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	19HS0810	Communicative English	3	-	-	3
2	19HS0835	Probability & Statistics	3	-	-	3
3	19HS0836	Discrete Mathematics	3	-	-	3
4	19MC9101	Computer Programming and Problem Solving	3	1	-	4
5	19MC9102	Computer Organization	3	-	-	3
6	19HS0811	Communicative English Lab	-	-	3	1.5
7	19MC9103	C Programming Lab	-	-	3	1.5
8	19MC9104	P. C. Software Lab	-	-	4	2
Contact periods/Week			15	1	10	21
			Total/Week 26			

MCA I Year - II Semester

S. No.	Course Code	Subject	L	T	P	C
1	19MB9056	Accounting & Financial Management	3	-	-	3
2	19MC9105	Object Oriented Programming through C++	3	-	-	3
3	19MC9106	Data Structures	3	-	-	3
4	19MC9107	Operating Systems	3	-	-	3
5	19MC9108	System Software	3	-	-	3
6	19MC9109	Programming in C++ Lab	-	-	4	2
7	19MC9110	Data Structures through C++ Lab	-	-	4	2
8	19MC9111	Operating Systems Lab	-	-	4	2
Audit Course						
9	19HS0843	Aptitude Practice – I	3	-	-	-
Contact periods/Week			18	-	12	21
			Total/Week 30			

MCA II Year -I- Semester

S. No.	Course Code	Subject	L	T	P	C
1	19MC9112	Computer Networks	3	-	-	3
2	19MC9113	JAVA Programming	3	-	-	3
3	19MC9114	Database Management System	3	-	-	3
4	19MC9115	Advanced Programming (Python & R Languages)	3	-	-	3
5	19MC9116	Design and Analysis of Algorithms	3	-	-	3
6	19MC9117	JAVA Programming Lab	-	-	4	2
7	19MC9118	Database Management System Lab	-	-	4	2
8	19MC9119	Advanced Programming Lab	-	-	4	2
Credit Course						
9	COE-I	Comprehensive Online Examination – I	-	-	-	1
Audit Course						
10	19HS0820	Comprehensive Soft Skills	3	-	-	-
		Contact periods/Week	18	-	12	22
			Total/Week 30			

MCA II Year-II- Semester

S. No.	Course Code	Subject	L	T	P	C
1	19MC9120	Data Warehousing and Data Mining	3	-	-	3
2	19MC9121	Web Technologies	3	-	-	3
3	19MC9122	Software Engineering	3	-	-	3
4	 19MC9123 19MC9124 19MC9125 19MC9126 19MC9127	**Elective – I** Software Testing Artificial Intelligence Distributed Systems Linux Programming Network Security	3	-	-	3
5	 19MC9128 19MC9129 19MC9130 19MC9131 19MC9132	**Elective – II** Human Computer Interaction Social Networks and Semantic Web Computer Graphics Internet of Things E-Commerce	3	-	-	3
6	19MC9133	Data Warehousing and Data Mining Lab	-	-	4	2
7	19MC9134	Web Technologies Lab	-	-	4	2
8	19MC9135	Software Engineering & UML Lab	-	-	4	2
Credit Course						
9	COE-II	Comprehensive Online Examination – II	-	-	-	1
Audit Course						
10	19HS0844	Aptitude Practice - II	3	-	-	-
		Contact periods/Week	18	-	12	22
			Total/Week 30			

MCA III Year-I- Semester

S. No.	Course Code	Subject	L	T	P	C
1	19MC9136	Cloud Computing	3	-	-	3
2	19MC9137	.Net Technologies	3	-	-	3
3	19MC9138	Mobile Application Development	3	-	-	3
4	19MC9139 19MC9140 19MC9141 19MC9142 19MC9143	**Elective – III** Cyber Security Software Project Management Neural Networks & Fuzzy logic Big Data Analytics Model Driven Framework	3	-	-	3
5	19MC9144 19MC9145 19MC9146 19MC9147 19MC9148	**Elective – IV** Bio-informatics Image Processing Design Patterns Machine Learning Enterprise Resource Planning	3	-	-	3
6	19MC9149	Cloud Computing Lab	-	-	4	2
7	19MC9150	.Net Technologies Lab	-	-	4	2
8	19MC9151	Mobile Application Development Lab	-	-	4	2
Credit Course						
9	COE-III	Comprehensive Online Examination – III	-	-	-	1
Audit Course						
10	19HS0821	Advanced English Language and Communication Skills Lab	-	-	4	-
		Contact periods/Week	15	-	16	22
			Total/Week 31			

MCA III Year-II- Semester

S. No.	Course Code	Subject	L	T	P	C
1	19MC9152	Seminar	-	-	4	2
2	19MC9153	Dissertation	-	-	20	10
		Contact periods/Week	-	-	24	12
			Total/Week 24			

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

Total Credits: 120

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

MCA I Year – I Semester

L	T	P	C
3	-	-	3

(19HS0810) COMMUNICATIVE ENGLISH

COURSE OBJECTIVES

1. *To facilitate effective listening skills for better comprehension of academic lectures and English spoken by native speakers.*
2. *To focus on appropriate reading strategies for comprehension of various academic texts and authentic materials.*
3. *To help improve speaking skills through participation in activities such as role plays, discussions and structured talks/oral presentations.*
4. *To impart effective strategies for good writing and demonstrate the same in summarizing, writing well organized essays, record and report useful information.*
5. *To provide knowledge of grammatical structures and vocabulary and encourage their appropriate use in speech and writing.*

COURSE OUTCOMES

1. *To understand social or transactional dialogues spoken by native speakers of English and identify the context, topic, and pieces of specific information.*
2. *To employ suitable strategies for skimming and scanning to get the general idea of a text and locate specific information.*
3. *To Participate in informal discussions and speak clearly on a specific topic or in general.*
4. *To Comprehend, discuss and respond to academic texts and use appropriate language for description and interpretation in writing*
5. *To form sentences using proper grammatical structures and correct word forms.*

UNIT – I

Part-1

Listening: Identifying the topic, the context and specific pieces of information by listening to short audio texts and answering a series of questions. **Speaking:** Asking and answering general questions on familiar topics such as home, family, work, studies and interests; introducing oneself and others. **Reading:** Skimming to get the main idea of a text; scanning to look for specific pieces of information. **Writing:** Beginnings and endings of paragraphs - introducing the topic, Letter writing. **Grammar and Vocabulary:** Parts of speech; singular and plural; Basic sentence structures; simple question form - wh-questions; word order in sentences and Content words

Part-2

Half a Rupee Worth by R K Narayan from Engage with English.

UNIT – II

Part-1

Listening: Answering a series of questions about main idea and supporting ideas after listening to audio texts. **Speaking:** Discussion in pairs/ small groups on specific topics followed by short structured talks. **Reading:** Identifying sequence of ideas; recognizing verbal techniques that help to link the ideas in a paragraph together. **Writing:** **M**echanics of writing - punctuations **Grammar and Vocabulary:** Cohesive devices - linkers, sign posts and transition signals; use of articles and zero article; prepositions and function words

Part-2

The Thakur's Well by Premchand from Paths to Skills in English.

UNIT – III

Part-1

Listening: Listening for global comprehension and summarizing what is listened to. **Speaking:** Discussing specific topics in pairs or small groups and reporting what is discussed **Reading:** Reading a text in detail by making basic inferences -recognizing and interpreting specific context clues; strategies to use text clues for comprehension. **Writing:** Summarizing Report Writing. **Grammar and Vocabulary:** Verbs - tenses; subject-verb agreement; direct and indirect speech, reporting verbs for academic purposes and word forms

Part-2

I am not that Woman by Kishwar Naheed from Engage with English.

UNIT – IV

Part-1

Listening: Making predictions while listening to conversations/ transactional dialogues without video; listening with video. **Speaking:** conversational English in academic contexts (formal and informal). **Reading:** Studying the use of graphic elements in texts to convey information, reveal trends/patterns/relationships, communicate processes or display complicated data. **Writing:** Information transfer **Grammar and Vocabulary:** Quantifying expressions - adjectives and adverbs; comparing and contrasting; degrees of comparison; use of synonyms and antonyms.

Part-2

What is my name? By Sathyavathi from Paths to Skills in English.

UNIT – V

Part-1

Listening: Identifying key terms. **Speaking:** Formal oral presentations on topics from academic contexts - without the use of PPT slides. **Reading:** Reading for comprehension. **Writing:** Writing structured essays on specific topics using suitable claims and evidences. **Grammar and Vocabulary:** Editing short texts –identifying and correcting common errors in grammar and usage (articles, prepositions, tenses, subject verb agreement).

Part-2

The Power of Prayer by A P J Abdul Kalam from Paths to Skills in English.

TEXTBOOKS

1. *Board of Editors Engage with English*, Orient Blackswan First Edition, 2016
2. *Paths to Skills in English*, Prof. G.M. Sundaravalli & A.S.Kamalakar, Orient Blackswan, First Edition, 2015.

REFERENCES

1. *Academic writing: A handbook for international students*, Bailey, Stephen. Routledge, 2014.
2. *Pathways: Listening, Speaking and Critical Thinking*, Chase, Becky Tarver. Heinley ELT; 2nd Edition, 2018.
3. *Cambridge Academic English (B2)*, Hewings, Martin, CUP, 2012.
4. *Study Reading: A Course in Reading Skills for Academic Purposes*, Eric H.Glendinning & Beverly Holmström, Cambridge University Press; 2 edition, 14 October 2004.

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

MCA I Year – I Semester

L	T	P	C
3	-	-	3

(19HS0835) PROBABILITY & STATISTICS

COURSE OBJECTIVES

1. *To train the students thoroughly in Mathematical concepts fundamentals of probability, test of hypothesis, Test of significance.*
2. *To prepare students for lifelong learning and successful careers using mathematicalConcepts of probability test of hypothesis, Test of significance.*
3. *To develop the skill pertinent to the practice of the mathematical concepts including the Student abilities to formulate and modeling the problems, to think creatively and to Synthesize information*

COURSE OUTCOMES

1. *Have acquired ability to participate effectively in group discussions*
2. *Have developed ability in writing in various contexts*
3. *Have acquired a proper level of competence for employability*
4. *To understand the basic notions of discrete and continuous probability.*
5. *To understand the methods of statistical inference and the role that sampling distributions play in those methods.*

UNIT – I

Probability: Sample spaces and Events - Definition of probability - Addition theorem - Conditional probability - Independence - Baye‟s theorem.

Random variables: Discrete and Continuous random variables - Probability distribution functions – properties - Expectation of Discrete - Continuous random variables.

UNIT-II

Probability Distributions: Binomial distribution – Poisson distribution - Normal distributions - Mean and Variance of Binomial – Poisson - Normal distributions - Related problems.

UNIT- III

Basic Statistics: Measures of Central tendency (Mean, Median & Mode) – Moments – Skewness – Kurtosis - Correlation – Regression – Rank correlation.

UNIT IV

Curve fitting: Curve fitting by the method of least squares - Fitting of straight lines - Second degree parabolas - More general curves (Exponential & Power curve).

Test of Hypothesis: Large sample test for single proportion – Difference of proportions - Single mean - Difference of means.

UNIT V

Tests of significance of Small Samples: t-test for single mean - Difference of means - F-test - Chi-square test for goodness of fit and Independence of attributes

TEXT BOOKS

1. *Higher Engineering Mathematics*, B.S. Grewal, Khanna Publishers, 42nd Edition,2015
2. *Probability & Statistics*, T.K.V. Iyengar, S.Chand publications, Fourth revised Edition, 2012.

REFERENCES

1. *Statistical methods*, S.P. Gupta, Sultan publications, Thirty four Reprint Edition, 2006.
2. *Probability and Statistics*, Dr. Shahnaz Bathul, Unitech Series, 2005-2006.
3. *Probability and Statistics*, E. Rukmangadachari, Pearson Publication, 2015.
4. *Probability and Statistics*, G.S.S. Bhishma Rao, SCI Tech, 3rd Edition, 2007
5. *Probability, Statistics and Random Process*, K. Murugesan, Anuradha Publication, 2000

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

MCA I Year – I Semester

L	T	P	C
3	-	-	3

(19HS0836) DISCRETE MATHEMATICS

COURSE OBJECTIVES

1. *Familiarize closed from solution of linear recurrence relations by various methods.*
2. *To introduce basics of group theory and its applications.*
3. *Bring awareness of basic concepts of graphs and its applications.*

COURSE OUTCOMES

1. *Have acquired ability to participate effectively in group discussions*
2. *Have developed ability in writing in various contexts*
3. *Have acquired a proper level of competence for employability*
4. *Be able to apply basic counting techniques to solve combinatorial problems*
5. *Be able to specify and manipulate basic mathematical objects such as sets, functions, and relations*

UNIT -I

Mathematical Logic : Connectives, negations, conjunctions, disjunctions, conditional and bi-conditional - Well-formed formulae – Tautologies - Equivalence of formulae - Tautological implications- Principal disjunctive and conjunctive normal forms - Inference calculus - Rules of inference - Indirect method of proof.

UNIT -II

Recurrence Relations: Recurrence relations - Solving linear recurrence relations by characteristic roots method - System of recurrence relations - Non-linear recurrence relations.

UNIT- III

Group Theory: Binary operations - Semi-group – Monoid – Group - Abelian group – Subgroup - Lagrange‟s theorem on finite groups - Normal subgroups.

UNIT-IV

Graph Theory: Definitions - Finite and infinite graphs - Incidence and degree - Isolated pendant vertices - Isomorphism - Sub graphs – Walk-Path and circuit - Connected and disconnected graphs - Components - Euler graph , Euler graph theorem - Operations on graphs - Decomposition of Euler graphs into circuits.

UNIT- V

Trees: Some properties of tress - Pendant vertices - Distance and centers - Rooted and binary trees - Spanning trees - Fundamental circuit - Shortest spanning trees - Kruskal‟s lagorithm.

TEXT BOOKS

1. *Discrete Mathematics with Applications*, Thomas Koshy, Academic Press,First Edition, 2006
2. *Discrete Mathematical Structures with Application to Computer Science*, J.P. Tremblay & R. Manohar, McGraw Hill Publication, 39th Reprint, 2011.

REFERENCES

1. *Discrete Mathematics for Computer Scientists & Mathematicians*, J.L.Mott, A. Kandel, T.P. Baker, PHI Ptd, 2nd Edition, 2001.
2. *Discrete Event System Simulation*, Jerry Banks, PHI Ptd, 3rd Edition, 2003 .
3. *Discrete Mathematical Structures*, Bernand Kolman, Robert C. Busby, Sharon Cutler Ross, Pearson Education, 6th Edition, 2009.

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

MCA I Year – I Semester

L	T	P	C
3	1	-	4

(19MC9101) COMPUTER PROGRAMMING AND PROBLEM SOLVING

COURSE OBJECTIVES

1. *To make the student learn a programming language.*
2. *To make student understand the syntax and semantics of C programing language and other features of the language.*
3. *Understand the basic concept of C Programming, and its different modules that include conditional and looping expressions, Arrays, Strings, Functions, Pointers, Structures and File programming.*
4. *File Handling for permanent storage of data or record.*
5. *Applications of Self- referential structure and Unions.*

COURSE OUTCOMES

1. *Students will be able to apply problem solving techniques in designing the solutions for a wide range of problems.*
2. *Students will be able to write, compile and debug programs in C language.*
3. *Students will be able to design programs involving decision structures, loops and functions.*
4. *Students will be able to explain the difference between call by value and call by reference*
5. *Students will be able to understand the dynamics of memory by the use of pointers.*

UNIT – I

Algorithms and Flow Charts: Definition of Algorithms and examples - Symbols used in Flow chart and examples.

The C Declarations: The C character Set – Delimiters - The C Keywords - Identifiers – Constants – Variables - Rules for defining variables - Data Types - Declaring variables - Initializing variables - Type Conversion - Constant and Volatile variables.

UNIT – II

Operators and Expressions: Introduction - Priority of operators and their clubbing - Comma and Conditional Operator - Arithmetic Operators - Relational Operators - Logical Operators - Bitwise Operators. Input and Output in C - Introduction - Formatted Functions - Unformatted Functions - Commonly use Library Functions.

Decision Statements: Introduction - The if Statement - The if .. else Statement - Nested if .. else Statement - The break - continue and goto Statements - The switch Statement. Loop Control Statements - Introduction - the for Loop - Nested for Loop - The while Loop - The do-while Statement.

UNIT – III

Arrays: Introduction - Array Initialization - Definition of Array - Characteristics of Array - One Dimensional Array - Two Dimensional and Multi-Dimensional Arrays - The sscanf() and sprint() Functions - Recursive Functions - Terminating Condition.

Strings: Introduction - Declaration and Initialization of Strings - Display of Strings with Different Formats - String Standard Functions.

UNIT – IV

Pointers: Introduction - Features of pointers - Pointers Declaration - Arithmetic Operations with Pointers - Pointers and Arrays - Array of Pointers - Pointer to Pointers - Pointers and Strings - Void Pointer - Dynamic Memory Allocation-malloc and calloc functions.

Functions: Introduction - Definition of Function - Function declaration and Prototype - The Return Statement - Types of Functions - Call by Value and Reference. Storage Classes - Introduction - Automatic Variables - External Variables - Static Variables - Register Variables.

UNIT – V

Preprocessor Directives: Introduction - The #define Directive - Undefining a Macro - The #include Directive - The #ifndef Directive - The #error Directive - The #line Directive - The #pragma Directive.

Structure and Union: Introduction - Features o Structures - Declaration and Initialization of Structures - Accessing Structures - Nested Structures - Arrays of Structures - Structures and Functions - Pointers to Structures - typedef - Bit fields - Enumerated Data Type - Union - Union of Structures. Input and Output- Files - Streams - Standard library Input Output Functions - Character Input Output Functions.

TEXT BOOKS

1. *“C and Data Structures”,* Ashok N. Kamthane, Pearson Education.
2. *“Computer Fundamentals and C Programming”,* Dr. P. Chenna Reddy, First Edition.

REFERENCES

1. *The C Programming Language*, Brian W. Kernighan, 2015
2. *The C Programming Language*, Kernighan BW and Ritchie DM, 2nd Edition, Prentice Hall of India, 2006
3. *Problem Solving and Program Design in C*, J.R. Hanly and E.B. Koffman, Pearson Education, 2007.
4. *Computer Programming and IT*, Ashok N. Kamthane et. al., Pearson Education, 2011.

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

MCA I Year – I Semester

L	T	P	C
3	-	-	3

(19MC9102) 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.*
4. *To understand the importance of the hardware-software interface.*

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 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.

Multi processors: 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	1.5

(19HS0811) COMMUNICATIVE ENGLISH LAB

COURSE OBJECTIVES

1. *Students will be exposed to a variety of self instructional, learner friendly modes of language learning.*
2. *Students will cultivate the habit of reading passages from the computer monitor. Thus providing them with the required facility to face computer based competitive exams like GRE, TOEFL, and GMAT etc.*
3. *Students will learn better pronunciation through stress, intonation and rhythm.*
4. *Students will be trained to use language effectively to face interviews, group discussions, public speaking.*
5. *Students will be initiated into greater use of the computer in resume preparation, report writing, format making etc.*

COURSE OUTCOMES

1. *Remember and understand the different aspects of the English language proficiency with emphasis on LSRW skills.*
2. *Apply communication skills through various language learning activities.*
3. *Analyze the English speech sounds, stress, rhythm, intonation and syllable division for better listening and speaking comprehension.*
4. *Evaluate and exhibit acceptable etiquette essential in social and professional Settings.*
5. *Create awareness on mother tongue influence and neutralize it in order to improve fluency in spoken English.*

UNIT – I

Part-1

Introduction to Phonetics

Part-2

Word Stress- Intonation

UNIT – II

Part-1

JAM - Oral Presentation

Part-2

Describing objects/places/persons- Minutes of Meeting

UNIT – III

Part-1

Situational dialogues – Greeting and Introduction - Telephonic Conversations

Part-2

Book Review-Report Writing

UNIT – IV

Part-1

Non-verbal Communication – Dumb Charade

Part-2

Debate/Group Discussion- Movie Review- Reading Comprehension.

UNIT – V

Part-1

Information Transfer

Part-2

Job Application and Resume Writing - Interview Skills

Suggested Software:

Walden InfoTech Software

REFERENCES

1. *A Textbook of English Phonetics for Indian Students*, T. Balasubramanian, Mcmillian second edition, 2012.
2. *A Course in Phonetics and spoken English,* DhamijaSethi, Prentice-hall of India Pvt. Ltd, 2000.
3. *Speaking English Effectively,* Krishna Mohan & NP Singh, Mcmillian, second Edition, 2011.
4. *A Hand Book of English Laboratories,* E.Sureshkumar & P.Sreehari Foundation books, 2011.
5. *Effective Technical Communication,* M Ashraf Rizvi, McGraw Hill Education, Second edition ,27 July 2017.

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

MCA I Year – I Semester

L	T	P	C
-	-	3	1.5

(19MC9103) C PROGRAMMING LAB

COURSE OBJECTIVES

1. *To make the student learn a programming language.*
2. *To teach the student to write programs in C to solve typical problems.*
3. *To make student solve problems, implement them using C language.*

COURSE OUTCOMES

Upon completion of the subject, students will be able to

1. *Write, compile and debug programs in C language.*
2. *Apply Problem solving techniques to find solutions to problems.*
3. *Ale to use C language features effectively and implement solutions using C language.*
4. *Able to improve logical skills.*
5. *Design programs involving decision structures, loops and functions.*
6. *Explain the difference between call by value and call by reference*
7. *Understand the dynamics of memory by the use of pointers.*
8. *Design programs involving files.*

Recommended Systems/Software Requirements:

Intel based desktop PC with ANSI C Compiler and Supporting Editors

<u>List of Sample Examples/ Experiments</u>

Exercise l.

a) Practice Programs: Finding the sum of three numbers, Exchange of two numbers, Maximum of two numbers, To read and write values of all data types of C language, To find the size of all data types.
b) Write a C program to find the sum of individual digits of a positive integer.
c) 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.
d) Write a C program to generate all the prime numbers between 1 and n, where value of n is supplied.
e) Write a C Program to
   i) Find whether the given number is palindrome or not.
   ii) Find whether the given number in Armstrong or not.
   iii) Find whether the given number is perfect or not.
   iv) Find whether the given number is prime or not.

Exercise 2

a) Write a C program to calculate the following
   i) Sum:Sum=$1-x^2/2! +x^4/4!-x^6/6!+x^8/8!-x^{10}/10!$
   ii) Sin x and Cos x values using series expansion

b) Write a C program to find the roots of a quadratic equation.

c) Write a C program that uses both recursive and non-recursive functions
   - i) To find the factorial of a given integer.
   - ii) To find the GCD (greatest common divisor) of two given integers.
   - iii) To solve Towers of Hanoi problem.

d) Conversion of Binary to Decimal, Octal, Hex-Decimal and vice versa.

Exercise 3

a) Write a C program to perform the following
   - i) Find both the largest and smallest number in a list of integers.
   - ii) Generate Identity matrix for the given odd number.
   - iii) Generate Upper and Lower triangle matrices for the given matrix.
   - iv) Generate Magic square matrix for the given odd number.

b) Write a C program that uses functions to perform the following:
   - i) Addition of Two Matrices
   - ii) Multiplication of Two Matrices

c) Programs for Bubble Sort, Selection Sort, Insertion Sort

d) Programs on Linear Search and Binary Search

e) Write a C program that uses functions to perform the following operations:
   - i) To insert a sub-string in to a given main string from a given position.
   - ii) To delete n Characters from a given position in a given string.

f) Write a C program to determine if the given string is a palindrome or not

g) Write a C program to sort given list of strings.

Exercise 4

a) Write a C program that displays the position or index in the string S where the string T begins, or – 1 if S doesn‟t contain T.

b) Write a C program to count the lines, words and characters in a given text.

c) Write a C program to generate Pascal‟s triangle.

d) Write a C program to construct a pyramid of numbers.

Exercise 5

a) Write a C program which copies one file to another.

b) Write a C program to reverse the first n characters in a file.
   (Note: The file name and n are specified on the command line.)

c) Write a C programme to display the contents of a file.

d) Write a C programme to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file)

REFERENCES

1. *The Spirit of C, an introduction to modern programming*, M.Cooper, Jaico Publishing House.
2. *Mastering C*, K.R. Venugopal and S.R. Prasad, TMH Publications.
3. *Computer Basics and C Programming*, V. Rajaraman, PHI Publications.

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

MCA I Year – I Semester

L	T	P	C
-	-	4	2

(19MC9104) 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&10: 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 11&12: 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 13&14: 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,5th edition,TMH

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

MCA I Year – II Semester

L	T	P	C
3	-	-	3

(19MB9056) ACCOUNTING & FINANCIAL MANAGEMENT

COURSE OBJECTIVES

1. *Explain financial accounting terminology and the recording process.*
2. *Discuss and apply the basic principles of accounting, the accounting model, and the accounting cycle.*
3. *Analyze a cash flow statement & Analyze financial statements using ratio analysis*

COURSE OUTCOMES

1. *Develop the ability to use the fundamental accounting equation to analyze the effect of Business transactions on an organization's accounting records and financial statements*
2. *Demonstrate the applicability of the concept of Financial Management to understand its objectives and role of a Financial Manager*
3. *Equip with the knowledge of accounting process and preparation of final accounts of sole trader*
4. *Demonstrate the identification of importance of capital structure and the sources of finance*
5. *Demonstrate the ability to apply cost volume profit analysis by managers to answer various operating decisions, such as what level sales required to break even, how many units of products are to be sold in order to earn a target level of profit*

UNIT-I

Introduction to Accounting: Principles, concepts and conventions, double entry system of accounting, classification of accounts and debit-credit rules.

Financial Statements: Introduction to basic books of accounts, journal and ledger – trial balance – preparation of final accounts: trading account, profit and loss account and balance sheet.

UNIT-II

Introduction to Financial Management: Meaning and scope, role of financial manager - objectives of time value of money – goals of financial management, leverages: operation - financial leverage and combined leverage.

Capital Structure: Cost of capital: cost of equity, preference shares, bonds – weighted average cost of capital – capital gearing – overcapitalization and undercapitalization - sources of finance.

UNIT-III

Financial Analysis through ratios: Ratio Analysis – classification of ratios – short term solvency and long term solvency – profitability ratios – analysis and interpretation of financial statements through ratios of liquidity - solvency and profitability.

UNIT-IV

Funds Flow and Cash Flow Analysis: Meaning, Importance, statement of changes in working capital - statement of sources and application of funds. Cash flow analysis - cash flow statements - preparation, analysis and interpretation.

Break Even Analysis: Concept of Break Even Point - Cost-Volume-Profit Analysis - Determination of Break Even Point - Margin of Safety and P/V ratio - Impact of changes in cost or selling price on BEP - Practical applications of Break Even Analysis.

UNIT-V

Capital Budgeting: Capital and its significance - types of capital - estimation of fixed and working capital requirements - methods and sources of raising capital. Capital budgeting: features – proposals - methods of capital budgeting - payback method - accounting rate of return (AAR) - Net Present Value Method(NPV) and Internal Rate of Return (IRR) - simple problems.

TEXT BOOKS

1. *Financial Accounting*, S.N.Maheshwari, Sultan Chand, 2009.
2. *Financial Management and Policy*, Van Horne, James,C., Pearson ,2009.

REFERENCES

1. *Financial Accounting*, Tulsian, S Chand, 2009.
2. *Financial Statement Analysis*, Khan and Jain, PHI, 2009
3. *Financial Management*, I.M.Pandey, Vikas Publications
4. *Financial Management*, BhatSundhindra, Excel: 2009
5. *Financial Management*, Prasanna Chandra, T.M.H, 2009.

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

MCA I Year – II Semester

L	T	P	C
3	-	-	3

(19MC9105) 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. *Have the ability to write a computer program to solve specified problems.*
4. *Be able to use the Turbo C++ environment to create, debug and run simple C++ programs.*
5. *Understand the fundamentals of Inheritance & Exceptions Handling concepts.*

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 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 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 – II Semester

L	T	P	C
3	-	-	3

(19MC9106) 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 choose the appropriate data structure for a specified application.*
4. *To understand and analyze various searching and sorting algorithms.*
5. *To write programs in C++ to solve problems using data structures such as array, linked lists, queues, trees, graphs, hash tables, search trees.*

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, 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++*, Sartaj Sahni, 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 – II Semester

L	T	P	C
3	-	-	3

(19MC9107) 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 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 services, Operating System Structure and operations - 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 – II Semester

L	T	P	C
3	-	-	3

(19MC9108) SYSTEM SOFTWARE

COURSE OBJECTIVES

1. *To understand the relationship between system software and machine architecture.*
2. *To know the design and implementation of assemblers.*
3. *To know the design and implementation of linkers and loaders.*
4. *To have an understanding of macroprocessor.*
5. *To have an understanding of system software tools.*

COURSE OUTCOMES

1. *Able to differentiate different instruction formats among machines.*
2. *To have an understanding of foundations of design of assemblers.*
3. *Able to distinguish between loaders and linkers.*
4. *Able to develop the own source code of Macro processor.*
5. *To have an understanding of issues in device drivers.*

UNIT – I

Introduction: system software and machine architecture - SIC, RISC, and CISC architecture.
Assembler: Basic Assembler functions - machine dependent and independent assembler features - assembler design options.

UNIT – II

Loading and linkers: Basic loader function - machine dependent and independent loader features - loader design options.
Macro processors: basic macro processor functions machines – independent macro processor features - macro processor design options.

UNIT – III

Compilers: Basic Compiler functions - machine dependent and independent compiler features - compiler design options.
Other system software: Text Editors – overview of editing process - user interface and editor structure. Interactive Debugging Systems – Debugging Functions and capabilities - User-Interface criteria.

UNIT – IV

Device Drivers: Introduction - Types of Device. Character Driver I: A Test Data Generator - Design issues - Driver. Character Driver II: A/D converter - Design Issues - Driver.
Block Drivers: Block Driver I: A Test Data Generator - Design Issues - Driver. Block Driver II: A RAM Disk - Design Issues - Driver.

UNIT – V

Linux: Introduction to Linux- Linux Architecture- Linux administration tools - Commands to use Linux OS
Shell Scripts: Executing Linux Shell scripts – Shell Programming concepts-Shell scripts.

TEXT BOOKS

1. *System Software: An Introduction to systems programming*, Leland .Beck, 3/e, Pearson Educations Asia, 2003.
2. *Writing Unix Drivers*, George pajari, Addison – Wesley,1991.
3. *Linux complete Reference*, Richard Petersen, McGraw Hill Education (India) Private Limited; 6 edition (21 November 2007)

REFERENCES

1. *System programming and operation Systems*, Dhamdhere, 2/E, Tata Mc Graw, Hill, 1999
2. *Compilers, Techniques and Tools*, A.V. Aho, Ravi Sethi and J D Ullman, Addison Wesley, 1986.
3. System Programming, Jhon J. Donovan, Tata Mc Graw Hill 2005.

MCA I Year – II Semester

L	T	P	C
-	-	4	2

(19MC9109) PROGRAMMING IN C++ LAB

COURSE OBJECTIVES

1. *To understand C++ and object-oriented concepts.*
2. *To write, debug, and document well-structured C++ applications*
3. *To understand decision and iteration control structures to implement algorithms*
4. *To write simple recursive algorithms*
5. *To implement interfaces, inheritance, and polymorphism as programming techniques*
6. *To apply exceptions handling.*

COURSE OUTCOMES

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

1. *Understand programming language concepts, particularly C++ and object-oriented concepts.*
2. *Write, debug, and document well-structured C++ applications*
3. *Effectively create and use objects from predefined class libraries*
4. *Understand the behavior of primitive data types, object references, and arrays*
5. *Apply decision and iteration control structures to implement algorithms.*
6. *Implement interfaces, inheritance, and polymorphism as programming techniques.*

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 a recursive function for solving Towers of Hanoi problem.
6. Write a **C++** program that uses functions
   a) To swap two integers.
   b) To swap two characters.
7. Write a **C++** program to find both the largest and smallest number in a list of integers.
8. Write a **C++** program to sort a list of numbers in ascending order.
9. Write a **C++** program that uses function templates to solve problems-7&8.
10. Write a **C++** program to sort a list of names in ascending order.
11. 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.

12. Write a template based **C++** program that determines if a particular value occurs in an array of values.
13. 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.
14. Write a **C++** program that uses a function to reverse the given character string in place, without any duplication of characters.
15. Write a **C++** program to make the frequency count of letters in a given text.
16. Write a **C++** program to count the lines, words and characters in a given text.
17. Write a **C++** program to determine if the given string is a palindrome or not.
18. Write a **C++** program to make frequency count of words in a given text.
19. Write a **C++** program to generate Pascal‟s triangle.
20. Write a **C++** program to construct of pyramid of numbers.
21. Write a **C++** program to display the contents of a text file.
22. Write a **C++** program which copies one file to another.
23. Write a **C++** program that counts the characters, lines and words in the text file.
24. 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.
25. Write a **C++** program to reverse the first n characters in a file.
26. Write a **C++** program that uses a function to delete all duplicate characters in the given string.
27. Write a **C++** program that uses a function to convert a number to a character string.
28. Write a **C++** program that uses a recursive function to find the binary equivalent of a given non-negative integer n.
29. Write a **C++** program
    a) To write an object to a file. b) To read an object from the file.
30. 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
31. 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.
32. Write a **C++** program that illustrates how run time polymorphism is achieved using virtual functions.
33. Write a **C++** program that illustrates the role of virtual base class in building class hierarchy.
34. Write a **C++** program that illustrates the role of abstract class in building class hierarchy.
35. Write a **C**++ program for Factorial using recursion.

REFERENCES

1. *Mastering C++*, K.R.Venu Gopal, 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
7. *Fundamentals of C++ Programming*,S.Subramanian,Jaico Publishing House.
8. *C++ Programming*, Al Stevens, 7th edition,Wiley India.

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

MCA I Year – II Semester

L	T	P	C
-	-	4	2

(19MC9110) DATA STRUCTURES THROUGH 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 choose the appropriate data structure for a specified application.*
4. *To understand and analyze various searching and sorting algorithms.*
5. *To write programs in C++ to solve problems using data structures such as array, linked lists, queues, trees, graphs, hash tables, search trees.*

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, 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. *C++ Plus 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
-	-	4	2

(19MC9111) OPERATING SYSTEMS LAB

COURSE OBJECTIVES

1. *To understand the services provided by and to design an operating system.*
2. *To understand what a process is and how processes are scheduled.*
3. *To understand what a process is and how processes are synchronized*
4. *To understand different approaches to memory management.*
5. *To understand the structure and organization of the file system.*

COURSE OUTCOMES

Upon completion of this course the students should:

1. *Understand process management, concurrent processes and threads*
2. How to allocate and free memory
3. Able to solve deadlock if occur
4. Compare performance of processor scheduling algorithms
5. Produce algorithmic solutions to process synchronization problems

List of Operating Systems Programs:

1. Simulate FCFS CPU scheduling algorithms
2. Simulate SJF CPU scheduling algorithms
3. Simulate Priority CPU scheduling algorithms
4. Simulate Round Robin CPU scheduling algorithms
5. Simulate all File Organization Techniques
   a) Single level directory b) Two level c) Hierarchical
6. Simulate Bankers Algorithm for Dead Lock Avoidance
7. Write a C program to create a child process and allow the parent to display “Hello” and the child to display “Welcome” on the screen.
8. Simulate FIFO page replacement algorithms.
9. Simulate LRU page replacement algorithms.
10. Simulate LFU page replacement algorithms.
11. Simulate Paging Technique of memory management.
12. Write C programs that make a copy of a file using Standard I/O
13. Write C programs that make a copy of a file using System Calls.
14. Write C programs that count the number of blanks in a text file using Standard I/O
15. Write C programs that count the number of blanks in a text file using System Calls.

REFERENCES

1. *Operating Systems*, P.P. Choudhury, PHI Learning Private Ltd.
2. *Operating Systems*, R.Chopra, S.Chand and Company Ltd
3. *Operating System Concepts* (2012), Silberschatz, Galvin and Gagne.

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

MCA II Year – I Semester

L	T	P	C
3	-	-	3

(19MC9112) COMPUTER NETWORK

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*
4. *To Acquire knowledge of various application protocol standard developed for Internet*

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 II Year – I Semester

L	T	P	C
3	-	-	3

(19MC9113) 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.*
4. *Understand the fundamentals of Java collection frame work, Exceptions and multi-threading concepts.*
5. *Understand the GUI concepts like Applets, Swings.*

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. *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 and TreeMap, 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- Jbutton, JToggleButton, JCheckBox, JRadioButton, JLabel, JTextField, JTextArea, JList, JComboBox, JMenu, 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: Inheritance hierarchy for 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

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

MCA II Year – I Semester

L	T	P	C
3	-	-	3

(19MC9114) 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.*
4. *To impart knowledge in transaction processing, concurrency control techniques and recovery procedures.*

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. *Develop a simple database applications using normalization.*
5. *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 II Year – I Semester

L	T	P	C
3	-	-	3

(19MC9115) ADVANCED PROGRAMMING (PYTHON & R LANGUAGES)

COURSE OBJECTIVES

1. *Introduction of Scripting Language*
2. *Exposure to various problem solving approaches of computer science*
3. *Understand the fundamentals of 'R' programming*
4. *Learn how to carry out a range of commonly used statistical methods including analysis of variance and linear regression.*
5. *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. *To build software for real needs.*
4. *Prior Introduction to testing software*
5. *Ability to Work on a real life Project, implementing R Analytics to create Business Insights.*
6. *Ability to analyze the data and results using R, a flexible and completely Cross - platform.*
7. *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 LocalVariables. 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*, Vamsi Kurama, 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	-	-	3

(19MC9116) DESIGN AND ANALYSIS OF ALGORITHMS

COURSE OBJECTIVES

1. *To know the importance of the complexity of a given algorithm.*
2. *To study various algorithm design techniques.*
3. *To utilize data structures and/or algorithmic design techniques in solving new problems.*
4. *To know and understand basic computability concepts and the complexity classes P, NP, and NP-Complete.*
5. *To study some techniques for solving hard problems.*

COURSE OUTCOMES

1. *Analyze the complexity of the algorithms*
2. *Use techniques divide and conquer, greedy to solve the problems.*
3. *Able to solve problems on dynamic programming, backtracking, branch and bound.*
4. *Identify and analyze criteria and specifications appropriate to new problems, and choose the appropriate algorithmic design technique for their solution.*
5. *Able to prove that a certain problem is NP-Complete.*

UNIT – I

Introduction: What is an Algorithm - Algorithm specification - Performance analysis.
Divide and Conquer: General method - Binary Search - Finding the maximum and minimum - Merge sort - Quick Sort - Selection sort - Stressen„s matrix multiplication.

UNIT – II

Greedy Method: General method - Knapsack problem - Job Scheduling with Deadlines - Minimum cost Spanning Trees - Optimal storage on tapes - Single-source shortest paths.
Dynamic programming: General Method - Multistage graphs - All-pairs shortest paths - Optimal binary search trees - 0/1 knapsack - The traveling sales person problem.

UNIT – III

Basic Traversal and Search Techniques: Techniques for binary trees - Techniques for Graphs - Connected components and Spanning trees - Bi-connected components and DFS
Back tracking: General Method - 8 – queens problem - Sum of subsets problem - Graph coloring and Hamiltonian cycles - Knapsack Problem.

UNIT – IV

Branch and Bound: The method - Travelling salesperson - 0/1 Knapsack problem - Efficiency Considerations.

Lower Bound Theory: Comparison trees - Lower bounds through reductions – Multiplying triangular matrices - inverting a lower triangular matrix - computing the transitive closure.

UNIT – V

NP – Hard and NP – Complete Problems: NP Hardness - NP Completeness - Consequences of being in P - Cook„s Theorem - Reduction Source Problems - Reductions: Reductions for some known problems

TEXT BOOKS

1. *"Fundamentals of Computer Algorithms"*, Ellis Horowitz, S. Satraj Sahani and Rajasekhran, 2nd edition, University Press.2014,
2. *"Design and Analysis of Algorithms"*, Parag Himanshu Dave, Himanshu Bhalchandra Dave, Pearson Education, Second Edition, 2009.

REFERENCES

1. "Introduction to Algorithms", T.H.Cormen, C.E.Leiserson, R.L.Rivest and C.Stein, PHI Pvt. Ltd. second edition, Pearson Education.
2. *"Introduction to Design and Analysis of Algorithms A strategic approach"*, R.C.T.Lee, S.S.Tseng, R.C.Chang and T.Tsai, Mc Graw Hill.
3. *"Data structures and Algorithm Analysis in C++"*, Allen Weiss, Second edition, Pearson education.
4. *"Design and Analysis of algorithms"*, Aho, Ullman and Hopcroft,Pearson education.
5. *"Algorithms",* Richard Johnson baugh and Marcus Schaefer, Pearson Education

R19

MCA

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

MCA II Year – I Semester

L	T	P	C
-	-	4	2

(19MC9117) 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 write simple recursive algorithms*
4. *To implement interfaces, inheritance, and polymorphism as programming techniques*
5. *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. *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 II Year – I Semester

L	T	P	C
-	-	4	2

(19MC9118) 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.*
4. *The course concludes with an overview of transaction management and introduction to advanced and non-relational databases.*

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 and Relational 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 principles of database transaction management, database recovery, security.*

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 II Year – I Semester

L	T	P	C
-	-	4	2

(19MC9119) ADVANCED PROGRAMMING LAB

COURSE OBJECTIVES

1. *To design experiments and working with WEKA tool.*
2. *The course is designed to provide Basic knowledge of Python & R.*
3. *Python & R programming is intended for software engineers, system analysts, program managers and user support personnel who wish to learn the Python & R programming language.*

COURSE OUTCOMES

1. *Design experiments using WEKA tool for Data mining applications*
2. *Able to write programs on operations using Python and R Programs.*
3. *Able to solve programs on Data Structures using Python and R*
4. *Ability to work on Files*
5. *Ability to work on Functions.*

List of Sample Programs/Experiments

S.No. Name of the Experiment

1 Credit Risk Assessment:

1) Find a loan officer who is willing to talk in the form of production rules.
2) Find some training manuals for loan officers and translate text into production rules.
3) Judging the credit worthiness of a loan applicant.
4) Find records of actual cases where competent loan officers correctly judged.

2 The German Credit Data:

1. List all the categorical (or nominal) attributes and the real-valued attributes separately.
2. What attributes do you think might be crucial in making the credit assessment? Come up with some simple rules in plain English using your selected attributes.

3 3. Train a Decision Tree using the complete dataset as the training data

4. 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. Describe what is cross validation briefly. Train a Decision Tree again using cross validation and report your results.

4 7. Comparing accuracy of Training using full features and training by removing some features.

8. Compare few combinations of attributes in training and describe the accuracy.
9. 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). 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. How does the complexity of a Decision Tree relate to the bias of the model for complex and simple decision trees?

5 11. Train the dataset using Reduced Error Pruning and describe the accuracy.

12. Report the rule obtained by training a one R classifier. Rank the performance of j48, PART and one R.

6 **Exercise 1** - Basics

a) Running instructions in Interactive interpreter and a Python Script

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

Exercise 2 - Operations

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

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

7 **Exercise - 3** Control Flow

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

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

Exercise - 4 - DS

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

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

8 **Exercise - 5** Files

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

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

Exercise - 6 Functions

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

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

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

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

e) Write function to compute gcd, lcm of two numbers. Each function shouldn‟t exceed one line.

9 1. Write a R program to take input from the user (name and age) and display the values. Also print the version of R installation.

2. Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.

3. Write a R program to create a vector which contains 10 random integer values between -50 and +50.

10 4. Write a R program to get the first 10 Fibonacci numbers.

5. Write a R program to get all prime numbers up to a given number (based on the sieve of Eratosthenes).

6. Write a R program to extract first 10 english letter in lower case and last 10 letters in upper case and extract letters between 22nd to 24th letters in upper case.

11 7. Write a R program to find the factors of a given number

8. Write a R program to find the maximum and the minimum value of a given vector

9. Write a R program to get the unique elements of a given string and unique numbers of vector.

12 10. Write a R program to draw an empty plot and an empty plot specify the axes limits of the graphic.

11. Write a R program to create a simple bar plot of five subjects marks.

12. Write a R program to create a Data frames which contain details of 5 employees and display the details.

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 – II Semester

L	T	P	C
3	-	-	3

(19MC9120) 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*
4. *To identify Business applications and Trends of Data mining*

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. *Evolve Multidimensional Intelligent model from typical system*
5. *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 - Based Methods - 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.

R19 MCA

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

MCA II Year – II Semester

L	T	P	C
3	-	-	3

(19MC9121) WEB TECHNOLOGIES

COURSE OBJECTIVES

1. *To understand the basics of java bean*
2. *To understand the web server and the server-side programming*
3. *To understand the DB connections and MVC architecture, JSP*
4. *To understand the concepts of PHP and AJAX*

COURSE OUTCOMES

Student is able to:

1. *Design and execute applications in java beam*
2. *Do the server side programming, maintain sessions.*
3. *Establish the DB connections and access the data.*
4. *Ability to work on MVC architecture*
5. *Design pages using PHP and AJAX.*

UNIT-I

Java Beans: Introduction to Java Beans - Advantages of Java Beans - BDK Introspection - Using Bound properties - Bean Info Interface - Constrained properties Persistence - Customizes - Java Beans API, and Introduction to EJBs

HTML Common tags- Introduction to HTML 5 - HTML 5 vs. previous version of HTML - List, Tables, images, forms, Frames, Layouts - Graphics – Canvas, SVG, Media. CSS.

JavaScript - Introduction to Java Script, JS data types - Built-in objects - Functions - Objects - User-defined objects - JS HTML Document Object Model - Browser object model - Dynamic HTML with Java Script - Introduction to JSON.

UNIT-II

Web Servers and Servlets: Tomcat web server - Installing the Java Software Development Kit - Tomcat Server & Testing Tomcat. Introduction to Servlets: Lifecycle of a Servlet - JSDK - The Servlet API - The javax.servlet Package - Reading Servlet parameters - Reading Initialization parameters. The javax.servlet HTTP package - Handling Http Request & Responses - Using Cookies-Session Tracking - Security Issues.

Databases – Connecting databases - Inserting - Retrieving - Updating the data - Statements – simple - Prepared - Batch - Callable Statements.

UNIT-III

Introduction to JSP: The Problem with Servlet. The Anatomy of a JSP Page - JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment.

JSP Application Development: Generating Dynamic Content - Using Scripting Elements Implicit JSP Objects - Conditional Processing – Displaying Values Using an Expression to Set an Attribute - Declaring Variables and Methods Error Handling and Debugging Sharing Data Between JSP pages - Requests and Users Passing Control and Date between Pages –

Sharing Session and Application Data – Memory Usage Considerations Accessing a Database from a JSP Page - Deploying JAVA Beans in a JSP Page - Introduction to struts framework.

UNIT-IV

PHP – Introduction - Data types - Flow of Control statements - Functions - Arrays - Forms - Handling Files - Error handling.

UNIT-V

AJAX – Introduction to AJAX - XMLHttp - Request - Response - Events - Database - XML - PHP using AJAX.

TEXT BOOKS

1. *The complete Reference Java*, Herbert Schildt. TMH.
2. *HTML5 and CSS3*, Elizabeth Castro and Bruce Hyslop, 7th edition,.
3. *Ajax: A beginner's guide*, steven Holzner.

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 – II Semester

L	T	P	C
3	-	-	3

(19MC9122) 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*

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, Witold Pedrycz, John Wiely.
3. *Systems Analysis and Design*- Shely Cashman Rosenblatt, Thomson Publications.
4. *Software Engineering principles and practice*- Waman S Jawadekar, The McGraw- Hill Companies.

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

MCA II Year – II Semester

L	T	P	C
3	-	-	3

(19MC9123) SOFTWARE TESTING
(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 learn the functionality of automated testing tools*
4. *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 - domains and 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 – II Semester

L	T	P	C
3	-	-	3

(19MC9124) ARTIFICIAL INTELLIGENCE
(ELECTIVE – I)

COURSE OBJECTIVES

1. *To know the concept of Artificial Intelligence.*
2. *To understand the searching strategies in AI.*
3. *To understand the Game Playing, Knowledge Based techniques.*
4. *To learn the Planning and Learning*
5. *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 solve the Game Playing problems.*
3. *Student can able to use to planning and learning techniques*
4. *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.*
5. *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 – II Semester

L	T	P	C
3	-	-	3

(19MC9125) DISTRIBUTED SYSTEMS
(ELECTIVE – I)

COURSE OBJECTIVES

1. *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. *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;*
3. *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;*
4. *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;*
5. *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 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;*
3. *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;*
4. *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*
5. *Able to build distributed software and higher – level middleware and languages.*

UNIT-I

Introduction: Characterization of Distributed Systems - System models - Networking and Internetworking – 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 RMI).

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 – II Semester

L	T	P	C
3	-	-	3

(19MC9126) LINUX PROGRAMMING
(ELECTIVE – I)

COURSE OBJECTIVES

1. *To understand the LINUX system structure.*
2. *To understand and use command line shell.*
3. *To make effective use of Unix utilities and Shell scripting language such as bash.*
4. *To produce programs similar to standard unix utilities such as ls, mv, cp etc. using Unix system calls.*
5. *To develop the skills necessary for Unix systems programming including file system programming, process and signal management, and inter-process communication.*
6. *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 shell script to automate different tasks as Linux administration.*
4. *Able to communicate with processes and memory.*
5. *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 - process states - 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 – II Semester

L	T	P	C
3	-	-	3

(19MC9127) NETWORK SECURITY
(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 security protocols for protecting data on networks*
4. *Understand vulnerability assessments and the weakness of using passwords for authentication*
5. *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 – II Semester

L	T	P	C
3	-	-	3

(19MC9128) HUMAN COMPUTER INTERACTION
(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 able to apply models from cognitive psychology to predicting user performance in various human-computer interaction tasks and recognize the limits of human performance as they apply to computer operation*
4. *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. *Use cognitive psychology in the design of devices for interaction*
4. *Able to apply models from cognitive psychology.*
5. *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

Controls: Characteristics of device based controls - Selecting the proper device based controls - 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* - Soren Lauesen - Pearson Education.
5. *Human –Computer Interaction* - Smith - Atakan - Cengage Learning.

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

MCA II Year – II Semester

L	T	P	C
3	-	-	3

(19MC9129) SOCIAL NETWORKS AND SEMANTIC WEB
(ELECTIVE – II)

COURSE OBJECTIVES

1. *To understand the need of semantic web in web services*
2. *To know the methods to discover, classify and build ontology for more reasonable results in searching*
3. *To build and implement a small ontology that is semantically descriptive of chosen problem domain*
4. *To implement applications that can access, use and manipulate the ontology*

COURSE OUTCOMES

1. *Understand semantic web basics, architecture and technologies*
2. *Able to represent data from a chosen problem in XML with appropriate semantic tags obtained or derived from the ontology*
3. *Able to understand the semantic relationships among these data elements using Resource Description Framework (RDF)*
4. *Able to design and implement a web services application that "discovers" the data and/or other web services via the semantic web*
5. *Able to discover the capabilities and limitations of semantic web technology for social networks*

UNIT-I

Web Intelligence: Thinking and Intelligent Web Applications - The Information Age - The World Wide Web - Limitations of Todays Web - The Next Generation Web - Machine Intelligence - Artificial Intelligence - Ontology - Inference engines - Software Agents - Berners-Lee www - Semantic Web Road Map - Logic on the semantic Web.

UNIT-II

Knowledge Representation for the Semantic Web Ontologies and their role in the semantic web: Ontologies Languages for the Semantic Web – Resource Description Framework(RDF) / RDF Schema - Ontology Web Language(OWL) -UML -XML/XML Schema. Ontology Engineering - Constructing Ontology - Ontology Development Tools - Ontology Methods - Ontology Sharing and Merging - Ontology Libraries and Ontology Mapping - Logic - Rule and Inference Engines.

UNIT-III

Semantic Web applications and services: Semantic Search - e-learning - Semantic Bioinformatics - Knowledge Base -XML Based Web Services - Creating an OWL-S Ontology for Web Services - Semantic Search Technology.

UNIT-IV

Social Network Analysis: What is networks analysis? - Development of Social Networks Analysis - Key concepts and measures in network analysis – The global structure of networks - The macro-structure of social networks - Personal networks.

Electronic Sources for Network Analysis: Electronic Discussion networks - Blogs and Online Communities - Web-based networks. Modeling and aggregating social network data. State-of-art in network data representation - Ontological representation of social individuals - Ontological representation of social relationships - Aggregating and reasoning with social network data.

UNIT-V

Developing social-semantic applications: Building Semantic Web Applications with social network features - Flink: the social networks of the Semantic Web community - Evaluation of web-based social network extraction.

Semantic-based Social Network Analysis in the sciences Methodology: Data acquisition - Representation - storage and reasoning - Visualization and Analysis - Results – Descriptive analysis - Structural and cognitive effects on scientific performance.

TEXT BOOKS

1. *Thinking on the Web* - Berners Lee - Godel and Turing - Wileyinterscience.
2. *Social Networks and the Semantic Web* - PeterMika -Springer.

REFERENCES

1. *Semantic Web Technologies* -Trends and Research in Ontology Based Systems - J.Davies - Rudi Studer - Paul Warren - JohnWiley& Sons.
2. *Semantic Web and Semantic Web Services* -Liyang Lu Chapman and Hall/CRC Publishers -(Taylor & Francis Group)
3. *Information Sharing on the semantic Web* - HeinerStuckenschmidt; Frank Van Harmelen - Springer Publications.
4. *Programming the Semantic Web* - T.Segaran - C.Evans - J.Taylor - O‟Reilly -SPD.

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

MCA II Year – II Semester

L	T	P	C
3	-	-	3

(19MC9130) COMPUTER GRAPHICS
(ELECTIVE – II)

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 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 clipping 3-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.

UNIT-V

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* - Zhigand xiang - Roy Plastock - Schaum"s outlines - *Second edition* - Tata McGraw hill edition.
2. *Procedural elements for Computer Graphics* - David F Rogers - Tata Mc Graw 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 II Year – II Semester

L	T	P	C
3	-	-	3

(19MC9131) INTERNET OF THINGS
(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 manage memory and performance of battery life*
5. *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 Embedded8 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, 1stEdition, 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 – II Semester

L	T	P	C
3	-	-	3

(19MC9132) E-COMMERCE
(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. *They can know the added value, risks and barriers in the adoption of e-Business and e-Commerce*
5. *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. Tharan Dillan and E.Chany, Wiley,2003.

REFERENCES

1. *Web Commrece Technology*, Danjel Minoli and Emuna Mimoli, 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

R19

MCA

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

MCA II Year – II Semester

L	T	P	C
-	-	4	2

(19MC9133) 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.*
2. *Their effective use in discovering interesting hidden patterns in large volume of data generated by businesses, science, web, and other sources.*
3. *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. *Ability to clustering data and pattern evaluation*
5. *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. (5 marks)

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. (5 marks)

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. (10 marks)

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? (10 marks)

5. Is testing on the training set as you did above a good idea? Why or Why not ? (10 marks)

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? (10 marks)

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. (10 marks)

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.) (10 marks)

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 marks)

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 ? (10 marks)

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 ? (10 marks)

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. (10 marks)

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 Hierarchies Dimension 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 – II Semester

L	T	P	C
-	-	4	2

(19MC9134) WEB TECHNOLOGIES LAB

COURSE OBJECTIVES

1. *To understand the basics of java bean*
2. *To understand the web server and the server-side programming*
3. *To understand the DB connections and MVC architecture, JSP*
4. *To understand the concepts of PHP and AJAX*

COURSE OUTCOMES

Student is able to:

1. *Design and execute applications in java beam*
2. *Do the server side programming, maintain sessions.*
3. *Establish the DB connections and access the data.*
4. *Ability to work on MVC architecture*
5. *Design pages using PHP and AJAX.*

List of Programs/Experiments

1. 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.
2. Write a servlet program to display “hello world”.
3. A servlet program to read the parameters and the initialization parameters.
4. User Authentication**:** Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a servlet for doing the following.
   a. Create a Cookie and add these four user id and passwords to this Cookie.
   b. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords) available in the cookies. If he is a valid user (i.e., user- name and password match) you should welcome him by name (user-name) else you should display “You are not an authenticated user “. Use init-parameters to do this. Store the user-names and passwords in the webinf.xml and access them in the servlet by using the getInitParameters() method.
5. Install a database. A servlet program to access the details of the table in the database. (Insert, Display, Update)
6. A JSP program to display the current system date and time.
7. A JSP program to access the details of the table in the database. (Insert, Display, Update)
8. A PHP program to display the form on the page.
9. A PHP program to access the database.
10. AJAX program to Change the content of the web page
11. AJAX program to retrieve data from XML
12. AJAX program to retrieve from the database.

**** Every student should design and develop a website that also contains 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

MCA II Year – II Semester

L	T	P	C
-	-	4	2

(19MC9135) SOFTWARE ENGINEERING & UML LAB

COURSE OBJECTIVES

1. *To understand the various phases of Spiral model.*
2. *To understand how to draw E-R diagram, DFD, UML diagrams for the project*
3. *To understand how to develop PERT and CPM project schedule methods.*
4. *To understand and analyze and prepare RMMM plan.*

COURSE OUTCOMES

1. *Able to prepare various phases of Spiral model.*
2. *Able to draw E-R diagram, DFD diagrams for the project*
3. *Able to draw UML diagrams for the project*
4. *Able to develop PERT and CPM project schedule methods.*
5. *Able to analyze and prepare RMMM plan.*

INDEX:

1. Studying various phases of Spiral Model.
2. Prepare SRS for the given problem.
3. Using COCOMO model estimate effort for the given problem.
4. Calculate effort using FP oriented estimation model
5. Analyze the Risk related to the project and prepare RMMM plan.
6. Develop Time-line chart and project table using PERT or CPM project scheduling methods.
7. Draw E-R diagram, DFD, UML diagrams for the project.
8. Design of the test cases.
9. Prepare FTR, version control and change control for software configuration items.

Assignment 1

Title: Studying various phases of Spiral Model

Objective: To get familiar with basic model used for software engineering.

Post Lab Assignment

1. Explain the spiral model and give its advantages over water fall model?
2. Compare waterfall and Spiral model.
3. Discuss Prototyping model.

Assignment 2

Title: Prepare SRS for the given problem.

Objective: To get familiar with preparing a document which is used before starting the project.

Post Lab Assignment

1. What is meant by software requirement definition? Elaborate on its importance
2. Explain varies steps involved in Requirement Engineering?

Assignment 3
Title: Using COCOMO model estimate effort for the given problem
Objective: To get familiar with estimation and comparing various estimation techniques.
Post Lab Assignment

1. Describe project metrics.
2. Compare FP-based and LOC-based estimation technique.

Assignment 4
Title: Calculate effort using FP oriented estimation model.
Objective: To study software estimation in early stages of software development.
Post Lab Assignment

1. Explain Metrics for small organizations?
2. Explain Metrics for software quality?

Assignment 5
Title: Analyze the Risk related to the project and prepare RMMM plan.
Objective: To study types of risk and preparing RMMM plan.
Post Lab Assignment

1. Explain Risk Identification?
2. Explain various risk strategies?

Assignment 6
Title: Develop Time-line chart and project table using PERT or CPM
Project: scheduling methods.
Post Lab Assignment

1. Explain degree of Rigor?
2. Explain various Adaptation Criteria?

Assignment 7
Title: Draw E-R diagram, DFD, UML diagrams for the project.
Objective: To understand actual system using analysis model.
Post Lab Assignment

1. Explain data modeling?
2. Compare Hatley and Pirabhai extensions and Ward and Mellor Extensions?

Assignment 8
Title: Design of the test cases.
Objective: To understand various testing techniques.
Post Lab Assignment

1. Compare testing and debugging?
2. Explain various system testing?

Assignment 9
Title: Prepare FTR. version control and change control for software configuration items.
Objective: To understand Software configuration management Software Engineering
Post Lab Assignment

1. Explain software configuration management with baselines SCIs?
2. Explain Identification of objects in the software configuration

REFERENCES

1. *Software Engineering*, Roger Pressman McGraw Hill Fifth edition
2. *Software Engineering*, Ian Somerville Pearson Education Sixth edition
3. *An Integrated Approach To Software Engineering*, Pankaj Jalote Narosa

R19

MCA

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

MCA III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9136) 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. *Address the core issues of cloud computing such as security, privacy and Interoperability*
5. *Design Cloud Services and Set a private cloud*

UNIT-I

Distributed Computing-An Introduction: Computing Trends - Distributed Computing-An Introduction
Distributed System Models: Grid Computing - Cluster Computing - Virtualization.

UNIT-II

Cloud Computing: What‟s Cloud Computing - Properties & Characteristics - Pros and cons of Cloud Development.
Cloud Platform Architectures: Amazon AWS - Microsoft Azure - Google App Engine - Google Map Reduce/Yahoo Hadoop - Eucalyptus - Nimbus - Open Stack.

UNIT-III

Cloud Service Models: Infrastructure as a Service (IaaS): Introduction to IaaS - Resouce Virtualization - Server - Storage - Network - Case studies.
Platform as a Service (PaaS): Introduction to PaaS -Cloud platform & Management - Computation - Storage - Case studies. Software as a Service (SaaS): Introduction to SaaS - Web services - Web 2.0 - Web OS.

UNIT-IV

Cloud Deployment Models: Deployment Models Introduction – Public Deployment Model - Private Deployment Model - Virtual Private Deployment Model - Hybrid Deployment Model - Community Deployment Model.

UNIT-V

Cloud Issues And Challenges: Organizational Readiness and Change management in cloud - Security in Cloud - Legal Issues in Cloud - and Product Readiness for Cloud Services. Simple application using simulator.

TEXT BOOKS

1. Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, "*Cloud computing: Principles and Paradigms"*, September 2010, John Wiley & Sons.
2. 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 III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9137) .NET TECHNOLOGIES

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 or C# using ADO.Net connectivity.*
3. *Performing Database operations for windows form.*
4. *Able to create a web applications.*
5. *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 and Executable – Metadata - JIT Compilation - Automatic Memory Management -Assemblies and Manifests - Intermediate Language (IL) - The CTS and CLS - CLR Execution.

Introduction to C#.Net Programming : A Demonstration of Visual C# - Common Elements in Visual C# - C# Core Language Features – Types – Classes – Structures – Enumeration - Inheritance - Interfaces Polymorphism - Arrays and Collections - Generics - Operator Overloading - Delegates and Events – Introduction to LINQ Programming - Exception Handling - MSIL Programming.

UNIT-II

Introduction To Visual basic .Net : Introduction to Visual Basic .NET- Modules- variables- error handling- Arrays, lists - collections – Files- directories- streams - Object serialization - Regular expressions – Threading - Assemblies and AppDomains - Reflection - Windows Forms applications and GDI+ - Windows Forms custom control creation - Windows services.

UNIT-III

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 in Windows Forms and web forms.

Introduction To Asp.Net: Introduction - Working in ASP.NET - ASP.NET Controls - Session & Cookies – Caching - Authentication & Authorization - Web User Controls - Working with Web Config file - Crystal Reports - Creating Setup and Deployment.

UNIT-IV

XML: Introduction to .NET and XML - Reading and Writing XML - Reading and Writing XML Data Using XmlReader and XmlWriter - Manipulating XML with DOM - XML Data Validation - XML DOM Object Model - Transforming XML Data with XSLT

UNIT-V

.NET Mobile: .NET Mobile Introduction - Mobile Example – Emulators – Forms – Events – Input – Validation – Lists – Selection – Images – Utilities

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*, Thiru Thangarathinam ,Wiley Publishing, Inc. 2006
4. *Building Microsoft® ASP.NET Applications for Mobile Devices*, Andy Wigley, Peter Roxburgh , Second Edition, Microsoft Press, 2003

MCA III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9138) 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*
4. *Program mobile applications for the Android operating system that use basic and advanced phone features*
5. *Deploy applications to the Android marketplace for distribution*

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. *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

Understanding the Components of a Screen: 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 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 III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9139) CYBER SECURITY (ELECTIVE – III)

COURSE OBJECTIVES

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

COURSE OUTCOMES

After learning the course the students should be able to:

1. *Understand cyber-attack and System Vulnerability Scanning.*
2. *Knowledge on cyber laws.*
3. *Able to protect themself and ultimately society from cyber-attacks.*
4. *Knowledge on Web application tools.*
5. *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 - 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 Martti Lehto
2. *Cyber Law & Cyber Crimes Simokified* by Adv. Prashant Mali

REFERENCES

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

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

MCA III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9140) SOFTWARE PROJECT MANAGEMENT (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. *Acquire knowledge and skills needed for the construction of highly reliable software project.*
5. *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 III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9141) NEURAL NETWORKS & FUZZY LOGIC (ELECTIVE – III)

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 teach about the concept of fuzziness involved in various systems. To provide adequate knowledge about fuzzy set theory.*
4. *To provide comprehensive knowledge of fuzzy logic control and adaptive fuzzy logic and to design the fuzzy control using genetic algorithm.*
5. *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 - 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 and operation of Fuzzy logic control systems: Design methodology and stability analysis of fuzzy 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 III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9142) BIG DATA ANALYTICS
(ELECTIVE – III)

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 various search methods and visualization techniques.*
4. *To learn to use various techniques for mining data stream.*
5. *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 HADOOP and Map Reduce technologies associated with big data analytics*
4. *Explore on Big Data applications Using Pig and Hive*
5. *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 – 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 III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9143) MODEL DRIVEN FRAMEWORK
(ELECTIVE – III)

COURSE OBJECTIVES

The course should enable the students to

1. *Understand the basic client server concepts.*
2. *Understand EJB architecture*
3. *Know EJB applications*
4. *Learn CORBA*
5. *Learn COM*

COURSE OUTCOMES

The student should be able to

1. *Have learnt the different types of server client concepts*
2. *Learn the design of EJB architecture*
3. *Deploy EJB for specific applications*
4. *Build an application using CORBA*
5. *Build an application using COM*

UNIT-I

Client / Server Concepts: Client server – File server – Database server – Group server – Object server – Web server – Middleware – General middleware – Service specific middleware – Client / server building blocks – RPC – Messaging – Peer-to-peer.

UNIT-II

EJB architecture: Overview of EJB software architecture – View of EJB – Conversation – Building and deploying EJB – Roles in EJB.
EJB Applications: EJB session beans – EJB entity beans – EJB clients – EJB deployment – Building an application with EJB.

UNIT-III

CORBA – Distributed systems – Purpose – Exploring CORBA alternatives – Architecture overview – CORBA and networking model – CORBA object model – IDL – ORB – Building an application with CORBA.

UNIT-IV

COM – Data types – Interfaces – Proxy and stub – Marshalling – Implementing server/ client – Interface pointers – Object creation – Invocation – Destruction – Comparison COM and CORBA.

UNIT-V

SOA: Defining SOA - Business value of SOA - SOA characteristics - Concept of a service, Basic SOA - Enterprise Service Bus (ESB) - SOA enterprise Software Models -Services

Other types of Middleware: Real-Time Middleware, Embedded Systems Middleware, Mobile Middleware, Oracle Fusion Middleware.

TEXT BOOKS

1. *The Essential Client / Server Survival Guide*, Robert Orfali, Dan Harkey and Jeri Edwards, Galgotia Publications Pvt. Ltd., 2002.
2. *Enterprise Java Beans*, Tom Valesky, Pearson Education, 2002.

REFERENCES

1. *COM and CORBA Side by Side*, Jason Pritchard, Addison Wesley, 2000
2. *C#*, Joel Murach,Anne Boehm, Murach,2012.
3. *Essential Software Architecture*, Ian Gorton, Springer, 2nd Edition, 2011.
4. *Inside CORBA*, Mowbray, Pearson Education, 2002.
5. *The Complete Book of Middleware*, Judith M. Myerson, Auerbach Publications, 1 edition, 2002.

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

MCA III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9144) BIO – INFORMATICS
(ELECTIVE – IV)

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. *Predict the secondary and tertiary structures of protein sequences.*

UNIT-I

Introduction to Bioinformatics: Definition and History and Applications of Bioinformatics - 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 III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9145) IMAGE PROCESSING
(ELECTIVE – IV)

COURSE OBJECTIVES

1. *The fundamentals of digital image processing*
2. *Understand image transform used in digital image processing*
3. *Understand image enhancement techniques used in digital image processing*
4. *Understand image restoration techniques and methods used in digital image processing*
5. *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 compress images using compression techniques used in digital image processing*
5. *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 III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9146) DESIGN PATTERNS
(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.*
4. *Draw a high level class diagram in UML for each pattern.*
5. *Implement this pattern in Java to a real world problem*

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. *Students can able draw class diagrams for different patterns.*
5. *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 III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9147) MACHINE LEARNING
(ELECTIVE – IV)

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. *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. Ethern Alpaydin, 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

R19

MCA

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

MCA III Year – I Semester

L	T	P	C
3	-	-	3

(19MC9148) ENTERPRISE RESOURCE PLANNING
(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. *With the knowledge of typical ERP systems, and the advantages and limitations of implementing ERP systems.*
4. *To comprehend the technical aspects of ERP systems*
5. *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 concepts of BPR, SCM and CRM.*
5. *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

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

MCA III Year – I Semester

L	T	P	C
-	-	4	2

(19MC9149) CLOUD COMPUTING LAB

COURSE OBJECTIVES

The student should be made to:

1. *Be exposed to tool kits for cloud environment.*
2. *Learn to run virtual machines of different configuration.*
3. *To design application on SaaS and PaaS*

COURSE OUTCOMES

1. *The student should be able to Design and Implement applications on the Cloud.*
2. *Ability to Design application on SaaS*
3. *Ability to Design application on PaaS*
4. *Use the cloud tool kits.*

List of Programs/Experiments

Programs on SaaS

1. Create an word document of your class time table and store locally and on the cloud with doc,and pdf format . (use **www.zoho.com** and ***docs.google.com)***
2. Create a spread sheet which contains employee salary information and calculate gross and total sal using the formula

   DA=10% OF BASIC
   HRA=30% OF BASIC
   PF=10% OF BASIC IF BASIC<=3000
   12% OF BASIC IF BASIC>3000
   TAX=10% OF BASIC IF BASIC<=1500
   =11% OF BASIC IF BASIC>1500 AND BASIC<=2500
   =12% OF BASIC IF BASIC>2500
   (use **www.zoho.com** and ***docs.google.com)***
   NET_SALARY=BASIC_SALARY+DA+HRA-PF-TAX
3. Prepare a PPT on cloud computing –introduction , models, services ,and architecture ***Ppt should contain explanations, images and at least 20 pages*** (use **www.zoho.com** and ***docs.google.com)***
4. Create your resume in a neat format using google and zoho cloud

Programs on PaaS

1. Write a Google app engine program to generate n even numbers and deploy it to google cloud
2. Google app engine program multiply two matrices.
3. Google app engine program to validate user; create a database login(username, password) in Mysql and deploy to cloud
4. Write a Google app engine program to display nth largest no from the given list of numbers and deploy it into google cloud

5. Google app engine program to validate the user Use Mysql to store user info and deploy on to the cloud

Note: Implement Program 1-5 using Microsoft Azure

Case Study- Cloud Computing

S. No.	**Title of Experiment**	**Aim of the Experiment**	**Demonstration Equipment/ Components to be required**	**Type of Experiment/ Demonstration (Lab/Classroom)**
1	Case Study of Amazon	To understand the services of Amazon elastic cloud.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
2	Case Study of Azure	To understand the services of Microsoft azure.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
3	Case Study of Hadoop	To understand the services of hadoop.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
4	Case Study of Aneka	To understand the services of aneka elastic cloud.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
5	Case Study of Google Apps	To understand the services of google apps engine.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
6	Google apps business solution for data access and data upload	To understand the business solution application of Google apps.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
7	Control panel software manager Application of hypervisors	To understand the application of hypervisors.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.

REFERENCES

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

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

MCA III Year – I Semester

L	T	P	C
-	-	4	2

(19MC9150) .NET TECHNOLOGIES LAB

COURSE OBJECTIVES

1. *Students will gain the ability to implement the algorithms in C#.net, VB.net and ASP.net.*

COURSE OUTCOMES

1. *Create Simple application using web controls*
2. *Work with States of ASP.NET Pages*
3. *Query textbox and Displaying records & Display records by using database Datalist link control & Databinding using dropdownlist control Inserting record into a database & Deleting record into a database*
4. *Databinding using datalist control & Datalist control templates Databinding using datagrid & Datagrid control template.*
5. *Datagrid hyperlink & Datagrid button column Datalist event & Datagrid paging Creating own table format using datagrid*

List of Programs/Experiments

1. Simple application using web controls
    a) Finding factorial Value
    b) Money Conversion
    c) Quadratic Equation
    d) Temperature Conversion
    e) Login control
3. States of ASP.NET Pages
4. Query textbox and Displaying records
5. Display records by using database
6. Databinding using dropdownlist control
7. Inserting record into a database, Deleting record into a database.
8. Databinding using datagrid
9. Datagrid button column
10. Creating own table format using datagrid
11. Datagrid paging

REFRENCES

1. *Visual studio 2010 - A beginners guide,* Joseph Mayo
2. The Complete Reference Visual Basic .NET, Jeffrey R. Shapiro, Tata Mcgraw Hill (2002 Edition).
3. *Pro ASP.NET 4 in C# 2010*, MacDonald an d Freeman
4. *Visual Studio 2010 and .NET 4 Six in – One,* Wrox Programmer to Programmer
5. *Learning c#,* Jessy Liberty.

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

MCA III Year – I Semester

L	T	P	C
-	-	4	2

(19MC9151) MOBILE APPLICATION DEVELOPMENT LAB

COURSE OBJECTIVES

The student should be made to:

1. *Know the components and structure of mobile application development frameworks for Android based mobiles.*
2. *Understand how to work with various mobile application development frameworks.*
3. *Learn the basic and important design concepts and issues of development of mobile applications.*
4. *Understand the capabilities and limitations of mobile devices.*

COURSE OUTCOMES

At the end of the course, the student should be able to:

1. *Design and implement various mobile applications using emulators.*
2. *Design applications on animations.*
3. *Deploy applications using layout and form controls.*
4. *Deploy applications with multiple activities*
5. *Deploy applications to hand-held devices*

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.