, biological database and will be able to apply these methods to research problems*

COURSE OUTCOMES

The students will be able to:

1. *Describe the contents and properties of the most important bioinformatics databases, perform text- and sequence-based searches.*
2. *Analyze and discuss the results in light of molecular biological knowledge*
3. *Explain the major steps in pairwise and multiple sequence alignment.*
4. *Explain the principle for, and execute pairwise sequence alignment by dynamic programming*
5. *Able to apply methods of photo genetics.*
6. *Predict the secondary and tertiary structures of protein sequences.*

UNIT-I

Introduction to Bioinformatics: Definition and History and Applications of Bioinformatics.

Resources: Internet resources - various databases and bioinformatics tools - organization of databases.

UNIT-II

Biological Databases: Nucleic acid sequence databases - Protein sequence databases - Repositories for high throughput genomic sequences - Genome Databases.

3D Structure Database - Chemical Structure database - Gene Expression database - Derived Databases - Structure classification database - Protein-Protein interaction database and Pathway database.

UNIT-III

Sequence Analysis: File formats - Basic concepts of sequence analysis - Scoring matrices - Pair wise sequence alignments - Multiple sequence alignment.

Database Searches: Keyword-based searches and Sequence-based searches - Profile-based searches - Markov chains and applications.

UNIT-IV

Global and Local Alignments: Algorithms - Similarities - Semi global alignment - Multiple Sequence Alignment Goals – Definition.

Complex Methods: Database of multiple Alignment - Searching database with multiple alignment.

UNIT-V

Methods of Photo Genetics: Distance Based Methods & their Comparison. RNA Structure: Amino Acids - Polypeptide Composition Algorithm - Modeling protein folding prediction - RNA Sequence Structure.

Proteomics: Classification - Techniques - Inheritors - Drying Design - Structures - X-Ray Crystal – NMR – Empirical methods and prediction techniques.

TEXT BOOKS

1. *Bioinformatics: Sequence, Structure and Databanks A Practical Approach (The Practical Approach Series, 236)*, Des Higgins (Editor), Willie Taylor. 1st edition,
2. *Bioinformatics: Sequence and Genome Analysis*, David W. Mount. 2nd edition
3. *Introduction to Bioinformatics*, T.K.Attwood, D.J. Parry-Smith, Pearson Education.

REFERENCES

1. *Introduction to Bioinformatics*, Teresa Attwood, David Parry-Smith, 1st edition, May 2001, Pearson Education.
2. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*, Andreas D. Baxevanis, B. F. Francis Ouellette. 3nd edition.
3. *Fundamental Concepts of Bioinformatics*, Dan E. Krane, Michale L. Raymer, Pearson Education Asia.
4. *Developing Bioinformatics Computer Skills*, Cynthia Gibas, Per Jambeck, O' Reilly.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	1	-	4

(20MC9138) NEURAL NETWORKS & FUZZY LOGIC
(PROGRAMME ELECTIVE – IV)

COURSE OBJECTIVES

1. *To cater the knowledge of Neural Networks and Fuzzy Logic.*
2. *To provide multi-layer perceptron.*
3. *To Control and use these for controlling real time systems.*

COURSE OUTCOMES

1. *To Expose the students to the concepts of feed forward neural networks*
2. *To provide adequate knowledge about feedback networks.*
3. *To*
4. *To teach about the concept of fuzziness involved in various systems. To provide adequate knowledge about fuzzy set theory.*
5. *To provide comprehensive knowledge of fuzzy logic control and adaptive fuzzy logic and to design the fuzzy control using genetic algorithm.*
6. *To provide adequate knowledge of application of fuzzy logic control to real time systems.*

UNIT-I

Introduction: Evolution of neural networks; Artificial Neural Network: Basic model - Classification - Feed forward and Recurrent topologies - Activation functions;

Learning Algorithms: Supervised - Un-supervised and Reinforcement; Fundamentals of connectionist modeling: McCulloach – Pits model - Perceptron -Adaline -Madaline.

UNIT–II

Topology of Multi-Layer Perceptron: Back propagation learning algorithm - limitations of Multi-layer perceptron.

Radial Basis Function Networks: Topology - learning algorithm; Kohenen's self-organising network: Topology - learning algorithm; Bidirectional associative memory Topology - learning algorithm - Applications.

UNIT–III

Recurrent Neural Networks: Basic concepts - Dynamics - Architecture and training algorithms - Applications;

Hopfield Network: Topology - learning algorithm - Applications; Industrial and commercial applications of Neural networks - Semiconductor manufacturing processes - Communication - Process monitoring and optimal control - Robotics - Decision fusion and pattern recognition.

UNIT–IV

Classical and Fuzzy Sets: Introduction - Operations and Properties - Fuzzy Relations - Cardinality - Operations and Properties - Equivalence and tolerance relation - Value assignment - cosine amplitude and max-min method;

Fuzzification and Defuzzification: Fuzzification - Membership value assignment- Inference - rank ordering - angular fuzzy sets. Defuzzification methods - Fuzzy measures - Fuzzy integrals - Fuzziness and fuzzy resolution; possibility theory and Fuzzy arithmetic - composition and inference - Considerations of fuzzy decision-making.

UNIT-V

Basic Structure of Fuzzy Logic Control Systems: Design methodology and stability analysis of fuzzy control systems;

Operation of Fuzzy Logic Control Systems: Applications of Fuzzy controllers. Applications of fuzzy theory.

TEXT BOOKS

1. *Neural Networks in Computer Intelligence*,Limin Fu, McGraw Hill, 2003.
2. *Soft Computing and Intelligent Systems Design, Theory, Tools and Applications*, Fakhreddine O. Karray and Clarence De Silva, Pearson Education, India, 2009.

REFERENCES

1. *Fuzzy Logic with Engineering Applications*, Timothy J. Ross, McGraw Hill,1995.
2. *Artificial Neural Networks*, B.Yegnanarayana, PHI, India, 2006.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	1	-	4

(20MC9139) IMAGE PROCESSING
(PROGRAMME ELECTIVE – IV)

COURSE OBJECTIVES

1. *The fundamentals of digital image processing*
2. *Understand image transform used in digital image processing*
3. *Understand image compression and Segmentation used in digital image processing*

COURSE OUTCOMES

1. *Able to enhance images using enhancement techniques.*
2. *Able to restore images using restoration techniques and methods used in digital image processing*
3. *Able to transform the image in digital image processing.*
4. *Able to image enhancement techniques used in digital image processing.*
5. *Able to compress images using compression techniques used in digital image processing*
6. *Able to segmentation of images using digital image processing.*

UNIT-I

Introduction: Fundamental steps of image processing - components of an image processing of system - the image model and image acquisition - sampling and quantization - station ship between pixels - distance functions - scanner.

UNIT-II

Statistical and Spatial Operations: Grey level transformations - histogram equalization - smoothing & sharpening - spatial filters - frequency domain filters - holomorphic filtering - image filtering & restoration.

Inverse and Wiener Filtering: FIR wiener filter. Filtering using image transforms - smoothing splines and interpolation.

UNIT-III

Morphological and Other Area Operations: basic morphological operations - opening and closing operations - dilation erosion - Hit or Miss transform - morphological algorithms - extension to grey scale images.

Segmentation and Edge Detection Region Operations: Basic edge detection - second order detection - crack edge detection - gradient operators - compass and laplace operators - edge linking and boundary detection -thresholding - region based segmentation - segmentation by morphological watersheds.

UNIT-IV

Image Compression: Types and requirements - statistical compression - spatial compression - contour coding - quantizing compression - image data compression - predictive technique - pixel coding - transfer coding theory - loss and lossless predictive type coding.

Basics of Color Image Processing: pseudo color image processing - color transformation - color smoothing and sharpening - color segmentation - color image compression - compression standards.

UNIT-V

Image Transforms: Fourier - DFT - DCT - DST -Haar - Hoteling -Karhunen - Loeve - Walsh -Hadamard - Slant.

Representation and Description: Chain codes - Polygonal approximation - Signatures Boundary Segments -Skeltons - Boundary Descriptors - Regional Descriptors - Relational Descriptors - PCA.

TEXT BOOKS

1. *Digital Image Processing* – by Rafael.C.Gonzalez& Richard E.Woods - Pearson Education.
2. *Digital Image Processing* -M.Anji Reddy -Y.Hari Shankar - BS Publications.
3. *Fundamentals of Digital Image Processing* – by A.K. Jain - PHI.

REFERENCES

1. *Digital Image Processing*–William K - Part I -John Wiley edition.
2. *Digital Image Processing using MATLAB* –by Rafael.C.Gonzalez - Richard E.Woods -& Steven L.Eddins - Pearson Education.
3. *Digital Image Processing* - Kenneth R. Castleman - Pearson Education.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	1	-	4

(20MC9140) DESIGN PATTERNS
(PROGRAMME ELECTIVE – IV)

COURSE OBJECTIVES

1. *Indicate which underlying object oriented design principle(s) it is based on.*
2. *Explain what specific object oriented design problem the pattern solves.*
3. *Provide a specific context for each pattern in which it can be applied.*

COURSE OUTCOMES

1. *Students demonstrate a thorough understanding of patterns and their underlying principles*
2. *Students know what design pattern to apply to a specific problem*
3. *Students demonstrate what tradeoffs need to be made when implementing a design pattern*
4. *Able to draw UML diagrams for different patterns.*
5. *Students can able draw class diagrams for different patterns.*
6. *Students will be able to use design patterns when developing software*

UNIT-I

Introduction about Design Pattern: What Is a Design Pattern? - Design Patterns in Smalltalk MVC - Describing Design Patterns - The Catalog of Design Patterns - Organizing the Catalog - How Design Patterns Solve Design Problems - How to Select a Design Pattern - How to Use a Design Pattern.

UNIT-II

A Case Study: Designing a Document Editor: Design Problems - Document Structure - Formatting - Embellishing the User Interface - Supporting Multiple Look-and-Feel Standards - Supporting Multiple Window Systems - User Operations Spelling Checking and Hyphenation - Summary.

UNIT-III

Creational Patterns: Abstract Factory - Builder - Factory Method - Prototype - Singleton - Discussion of Creational Patterns.

UNIT-IV

Structural Pattern: Adapter - Bridge - Composite - Decorator - Facade - Flyweight - Proxy.

UNIT-V

Behavioral Patterns: Chain of Responsibility - Command - Interpreter - Iterator - Mediator - Memento - Observer - State - Strategy - Template Method - Visitor.

TEXT BOOKS

1. *Design Patterns,* Erich Gamma, Pearson Education
2. *Pattern's in JAVA Vol-I,* Mark Grand, Wiley DreamTech.

REFERENCES

1. *Pattern's in JAVA Vol-II,* Mark Grand, Wiley DreamTech.
2. *JAVA Enterprise Design Patterns Vol-III,* Mark Grand, Wiley DreamTech.
3. *Head First Design Patterns,* Eric Freeman-Oreilly-spd.
4. *Design Patterns Explained,* Alan Shalloway, Pearson Education.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	1	-	4

(20MC9141) BIG DATA ANALYTICS
(PROGRAMME ELECTIVE – IV)

COURSE OBJECTIVES

1. *To explore the fundamental concepts of big data analytics*
2. *To learn to analyze the big data using intelligent techniques.*
3. *To understand the applications using Map Reduce Concepts.*

COURSE OUTCOMES

The students will be able to:

1. *Work with big data platform and analyze the big data analytic techniques for useful business applications.*
2. *Design efficient algorithms for mining the data from large volumes.*
3. *Analyze the HADOOPtechnologies associated with big data analytics*
4. *Analyze the Map Reduce technologies associated with big data analytics*
5. *Explore on Big Data applications Using Pig and Hive*
6. *Understand the fundamentals of various bigdata analysis techniques*

UNIT-I

Introduction to BigData Platform: Challenges of Conventional Systems - Intelligent data analysis – Nature of Data - Analytic Processes and Tools - Analysis vs Reporting - Modern Data Analytic Tools.

Statistical Concepts: Sampling Distributions - Re-Sampling - Statistical Inference - Prediction Error.

UNIT-II

Introduction To Streams Concepts: Stream Data Model and Architecture - Stream Computing - Sampling Data in a Stream – Filtering Streams – Counting Distinct Elements in a Stream – Estimating Moments – Counting Oneness in a Window – Decaying Window.

Real time Analytics Platform: RTAP Applications - Case Studies - Real Time Sentiment Analysis, Stock Market Predictions.

UNIT-III

History of Hadoop: - The Hadoop Distributed File System – Components of Hadoop - Analyzing the Data with Hadoop - Scaling Out - Hadoop Streaming - Design of HDFS - Java interfaces to HDFS Basics.

Developing a Map Reduce Application: How Map Reduce Works - Anatomy of a Map Reduce Job run – Failures - Job Scheduling - Shuffle and Sort – Task execution - Map Reduce Types and Formats - Map Reduce Features.

UNIT-IV

Setting up a Hadoop Cluster: Cluster specification - Cluster Setup and Installation – Hadoop Configuration - Security in Hadoop.

Administering Hadoop: Administering – HDFS – Monitoring – Maintenance – Hadoop benchmarks - Hadoop in the cloud.

UNIT-V

Applications on Big Data Using Pig and Hive: Data processing operators in Pig – Hive services – HiveQL – Querying Data in Hive - fundamentals of HBase and ZooKeeper - IBM InfoSphereBigInsights and Streams.

Visualizations: Visual data analysis techniques, interaction techniques; Systems and applications

TEXT BOOKS

1. *Intelligent Data Analysis*, Michael Berthold, David J. Hand, Springer, 2007.
2. *Hadoop: The Definitive Guide,* Tom White, Third Edition, O'reilly Media, 2012.

REFERENCES

1. *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*,Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, McGrawHill Publishing, 2012
2. *Mining of Massive Datasets*, AnandRajaraman and Jeffrey David Ullman, Cambridge University Press, 2012.
3. *Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics*, Bill Franks, JohnWiley& sons, 2012.
4. *Making Sense of Data,* Glenn J. Myatt, John Wiley & Sons, 2007

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY :: PUTTUR
(AUTONOMOUS)

MCA II Year – II Semester

L	T	P	C
3	1	-	4

(20MC9142) ENTERPRISE RESOURCE PLANNING
(PROGRAMME ELECTIVE – IV)

COURSE OBJECTIVES

1. *With the basic concepts of ERP systems for manufacturing or service companies, and the differences among (Material Requirement Planning) MRP, MRP II, and ERP systems;*
2. *Apply the principles of ERP systems, their major components, and the relationships among these components;*
3. *To be able to map business processes using ERP concepts and techniques.*

COURSE OUTCOMES

After completing this course, student will be able to

1. *Classify different processes of the organization and relationship among all processes.*
2. *Examine systematically the planning mechanisms in an enterprise, and identify all components in an ERP system and the relationships among the components;*
3. *To describe the Generic Model of ERP and General ERP Implementation Methodology.*
4. *To apply the concept of ERP.*
5. *To apply the concepts of CRM and ERP*
6. *To demonstrate knowledge of SAP and Oracle Apps.*

UNIT-I

Introduction to Enterprise Resource Planning: Introduction of the term Business Process Reengineering(BPR) - BPR Methodology - Current BPR Tools.

Introduction to Material Requirement Planning (MRP): Definition of Enterprise Resource Planning (ERP) - Evolution of ERP - Characteristics - Features - Components and needs of ERP - ERP Vendors - Benefits & Limitations of ERP Packages.

UNIT-II

Enterprise Modeling of ERP: Need to focus on Enterprise Integration/ERP - Information mapping - Role of common shared Enterprise database - System Integration - Logical vs. Physical System.

Integration of ERP: Benefits & limitations of System Integration - ERP's Role in Logical and Physical Integration

UNIT-III

ERP Architecture and Implementation Methodology of ERP: Generic Model of ERP system - Core Modules functionality.

Types of ERP Architecture: Client Server Architecture – Web-based Architecture - Service Oriented Architecture (SOA) - Difficulty in selecting ERP - Approach to ERP selection - Request for Proposal approach - Proof-of-Concept approach - General Implementation

Methodology of ERP - Vanilla Implementation - Evaluation Criteria of ERP packages - Project Implementation Team Structure

UNIT-IV

Introduction to SAP: SAP - Integrated SAP Model - SAP Architecture - SAP R/3 System &mySAP - SAP Modules.

Oracle Apps: Oracle AIM Methodology - Oracle Fusion Modules - A Comparative assessment of ERP Packages.

UNIT-V

Supply Chain Management and ERP: Definition of Supply Chain Management (SCM) - Supply Chain Council's SCOR Model - Stevens Model of Supply Chain Management - Aims of SCM - SCM Key Drivers - Collaborative Design & Product Development - Benefits of SCM - ERP Vs SCM - Key SCM Vendors

Customer Relationship Management and ERP: Definition of Customer Relationship Management (CRM) - CRM Evolution - CRM Delivery Processes - CRM support Processes - CRM Analysis Processes - CRM Components - Key CRM Vendors.

TEXT BOOKS

1. *Enterprise Systems for Management*, Luvai F. Motiwalla, Jeff Thompson, Pearson Education., 2nd Ed., 2011.
2. *Enterprise Resource Planning*, Ravi Shankar, S.Jaiswal, Galgotia Publication Pvt. Ltd., 1st Ed., 1999.

REFERENCES

1. *CRM at the speed of Light: Social CRM strategies, tools and techniques for engaging your customers:* 4th edition by Paul Greenberg , McGraw Hill ,2009
2. *Supply Chain Management Casebook : The Comprehensive Coverage and Best Practices in SCM*, Chuck Munson , Pearson FT Press 2013, ISBN-13: 978-0- 13-336723-2
3. *Definitive Guide to Supply Chain Best Practices, The Comprehensive Lessons and Cases in Effective SCM*, Robert Frankel , Pearson FT Press , 2014
4. *Enterprise Resource Planning*, Mary Sumner , Prentice Hall , 2005
5. *Supply Management*, David Burt, McGraw Hill Publications, 8th Ed., 2010