SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)
DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS (MCA)
COURSE STRUCTURE

I Year-I- Semester

S.No.	Course Code	Subject	L	T	P	C
1	16HS601	Functional English	3	-	-	3
2	16HS613	Probability &Statistics	3	1	-	3
3	16HS618	Mathematical Foundations of Computer Science	3	1	-	3
4	16MC801	Computer Programming and Problem Solving	3	1	-	3
5	16MC802	Computer Organization	3	1	-	3
6	16HS607	English Language and Communication Skills Lab	-	-	4	2
7	16MC803	P. C. Software Lab	-	-	4	2
8	16MC804	C Programming Lab	-	-	4	2
		Contact periods/Week	15	4	12	21
			Total/Week 31			

I Year-II- Semester

S.No.	Course Code	Subject	L	T	P	C
1	16HS610	Professional English	3	-	-	3
2	16MB749	Accounting & Financial Management	3	1	-	3
3	16MC805	OOPs through Java	3	1	-	3
4	16MC806	Data Structures	3	1	-	3
5	16MC807	Database Management Systems	3	1	-	3
6	16MC808	Java Programming Lab	-	-	4	2
7	16MC809	Data Structures through C Lab	-	-	4	2
8	16MC810	Database Management Systems Lab	-	-	4	2
Audit Course						
9	16HS616	Aptitude Practice – I	3	-	-	-
		Contact periods/Week	18	4	12	21
			Total/Week 34			

II Year-I- Semester

S.No.	Course Code	Subject	L	T	P	C
1	16MC811	Computer Networks	3	1	-	3
2	16MC812	Object Oriented Analysis and Design (Using UML)	3	1	-	3
3	16MC813	Operating Systems	3	1	-	3
4	16MC814	Linux Programming	3	1	-	3
5	16MC815	Data Warehousing and Mining	3	1	-	3
6	16MC816	UML Lab	-	-	4	2
7	16MC817	Operating Systems and Linux Programming Lab	-	-	4	2
8	16MC818	Data Warehousing and Mining Lab	-	-	4	2
		Credit Course				
9	COE-I	Comprehensive Online Examination-I	-	-	-	1
		Audit Course				
10	16HS614	Comprehensive Soft Skills	3	-	-	-
		Contact periods/Week	18	5	12	22
			Total/Week 35			

II Year-II- Semester

S.No.	Course Code	Subject	L	T	P	C
1	16MC819	Software Engineering	3	1	-	3
2	16MC820	Web Technologies	3	1	-	3
3	16MC821	Computer Graphics	3	1	-	3
4	16MC822 16MC823 16MC824 16MC825	**Department Elective – I** Big Data Analytics Neural Networks& Fuzzy logic Distributed Systems Service Oriented Architecture	3	1	-	3
5	16MC826 16MC827 16MC828 16MC829	**Department Elective – II** Information Security Social Networks and Semantic Web Geological Information Systems .NET Technologies	3	1	-	3
6	16MC830	Software Engineering Lab	-	-	4	2
7	16MC831	Web Technologies Lab	-	-	4	2
8	16MC832	Computer Graphics Lab			4	2
		Credit Course				
9	COE-II	Comprehensive Online Examination-II	-	-	-	1
		Audit Course				
10	16HS617	Aptitude Practice - II	3	-	-	-
		Contact periods/Week	18	5	12	22
			Total/Week 35			

III Year-I- Semester

S.No.	Course Code	Subject	L	T	P	C
1	16MC833	Software Testing	3	1	-	3
2	16MC834	Mobile Application Development	3	1	-	3
3	16MC835	Software Project Management	3	1	-	3
4	 16MC836 16MC837 16MC838 16MC839	**Department Elective – III** Cyber Security Image Processing Artificial Intelligence Information Retrieval Systems	3	1	-	3
5	 16MC840 16MC841 16MC842 16MC843	**Department Elective – IV** M-Commerce Cloud Computing Design Patterns Cognitive Computing	3	1	-	3
6	16MC844	Software Testing Lab	-	-	4	2
7	16MC845	Mobile Application Development Lab	-	-	4	2
8	16MC846	Software Project Management Lab	-	-	4	2
		Credit Course				
9	COE-III	Comprehensive Online Examination-III	-	-	-	1
		Audit Course				
10	16HS615	Advanced English Language and Communication Skills Lab	-	-	4	-
		Contact periods/Week	15	5	16	22
			Total/Week 36			

III Year-II- Semester

S.No	Course Code	Subject	L	T	P	C
1	16MC847	Seminar	-	-	4	2
2	16MC848	Dissertation/ Thesis	-	-	20	10
		Contact periods/Week	-	-	24	12
			Total/Week 24			

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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

I MCA I- Semester

L	T	P	C
3	0	0	3

(16HS601) FUNCTIONAL ENGLISH

Course Objectives:

- To develop communication skills among the students.
- To construct proficiency in academic and social purpose to improve their grammatical accuracy.
- To understand LSRW skills and inculcate the habit of reading for pleasure.
- To obtain study skills and communication skills in formal and informal situations.
- To use appropriate vocabulary

Course Outcomes

Students will be able to

- Use LSRW skills through the prescribed text and develop their ability to communicate effectively.
- Articulate well among themselves and with Faculty.
- Construct compound sentences using common conjunctions.
- Manage to organize and deliver oral presentations.
- Demonstrate the skills needed to participate in a conversation that builds knowledge collaboratively.

UNIT-I

*Mindscapes (*Environmental Consciousness: Pollution - How to Regain Green Cover)

1. Learning English Language through Literature (*Secret of work*- Swami Vivekananda)
2. Present, Past and Future aspects
3. Introducing oneself

Grammar: Parts of speech-Kinds of sentences-Tenses

Vocabulary: Synonyms & Antonyms-Affixes – Phrasal verbs

Listening & Reading Activities

Writing: Paragraph writing-Note taking & Note making

Phonetics- Syllabification

UNIT-II

*Mindscapes (*Emerging Technologies: Solar Thermal Power- Nano Technology)

1. Learning English Language through Literature (*Stopping by Woods on a Snowy Evening*- Robert Frost)
2. Set in the Past
3. Inter-personal skills

Grammar – Articles – Past Events – Voice & Impersonal passive voice – Gerund & -*ing*forms to-infinitives

Vocabulary: Phrases – Idioms – word roots

Listening & Reading Activities

Writing: Letter writing-Informal-Formal

Phonetics – Accent

UNIT-III

Mindscapes (Global Issues: Child Labor- E- Waste)

1. Learning English Language through Literature (*What is my Name?*-P.Satyavathi)
2. Describing a person, place and object
3. Possibilities

Grammar: Modals – Conditionals – Framing Questions – Compound nouns Verbs

Vocabulary: One word substitute– Fixed expressions– Clauses

Listening & Reading Activities

Writing: Information transfer

Intonation: Falling & Rising

UNIT-IV

*Mindscapes (*Space Trek: Hubble Telescope- Genesis of ISRO)

1. Learning English Language through Literature (*Man in Black*-Oliver Goldsmith)
2. Analytical thinking
3. Co-operative learning

Grammar: Concord–Reported speech-compare & contrast

Vocabulary: Numerical expressions-definitions-collocations

Listening & Reading

Writing: Summary-Essay writing-Making instructions

JAM

UNIT-V

*Mindscapes (*Media Matters: History Of Media- Power of Media- Interviews)

1. Learning English Language through Literature (*The Power of Prayer*-Abdul Kalam)
2. Exploring creative ideas
3. Synthesis of sentences

Grammar: Simple, compound and complex-Spotting errors

Vocabulary: Discourse markers-Homonyms-Homophones-Homographs

Listening & Reading Activities

Writing: Writing recommendations-scrambled sentences

Convincing others

TEXT BOOKS:

1. *Mindscapes: English for Technologists and Engineers*- Orient Black Swan,2014.
2. *Paths to Progress in English:* Orient Black Swan

REFERNCES:

1. Raymond Murphy's *Intermediate English Grammar* with CD, Raymond Murphy, Cambridge University Press, 2012
2. *Communication Skills*, Sanjay Kumar & Pushpalatha, Oxford University Press, 2012.

3. *Writing Tutor. Advanced English Learners' Dictionary*, 9th Edition, Oxford University Press, 2015.
4. *Powerful Vocabulary Builder*, Anjana Agarwal, New Age International Publishers, 2011.
5. *Keep Talking*, F. Klippel, Cambridge University Press, 2013.
6. *Listening Extra*, Miles Craven, Cambridge University Press, 2008.
7. *Reading Extra*, Liz Driscoll, Cambridge University Press, 2004.
8. *Writing Extra*, Graham Palmer, Cambridge University Press, 2004.
9. *Speak Well*, Jayashree Mohanraj et al, Orient Blackswan, 2013.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

I MCA I- Semester

L	T	P	C
3	1	0	3

(16HS613) PROBABILITY &STATISTICS

Course Objectives:

- To train the students thoroughly in Mathematical concepts fundamentals of probability, test of hypothesis, Test of significance and ANOVA
- To prepare students for lifelong learning and successful careers using mathematical concepts of probability, test of hypothesis, Test of significance and ANOVA
- 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:

At the end of the course, students would be expected to:

- Have acquired ability to participate effectively in group discussions
- Have developed ability in writing in various contexts
- Have acquired a proper level of competence for employability

UNIT-I

Probability: Introduction, Conditional probability, Baye’s theorem, Random variables, Discrete and continuous Distributions, Expectation, Variance, Moments, Moment generating functions

UNIT-II

Distributions: Binomial, Poisson, Normal, Uniform, Exponential and Gamma distributions, related properties and applications

UNIT-III

Test of Hypothesis: Population and Sample, Confidence interval of mean from Normal distribution, Null and Alternative hypothesis, Level of significance
Test of significance: Test based on normal distribution, Z test for means and proportions, Small samples t- test for one sample and two sample problem and paired t-test, F-test and Chi-square test (testing of goodness of fit and independence).

UNIT-IV

Analysis of variance (ANOVA) one way classification and two way classification (Latin square Design and RBD)

UNIT-V

Statistical Quality Control: Concept of quality of a manufactured product, Defects and Defectives, Causes of variations, Random and assignable, The principle of Shewhart Control

Chart, Charts for attribute and variable quality characteristics, Constructions and operation of X - Chart, R-Chart, P-Chart and C-Chart.

Text Books:

1. *Statistical methods* by S.P. Gupta, S.Chand publications.
2. *Probability & Statistics* by T.K.V. Iyengar, S.Chand publications.

Reference Books:

1. *Probability & Statistics* by E. Rukmangadachari& E. Keshava Reddy, Pearson Publisher.
2. *Probability & Statistics for engineers* by Dr. J. Ravichandran WILEY-INDIA publishers.
3. *Probability & Statistics for Science and Engineering* by G.ShankerRao, Universities Press.
4. *Probability and Statistics for Engineering and Sciences* by Jay L.Devore, CENGAGE.
5. *Probability and Statistics* by R.A. Jhonson and Gupta C.B.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

I MCA I- Semester

L	T	P	C
3	1	0	3

(16HS618) MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

Course Objectives:

- To train the students thoroughly in Mathematical concepts of Mathematical logic, Relations, Algebraic structures, Recurrence Relation, Graph Theory
- To prepare students for lifelong learning and successful careers using mathematical concepts of Mathematical logic, Relations, Algebraic structures, Recurrence Relation, Graph Theory
- 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:

At the end of the course, students would be expected to:

- Have acquired ability to participate effectively in group discussions
- Have developed ability in writing in various contexts
- Have acquired a proper level of competence for employability

UNIT-I

Mathematical Logic: Statements and notations, Connectives, Well-formed formulas, Truth Tables, tautology, equivalence implication, Normal forms, Quantifiers, universal quantifiers

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

UNIT-II

Relations: Properties of Binary Relations, equivalence, transitive closure, compatibility and partial ordering relations, Functions: Inverse Function, Composition of functions, recursive Functions

Algebraic structures: Algebraic systems examples and general properties, Semi groups and monads, groups, sub groups homomorphism, Isomorphism.

UNIT-III

Elementary Combinatorics:Basis of counting, Enumerating Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients, Binomial Multinomial theorems, the principles of Inclusion – Exclusion. Pigeon hole principles and its application

UNIT-IV

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

UNIT-V

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

TEXT BOOKS:

1. *Discrete Mathematics with Applications*, Thomas Koshy, Elsevier.
2. *Discrete Mathematics and its applications*, 6th edition, K.H.Rosen, TMH.

REFERENCES:

1. *Elements of Discrete Mathematics*- A Computer Oriented Approach, C.L.Liu,D.P. Mohapatra, 3/e,TMH.
2. *Discrete Mathematics for Computer Scientists & Mathematicians*, 2/e, J.L.Mott, Kandel, T.P. Baker, PHI
3. *Discrete Mathematical Structures with Application to Computer Science*, Tremblay,
4. Manohar McGraw Hill Publication
5. *Discrete and Combinatorial Mathematics*- An Applied Introduction, Ralph.
6. P.Grimaldi, 5/e, Pearson Education
7. *Discrete Mathematical Structures*, Mallik and Sen, Cengage Learning.
8. *Discrete Mathematical Structures*, BernandKolman, Robert C. Busby, Sharon CutlerRoss, PHI/ Pearson Education

7. *Discrete Mathematics*, Lovasz, Springer.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

I MCA I- Semester

L	T	P	C
3	1	0	3

(16MC801) COMPUTER PROGRAMMING AND PROBLEM SOLVING

Course Objectives:

- To make the student learn a programming language.
- To make student understand the syntax and semantics of C programing language and other features of the language.
- 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.
- File Handling for permanent storage of data or record.
- Applications of Self- referential structure and Unions.

Course Outcomes:

Upon completion of the subject, students will be able to

- Student can effectively apply problem solving techniques in designing the solutions for a wide range of problems.
- Write, compile and debug programs in C language.
- Design programs involving decision structures, loops and functions.
- Explain the difference between call by value and call by reference
- Understand the dynamics of memory by the use of pointers.

UNIT-I

Introduction to Computers: Introduction, Characteristics of computers, Evolutions of computer, Computer generations, Classifications of computers, Applications of computers, Basic Computer organization, Number Systems (Binary, Octal, Decimal, Hex), Representation of numbers (fixed and floating point)

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

The C Declarations–Introduction, 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.

Recursion- 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, type def, 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. *Fundamentals of Computers: For Undergraduate Courses in Commerce and Management'*by ITL Education Solutions Limited ,Pearson Education India
3. Introduction to Computer ScienceI. T. L. Education Solutions Limited, ItlEsl Pearson Education India
4. *"Computer Fundamentals and C Programming"*, First Edition, Dr.P.Chenna Reddy, Available at: www.Pothi.com.

REFERENCES:

1. *'The C Programming Language'*, Brian W. Kernighan, 2015
2. Kernighan BW and Ritchie DM, *the C Programming Language*, 2nd Edition, Prentice Hall of India, 2006

3. J.R. Hanly and E.B. Koffman, *Problem Solving and Program Design in C*, Pearson Education, 2007.
4. B. A. Forouzan and R.F. Gilberg, *C Programming & Data Structures*, Cengage Learning, 2007.
5. Ashok N. Kamthaneet. al., *Computer Programming and IT*, Pearson Education, 2011
6. Mahapatra, *Thinking In C* , PHI Publications, 1998.
7. *Let Us C* - 14th Revised and Updated Edition, 13 Jul 2016

I MCA I- Semester

L	T	P	C
3	1	0	3

(16MC802) COMPUTER ORGANIZATION

Course Objectives:

- To impart the knowledge in the field of digital electronics.
- To impart knowledge about the various components of a computer and its internals.
- To design and realize the functionality of the computer hardware with basic gates and other components using combinational and sequential logic.
- To understand the importance of the hardware-software interface.

Course Outcomes:

- Able to design digital circuits by simplifying the Boolean functions
- Able to understand the organization and working principle of computer hardware components
- Able to understand mapping between virtual and physical memory
- Acquire knowledge about multiprocessor organization and parallel processing
- 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 (AUTONOMOUS)

I MCA I- Semester

L	T	P	C
0	0	4	2

(16HS607) ENGLISH LANGUAGE AND COMMUNICATION SKILLS LAB

The Language Lab focuses on the production and practice of sounds of language and familiarizes the students with the use of English in everyday situations and contexts.

Course objectives:

- To enable students to learn better pronunciation through stress on word accent, Intonation and rhythm.
- To help the second language learners to acquire fluency in spoken English and neutralize mother tongue influence.
- To train students to use language appropriately in both formal and informal situations.
- To enhance written communication among the students.

Course outcomes:

- To become active participants in the learning process and acquire proficiency in spoken English.
- To speak with clarity and confidence thereby enhances employability skills.
- To prepare effective job application

UNIT - I

1. Phonetics -Importance
2. Introduction to Sounds of Speech
3. Vowels and Consonant sounds
4. Phonetic Transcription

UNIT - II

5. Word Stress
6. Syllabification
7. Rules of Word Stress
8. Intonation

UNIT - III

9. Situational Dialogues/ Role Play
10. Telephonic Communication
11. JAM

UNIT – IV

12. Describing Persons/ places/ things
13. Oral Presentations
14. Debate

UNIT – V

15. Group Discussion
16. Job application
17. Interview skills

Minimum Requirements for ELCS Lab:

The English Language Lab shall have two parts:

1. Computer Assisted Language Learning (CALL) Lab: The Computer aided Language Lab for 60 students with 60 systems, one master console, LAN facility and English language software for self- study by learners.
2. The Communication Skills Lab with movable chairs and audio-visual aids with a P.A. system, Projector, a digital stereo-audio & video system and camcorder etc.

System Requirement (Hardware component):

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

- i) P – IV Processor
  - a) Speed – 2.8 GHZ
  - b) RAM – 512 MB Minimum
  - c) Hard Disk – 80 GB
- ii) Headphones of High quality

Suggested Software:

1. Clarity Pronunciation Power – Part I (Sky Pronunciation)
2. Clarity Pronunciation Power – part II
3. K-Van Advanced Communication Skills
4. Walden InfoTech Software.

REFERENCES:

1. *A Textbook of English Phonetics for Indian Students* 2nd Ed T. Balasubramanian. (Macmillian), 2012.
2. *A Course in Phonetics and Spoken English*, DhamijaSethi, Prentice-Hall of India Pvt.Ltd
3. *Speaking English Effectively*, 2nd Edition Krishna Mohan & NP Singh, 2011. (McMillan).
4. *A Hand book for English Laboratories*, E.Suresh Kumar, P.Sreehari, Foundation Books, 2011.
5. *Spring Board Success*, Sharada Kouhik, Bindu Bajwa, Orient Blackswan, Hyderabad, 2010.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

I MCA I- Semester

L	T	P	C
0	0	4	2

(16MC803) P.C. SOFTWARE LAB

Course Objectives:

- To impart the knowledge Hardware and software components of PC.
- To impart knowledge about the various components of a computer and its internals.
- To disassemble and assemble the PC back to working condition.
- To understand the installation of softwares.
- To impart the knowledge on Software troubleshooting and Hardware troubleshooting.
- To impart knowledge on MS Office tools.

Course Outcomes:

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

PC Hardware

Exercise 1 – Task 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 – Task 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 – Task 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 – Task 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 – Task 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 – Task 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 - Task 1: **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 - Task 3: **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 DavidAnfinson and Ken Quamme. – CISCO Press, Pearson Education.
7. *IT Essentials PC Hardware and Software Labs and Study Guide, T*hird Edition by Patrick Regan – CISCO Press, Pearson Education.
8. *Troubleshooting, Maintaining and Repairing PCs*,S.J.Bigelow,5thedition,TMH

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

I MCA I- Semester

L	T	P	C
0	0	4	2

(16MC804) C PROGRAMMING LAB

Course Objectives:

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

Course Outcomes:

Upon completion of the subject, students will be able to

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

Recommended Systems/Software Requirements:

- Intel based desktop PC with ANSI C Compiler and Supporting Editors

List of Sample Examples/ Experiments

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
(AUTONOMOUS)

I MCA II- Semester

L	T	P	C
3	0	0	3

(16HS610) PROFESSIONAL ENGLISH

Course Description: The course content focuses on LSRW skills and vocabulary building to enrich their command over language. Relevant task based activities are also carried out to enhance their communication skills.

Course Objectives:

- To develop communication skills among the students
- To construct proficiency in academic and social purpose.
- To improve their grammatical accuracy.
- To understand LSRW skills and inculcate the habit of reading for pleasure.

Course Outcomes:

Students will be able to

- Use LSRW skills through the prescribed text and develop their ability to communicate effectively
- Articulate well among themselves and with Faculty.
- Construct compound sentences using common conjunctions.
- Manage to organize and deliver oral presentations.
- Demonstrate the skills needed to participate in a conversation that builds knowledge Collaboratively

UNIT-I

*Mindscapes (*Lessons from the past: Importance of the Past)

1. Learning English Language through Literature (*Playing the English Gentleman* M.K.Gandhi)
2. Oral presentation
3. Effective writing

Grammar: Relative clauses-Adjectives
Vocabulary: Sequencing words
Listening & Reading Activities
Writing: Analytical paragraph writing-Minutes of meeting

UNIT-II

*Mindscapes (*Energy: Renewable and Non-renewable Sources - Alternative Sources)

1. Learning English Language through Literature. (*The Portrait of a Lady* -Kushwant Singh)
2. Preparing and presenting slides, Telephone etiquette
3. Making drafts

Grammar: Adverbs - prepositions -cause and effect expressions
Vocabulary: phrasal verbs - Technical vocabulary-Extended definitions

Listening & Reading Activities

Writing: Report writing

UNIT-III

*Mindscapes (*Engineering Ethics: Biotechnology - Protection from Natural Calamities)

1. Learning English Language through Literature (*La Belle Dame Sans Mercy*-John Keats)
2. Poster presentation, Debate
3. Technical drafting

Grammar: Using connectives-Gap filling exercise using appropriate tense form

Vocabulary: Acronyms & Abbreviations

Listening & Reading Activities

Writing: Writing projects

UNIT-IV

Mindscapes (Travel and Tourism: AtithiDevoBhava- Tourism in India)

1. Learning English Language through Literature (*A Marriage Proposal*-Anton Chekov)
2. Group Discussion
3. Reading comprehension

Grammar: Structure indicating purpose-Subject-verb agreement

Vocabulary: emoticons-cloze test

Listening & Reading

Writing: Intensive and extensive

UNIT-V

Mindscapes (Getting Job Ready: SWOT Analysis- Preparing for Interviews)

1. Learning from Literature (*Bird Sanctuary* -Sarojini Naidu)
2. Interview etiquette
3. Job application

Grammar: Spotting errors, Gap filling exercises using “gerunds” & present participle forms

Vocabulary: verbal ability

Listening & Reading Activities

Writing: Covering letter, Resume, Curriculum vitae

Convincing others

TEXT BOOKS:

1. *Mindscapes*: *English for Technologists and Engineers*, Orient Blackswan, 2014
2. *Paths to Progress in English:*Orient Black Swan

REFERENCES:

1. *Effective Tech Communication***,** Rizvi,Tata McGraw-Hill Education, 2007.
2. *Technical Communication,*Meenakshi Raman, Oxford University Press.
3. *English Conversations Practice,*Grant Taylor, Tata McGrawHill publications,2013**.**
4. *Practical*
5. *English Grammar,*Thomson and Martinet, OUP, 2010.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

I MCA II- Semester

L	T	P	C
3	1	0	3

(16MB749) ACCOUNTING & FINANCIAL MANAGEMENT

Course Objectives:

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

Course Outcomes:

- This course is designed to introduce students to the principles, concepts, and applications of financial accounting and management.

.

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
(AUTONOMOUS)

I MCA II- Semester

L	T	P	C
3	1	0	3

(16MC805) OOPS THROUGH JAVA

Course Objectives:

- Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
- Understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
- Have the ability to write a computer program to solve specified problems.
- Be able to use the Java SDK environment to create, debug and run simple Java programs.
- Understand the fundamentals of Java collection frame work, Exceptions and multi-threading concepts.
- Understand the GUI concepts like Applets, Swings.
- Understand the event handling mechanisms.

Course Outcomes:

Students who have completed this course able to:

- Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
- Understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
- Have the ability to write a computer program to solve specified problems.
- Able to do the java collection framework programs.
- 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, Review of 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.

Inner classes: Uses of inner classes, local inner classes, anonymous inner classes, static inner classes, examples.

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, TreeMap, Collection Interfaces – Collection, Set, List, 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,* 7th edition, Herbert Schildt, TMH.
2. *Core Java, Volume 1-Fundamentals*, eighth edition, Cay S.Horstmann and Gary Cornell, Pearson education

REFERENCES:

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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

I MCA II- Semester

L	T	P	C
3	1	0	3

(16MC806) DATASTRUCTURES

Course Objectives:

- To understand the basic structure concept such as Abstract Data Types, Linear and Non Linear Data structures.
- To understand the notations used to analyze the Performance of algorithms.
- To understand the behavior of data structures such as stacks, queues, trees, directories, trees, Graph and their representations.
- To choose the appropriate data structure for a specified application.
- To understand and analyze various searching and sorting algorithms.
- 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:

- Learn how to use data structure concepts for realistic problems.
- Ability to identify appropriate data structure for solving computing problems in C language.
- Ability to solve problems independently and think critically.

UNIT-I

Sorting: Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Radix Sort and Quick Sort

Searching: Linear Search, Binary Search, and Fibonacci Search.

Dictionaries: Sets, Dictionaries, Hash Tables, Open Hashing, Closed Hashing (Rehashing Methods), Hashing Functions (Division Method, Multiplication Method, Universal Hashing), Analysis of Closed Hashing Result (Unsuccessful Search, Insertion, Successful Search, Deletion), Hash Table Restructuring, Skip Lists, Analysis of Skip Lists.

UNIT-II

Linked list: Definition, Operations on Single linked lists, Doubly linked lists, Circular linked lists and Circular Double linked lists

Applications of Linked list: Sparse Matrix Manipulation, Polynomial Representation

UNIT-III

Stacks: Introduction, Definition, Representation of Stacks- Arrays and Linked lists, Operations on stacks, Applications of stacks-Evaluation of Arithmetic Expression, Implementation of Recursion, Factorial Calculations, Towers of Hanoi.

Queues: Introduction, Definition, Representation of Queues- Arrays and Linked lists, Various Queue structures, Operations on Queues, Applications, Priority queues.

UNIT-IV

Search Trees: Binary Search Trees, Definition, ADT, Implementation, Operations-Searching, Insertion and Deletion.

AVL Trees, Definition, Height of an AVL Tree, Operations – Insertion, Deletion and Searching.

Introduction to Red –Black and Splay Trees, B-Trees, B-Tree of order m, height of a B-Tree, insertion, deletion and searching, Comparison of Search Trees

UNIT-V

Graphs: Operations on Graphs: Vertex insertion, vertex deletion, find vertex, edge addition, edge deletion, Graph Traversals- Depth First Search and Breadth First Search (Non recursive) Graph storage Representation- Adjacency matrix, adjacency lists.

Graph algorithms: Minimum-Cost Spanning Trees-Prim's Algorithm, Kruskal's Algorithm Shortest Path Algorithms: Dijkstra's Algorithm, All Pairs Shortest Paths Problem: Floyd's Algorithm, Warshall's Algorithm,

TEXT BOOKS:

1. *'Programming in C and Data Structures'*, J.R. Hanly, Ashok N. Kamthane, A. AnandaRao, Pearson Education.
2. *'C Programming & Data Structures'*, B.A.Forouzan and R.F. Gilberg, 3/e, Cengage Learning.
3. *'An Introduction to Data Structures With Applications'*, Trembley, Sorenson, 2/e, TMH.
4. *'Classic Data Structures'*, D.Samanta, 2nd edition, PHI.

REFERENCES:

1. Programming in C – Stephen G. Kochan, III Edition, PearsonEductaion.
2. C Programming with problem solving, J.A. Jones & K. Harrow, Dreamtech Press
3. Data Structures using C – A.M.Tanenbaum, Y.Langsam, and M.J. Augenstein, 8/e, Pearson Education / PHI,.
4. Fundamentals of Data structures in C, 2nd Edition, E.Horowitz, S.Sahni and Susan Anderson-Freed, Universities Press
5. Data structures A Programming Approach with C, D.S.Kushwaha and A.K.Misra, PHI.
6. Data structures: A Pseudocode Approach with C, 2nd edition, R.F.GilbergAndB.A.Forouzan, Cengage Learning.
7. Data Structures using C, A.M.Tanenbaum,Y. Langsam, M.J.Augenstein, Pearson.

I MCA II- Semester

L	T	P	C
3	1	0	3

(16MC807) DATABASE MANAGEMENT SYSTEMS

Course Objectives:

- To understand the fundamentals of data models and conceptualize and depict a database system using ER diagram
- To make a study of SQL and relational database design.
- To know about data storage techniques a query processing.
- To impart knowledge in transaction processing, concurrency control techniques and recovery procedures.

Course Outcomes:

- Understand the basic concepts of the database and data models.
- Design a database using ER diagrams and map ER into Relations and normalize therelations.
- Acquire the knowledge of query evaluation to monitor the performance of the DBMS.
- Develop a simple database applications using normalization.
- 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. E/R 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, 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 (AUTONOMOUS)

I MCA II- Semester

L	T	P	C
0	0	4	2

(16MC808) JAVA PROGRAMMING LAB

Course Objectives:

- To understand Java and object-oriented concepts.
- To write, debug, and document well-structured Java applications
- To understand decision and iteration control structures to implement algorithms
- To write simple recursive algorithms
- To implement interfaces, inheritance, and polymorphism as programming techniques
- To apply exceptions handling
- To implement Java collection framework as programming techniques.

Course Outcomes:

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

- Understand programming language concepts, particularly Java and object-oriented concepts.
- Write, debug, and document well-structured Java applications
- Implement Java classes from specifications
- Effectively create and use objects from predefined class libraries
- Understand the behavior of primitive data types, object references, and arrays
- Apply decision and iteration control structures to implement algorithms
- Write simple recursive algorithms
- Implement interfaces, inheritance, and polymorphism as programming techniques.
- Implement Java collection frame work as programming techniques.

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

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+ib, where a and b are real numbers. The operations supported by this ADT are:
    a) Reading a complex number.
    b) Writing a complex number.
    c) Addition of Complex numbers.
    d) Subtraction of complex numbers.
    e) Multiplication 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 number of Sides ( ). 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 number of Sides ( ) 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
(AUTONOMOUS)

I MCA II- Semester

L	T	P	C
0	0	4	2

(16MC809) DATA STRUCTURES THROUGH C LAB

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 Queueand Priority Queue using arrays and linked list concepts.
8. Write a menu driven program in C to
   i) Create a binary search tree.
   ii) Traverse the tree in Inorder, Preorder and Post Order.
   iii) Search the tree for a given node and delete the node.
9. Write a program in C to implement insertion and deletion in B- tree.
10. Write a program in C to implement insertion and deletion in AVL tree.
11. Write a menu driven program that implements Heap tree (Maximum and Minimum Heap tree) for the following operations. (Using array) Insert, Delete.
12. Write a program to implement double hashing technique to map given key to the address space. Also write code for collision resolution (linear probing).
13. Write a program in C to implement Dijkstras shortest path algorithm for a directed graph.
14. Write a program in C to insert and delete nodes in graph using adjacency matrix.
15. Write a program in C to implement BFS using linked representation of graph.
16. Write a program in C to implement Depth first search using linked representation of graph.
17. Write a program in C to create a minimum spanning tree using Kreskas's algorithm.
18. Write a program in C to create a minimum spanning tree using Prim'salgorithm.

REFERENCES:

1. Programming in C – Stephen G. Kochan, III Edition, PearsonEductaion.
2. C Programming with problem solving, J.A. Jones & K. Harrow, Dreamtech Press
3. Data Structures using C – A.M.Tanenbaum, Y.Langsam, and M.J. Augenstein, 8/e, Pearson Education / PHI,.
4. Fundamentals of Data structures in C, 2nd Edition, E.Horowitz, S.Sahni and Susan Anderson-Freed, Universities Press
5. Data structures A Programming Approach with C, D.S.Kushwaha and A.K.Misra, PHI.
6. Data structures: A Pseudocode Approach with C, 2nd edition, R.F.GilbergAndB.A.Forouzan, Cengage Learning.
7. Data Structures using C, A.M.Tanenbaum,Y. Langsam, M.J.Augenstein, Pearson.
8. Data Structures and Algorithms made easy in JAVA, 2nd Edition, NarsimhaKarumanchi, CareerMonk Publications.
9. Data Structures, S.Lipscutz,Schaum’s Outlines, TMH.
10. Data structures using C, A.K.Sharma, 2nd edition, Pearson..

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

I MCA II- Semester

L	T	P	C
0	0	4	2

(16MC810) DATABASE MANAGEMENT SYSTEMS LAB

Course Objectives:

- The objective of the course is to enable students to understand and use a relational database system.
- Introduction to Databases, Conceptual design using ERD, Functional dependencies and Normalization, Relational Algebra are covered in detail.
- Students learn how to design and create a good database and use various SQL operations.
- The course concludes with an overview of transaction management and introduction to advanced and non-relational databases.

Course Outcomes:

- Able to master the basic concepts and understand the applications of database systems.
- Able to construct an Entity-Relationship (E-R) model from specifications and to transform to relational model.
- Able to construct unary/binary/set/aggregate queries in Relational Algebra.
- Understand and apply database normalization principles.
- Able to construct SQL queries to perform CRUD operations on database. (Create, Retrieve, Update, Delete)
- Understand principles of database transaction management, database recovery, security.
- Be aware of non-relational databases and applications.

<u>**List of Sample Programs/Experiments**</u>

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.

I MCA II- Semester

L	T	P	C
3	0	0	0

(16HS616) APTITUDE PRACTICE-I

Course Objectives:

After thorough learning of Quantitative Aptitude and Reasoning, a student:

- Will be able to critically evaluate various real life situations by resorting to Analysis of key issues and factors.
- Will be able to read between the lines and understand various language structures.
- Will be able to demonstrate various principles involved in solving mathematical problems and thereby reducing the time taken for performing job functions.

Course Outcomes:

At the end of the course, students would be expected to:

- Have developed the subtle way of approaching in the candidate.
- Have acquired the decision making with in no time.
- Have acquired logical thinking during professional tenure.
- Have obtained quick decision making skills.

UNIT-I

Numbers: Classification of numbers, Divisibility rules, finding the units digit, decimal fractions, simplifications, LCM and HCF Models. Square roots and Cube roots.

Averages: Averages, Mixtures and Allegations

Ages: Problems on Ages

UNIT-II

Time and Distance: Relation between speed, distance and time, Converting kmph into m/s and vice versa, Problems on average speed, Relative speed, Trains, Boats and Streams, circular tracks and Races.

Time and Work: Problems on unitary method, Relation between Men, Days, Hours and Work. Problems on Man-Day-Hours method, Problems on alternate days, Problems on Pipes and Cisterns.

UNIT-III

Percentages: Converting percentage into decimals and vice versa. Equivalent percentage of fractions.

Partnership: Introduction, Relation between capitals, Period of investments and shares **Ratio and proportion:** Ratio and its properties, Comparison of ratios, Problems on ratios, Compound Ratio, Problems on proportion, Mean proportional and continued proportion.

UNIT-IV

Profit and Loss: Problems on Profit and Loss, Relation between Cost Price and Selling price, Discount and Marked Price, Two different articles sold at same Cost Price, Two different articles sold at same Selling Price, Gain% and Loss%.

Simple Interest: Definitions, Problems on interest and amount, Problems on rate of interest and time period.

Compound Interest: Definition and formula for amount in compound interest, Difference between simple interest and compound interest for 2 years on the same, Principle and time period.

UNIT-V

Clocks: Finding the angle when the time is given, Finding the time when the angle is known, Relation between Angle, Minutes and Hours, Exceptional cases in clocks

Calendars: Definition of a Leap Year, Finding the number of Odd days, framing the year code for centuries, Finding the day of any random calendar date

Blood relations: Defining the various relations among the members of a family, Solving Blood Relation puzzles, solving the problems on Blood Relations using symbols and notations.

TEXT BOOKS:

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

REFERENCES:

1. *Quantitative Aptitude*, R S Agarwal, S Chand,
2. *Quantitative Aptitude*, G. L BARRONS
3. *Quantitative Aptitude*, AbhijitGuhaMcGraw Hills
4. *Magical Book on Quicker Maths*, Tyra, BSC publishing company

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

II MCA I- Semester

L	T	P	C
3	1	0	3

(16MC811) COMPUTER NETWORKS

Course Objectives:

- To understand networking concepts and basic communication model
- To understand network architectures and components required for data communication.
- To analyze the function and design strategy of physical, data link, network layer and transport layer
- To Acquire knowledge of various application protocol standard developed for Internet

Course Outcomes:

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

UNIT-I

Network hardware, Reference models, Transmission media, Narrow band and Broadband ISDN. DATA LINK LAYER DESIGN ISSURES - 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, World Wide Web-HTTP, Network Management, SNMP. Network Security: Cryptography – Substitution Ciphers, Transposition Ciphers. Symmetric & Public Key algorithms – DES, 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
(AUTONOMOUS)

II MCA I- Semester

L	T	P	C
3	1	0	3

(16MC812) OBJECT ORIENTED ANALYSIS AND DESIGN (Using UML)

Course Objectives:

- To provide a brief, hands-on overview of object-oriented analysis in software process
- To discuss case studies based project specifications to develop object-oriented models and identify implementation strategies.
- To demonstrate and apply basic object oriented techniques to create and modify object oriented analysis and design models.

Course Outcomes:

- Understand the basic concepts to identify state & behavior of real world objects
- Able to learn the various object oriented methodologies and choose the appropriate one for solving the problem with the help of various case studies
- Understand the concept of analysis, design & testing to develop a document for the project
- Able to implement analysis, design & testing phases in developing a software project

UNIT-I

Introduction to UML: Importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture, and Software Development Life Cycle.

UNIT-II

Basic Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams.
Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages, E. R. Diagram.

UNIT-III

Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.
Basic Behavioral Modeling: Interactions, Interaction diagrams, Use case Diagrams, Activity Diagrams.

UNIT-IV

Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams. Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams.

UNIT-V

Patterns and Frameworks, Artifact Diagrams. Case Study: The Unified Real time application.

TEXT BOOKS:

1. Grady Booch, James Rumbaugh, Ivar Jacobson: *The Unified Modeling Language User Guide,* Pearson Education 2nd Edition.

REFERENCES:

1. Meilir Page-Jones: *Fundamentals of Object Oriented Design in UML,* Pearson Education.
2. Pascal Roques: *Modeling Software Systems Using UML2,* WILEY-DreamTech India Pvt. Ltd.
3. *Appling UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process*, Craig Larman, Pearson Education.
4. *Object-Oriented Analysis and Design with the Unified Process* By John W. Satzinger, Robert B Jackson and Stephen D Burd, Cengage Learning.
5. *UML and C++*,R.C.Lee, and W.M.Tepfenhart, PHI.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

II MCA I- Semester

L	T	P	C
3	1	0	3

(16MC813) OPERATING SYSTEMS

Course Objectives:

- To be aware of the evolution and fundamental principles of operating system, processes and their communication
- 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
- To be aware of components of operating system with relevant case study

Course Outcomes:

- Able to understand the operating system components and its services
- Implement the algorithms in process management and solving the issues of IPC
- Able to demonstrate the mapping between the physical memory and virtual memory
- Able to understand file handling concepts in OS perspective
- 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 and Virtual Memory - 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. William Stallings, “*Operating Systems – Internals and Design Principles”*, 7th Edition, Prentice Hall, 2011.
2. Andrew S. Tanenbaum, “*Modern Operating Systems*”, Second Edition, Addison Wesley, 2001.
3. D M Dhamdhere, “*Operating Systems: A Concept-Based Approach*”, Second Edition, Tata McGraw-Hill Education, 2007.
4. Charles Crowley, “*Operating Systems: A Design-Oriented Approach*”, Tata McGraw Hill Education”, 1996.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

II MCA I- Semester

L	T	P	C
3	1	0	3

(16MC814) LINUX PROGRAMMING

Course Objectives:

- To understand the LINUX system structure.
- To understand and use command line shell.
- To make effective use of Unix utilities and Shell scripting language such as bash.
- To produce programs similar to standard unix utilities such as ls, mv, cp etc. using Unix system calls.
- To develop the skills necessary for Unix systems programming including file system programming, process and signal management, and inter-process communication.
- To develop the basic skills required to write network programs using Sockets.

Course Outcomes:

- Work confidently in Linux environment.
- Work with shell script to automate different tasks as Linux administration

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: File Concept, File System Structure, Inodes, File types, The standard I/O (fopen, fclose, fflush, fseek, fgetc, getc, getchar, fputc, putc, putchar, fgets, gets etc.), formatted I/O, stream errors, kernel support for files, System calls, library functions, file descriptors, low level file access - usage of open, creat, read, write, close, lseek, stat family, umask, dup, dup2, fcntl, file and record locking. file and directory management - chmod, chown, links(soft links & hard links - unlink, link, symlink), mkdir, rmdir, chdir, getcwd, Scanning Directories-opendir, readdir, closedir,rewinddir, seekdir, telldir functions.

UNIT-III

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

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

UNIT-IV

Inter process 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 (AUTONOMOUS)

II MCA I- Semester

L	T	P	C
3	1	0	3

(16MC815) DATAWAREHOUSING AND MINING

Course Objectives:

- To expose the students to the concepts of Data warehousing Architecture and implementation
- To Understand Data mining principles and techniques and Introduce DM as a cutting edge business intelligence
- To learn to use association rule mining for handling large data
- To understand the concept of classification for the retrieval purposes
- To know the clustering techniques in details for better organization and retrieval of data
- To identify Business applications and Trends of Data mining

Course Outcomes:

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

- Store voluminous data for online processing
- Preprocess the data for mining applications
- Apply the association rules for mining the data
- Design and deploy appropriate classification techniques
- Cluster the high dimensional data for better organization of the data
- Discover the knowledge imbibed in the high dimensional system
- Evolve Multidimensional Intelligent model from typical system
- 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 Backpropagation, 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 & Micheline Kamber, 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* – Paulraj Ponnaiah 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.
6. *Data Mining Introductory and advanced topics*–Margaret H Dunham, Pearson education
7. *Data Mining Techniques* – Arun K Pujari,2nd edition, Universities Press.
8. *Data Mining*, V.Pudi and P.Radha Krishna, Oxford University Press.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

II MCA I- Semester

L	T	P	C
0	0	4	2

(16MC816) UML LAB

1. The student should take up the case study of Unified Library application which is mentioned in the theory, and Model it in different views i.e. Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.
2. Student has to take up another case study of his/her own interest and do the same whatever mentioned in first problem. Some of the ideas regarding case studies are given in reference books which were mentioned in theory syllabus can be referred for some idea.
3. Design the UML diagrams for the Library management System
4. Design the UML diagram for the Student Marks management System
5. Understanding the Forward engineering and Reverse engineering for a project.
6. **Case Study:** Design the UML diagrams for your own project.

REFERENCES:

1. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Education 2nd Edition.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY-DreamTech India Pvt. Ltd.
3. Appling UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

II MCA I- Semester

L	T	P	C
0	0	4	2

(16MC817) OPERATING SYSTEMS AND LINUX PROGRAMMING LAB

Course Objectives:

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

Course Outcomes:

Upon completion of this course the students should:

- Understand process management, concurrent processes and threads, memory
- management, virtual memory concepts, deadlocks
- Compare performance of processor scheduling algorithms
- Produce algorithmic solutions to process synchronization problems

I) Operating Systems Programs:

1. Simulate the following CPU scheduling algorithms
   a) Round Robin b) SJF c) FCFS d) Priority
2. Simulate all File Organization Techniques
   a) Single level directory b) Two level c) Hierarchical d) DAG
3. Simulate Bankers Algorithm for Dead Lock Avoidance
4. 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.
5. Simulate all page replacement algorithms such as
   a) FIFO b) LRU c) LFU
6. Simulate Paging Technique of memory management.
7. Write C programs that make a copy of a file using
   i) Standard I/O ii) system calls.
8. Write C programs that count the number of blanks in a text file using
   i) Standard I/O ii) system calls.

REFERENCES:

1. *Operating Systems*, P.P. Choudhury, PHI Learning Private Ltd.
2. *Operating Systems*, R.Chopra, S.Chand and Company Ltd

II) LINUX Programs:

Course Objectives:

- To know about Linux operating system and shell scripting.
- To comprehend about Linux utilities of file, process, communication etc.
- To know about system calls related to file, process and IPC.

Course Outcomes:

- Able to use appropriate Linux commands contextually
- Able to write Shell scripts to automate the jobs and processes.
- Able to use system calls related to file, processes and IPC.

Linux Programming Programs:-

Note: Use Bash for Shell scripts.

1. Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
3. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
4. Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
5. Write a shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
6. Write a shell script to list all of the directory files in a directory.
7. Write a shell script to find factorial of a given integer.
8. Write an awk script to count the number of lines in a file that do not contain vowels.
9. Write an awk script to find the number of characters, words and lines in a file.
10. Write a c program that makes a copy of a file using standard I/O and system calls. Implement in C the following Unix commands using System calls

    a. cat b. ls c. mv
11. Write a program that takes one or more file/directory names as command line input and reports the following information on the file.

    a. File type. b. Number of links. c. Time of last access.

    d. Read, Write and Execute permissions.
12. Write a C program to emulate the Unixls –l command.
13. Write a C program to list for every file in a directory, its inode number and file name.
14. Write a C program that demonstrates redirection of standard output to a file.Ex: ls> f1.
15. Write a C program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen.
16. Write a C program to create a Zombie process.
17. Write a C program that illustrates how an orphan is created.

18. Write a C program that illustrates how to execute two commands concurrently with a command pipe. Ex:- ls –l | sort
19. Write C programs that illustrate communication between two unrelated processes using named pipe.
20. Write a C program (sender.c) to create a message queue with read and write permissions to write 3 messages to it with different priority numbers.
21. Write a C program (receiver.c) that receives the messages (from the above message queue as specified in (20)) and displays them.

REFERENCES:

1. *Unix and Shell Programming*, M.G. Venkatesh Murthy, Pearson Education, 2005.
2. *Unix Shells by Example*, 4th Edition, Elllie Quigley, Pearson Education.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

II MCA I- Semester

L	T	P	C
0	0	4	2

(16MC818) DATA WAREHOUSING AND MINING LAB

Course Objectives:

This Lab course is intended to

- Introduce data mining techniques including predictive, descriptive and visualization modeling and their effective use in discovering interesting hidden patterns in large volume of data generated by businesses, science, web, and other sources.
- 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

- Synthesize the data mining fundamental concepts and techniques from multiple perspectives.
- Develop skills and apply data mining tools for solving practical problems
- Advance relevant programming skills.
- 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 orWhynot ? (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*, 3rd Edition by Ian H. Witten, Eibe Frank, Mark A. Hall

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

II MCA I- Semester

L	T	P	C
3	0	0	0

(16HS614) COMPREHENSIVE SOFTSKILLS

Course Description:

Soft Skills is an intangible idea in which the qualities like attitude, ability, integrity, reliability, positivity, flexibility, dependability, punctuality, management, cooperation, habits and practices are combined proficiently to capitalize on a person's work efficacy. Soft Skills do the work of combining all these components in accurate share into skills and shaping them into competencies. Companies opt for, maintain and prop up persons, who are trustworthy, ingenious, principled and good communicators and who are prepared to work under stress. These lessons are developed with a view to create awareness of the importance of the soft skills and assist the learners to improve them.

Course Objectives:

The main objectives of this course are:

- To help the students understand interpersonal skills.
- To support them in building interpersonal skills.
- To enhance the ability to work with others.

Course Outcomes:

- To know the importance of Soft Skills.
- To apply Soft Skills in the different environment.
- To enrich the different levels of Soft Skills to develop their personality.

UNIT-I

Nonverbal Communication – Body Cues – Smiling, Posture, Gesture, Eye-contact – Stage appearance – Interpersonal and Intrapersonal skill Telephonic Etiquette – Dos and Don's of Telephonic Conversation

UNIT-II

Self-exploration – Self Discovery – Self acceptance – Self-esteem – Self-confidence – Personal grooming – Attitudes – Confidence building. Interpersonal relationship in the present context – Kinds of relationships – Team building – Formation of team

UNIT-III

Vision and Goal setting – Personal goal – Career goal – Types of Organization – Deep dive of company profiles – Win-win situation – Proactive skills – Entrepreneurial skills and model start-ups- Developing Mind skills – quizzes – General knowledge – Puzzles – Reading Comprehension - Spell Bee - Seminar – Who is who? – Biographies

UNIT-IV

Flight Leadership: Assessing Leadership qualities – Experiential learning of leadership skills exercise in team work Time and Stress Management: Importance of Time Management – The art of prioritizing and scheduling – Stress and Source of Stress Types of Stress – Managing stress

UNIT-V

Change: Coping skills – Critical and Adaptive Mindsets – Changes in Career/ Life/ people – Just A Minute – Mock GDs and Mock Interviews

REFERENCES:

1. *Business Communication*, Aruna Koneru
2. *Effective Tech Communication*, Rizvi, Tata McGraw – Hill Education, 2007.
3. *Reading Extra*, Liz Driscoll, Cambridge University Press, 2004.
4. *Speak Well*, Jayashree Mohanraj et al, Orient Blackswan, 2013.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

II MCA II- Semester

L	T	P	C
3	1	0	3

(16MC819) SOFTWARE ENGINEERING

Course Objectives:

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

Course Outcomes:

- Get an insight into the processes of software development
- Able to understand the problem domain for developing SRS and various models of software engineering
- Able to Model software projects into high level design using DFD diagrams
- Able to Measure the product and process performance using various metrics
- 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 use cases. Requirements modeling, Requirements validation, Software requirements specification (SRS), Requirements management.

Requirements modeling: Scenario based, class based, Web/Mobile App based modeling.

Software Project Estimation: Empirical estimation models.

UNIT-III

Design Concepts: Software design quality guidelines and attributes, Design concepts, Design model.

Architecture Design: Architecture and its importance, Architectural Styles, Data design, Architectural design.

Component-level Design: Component, Designing Class based components, Component-level design for web and mobile apps, component based-development.

User Interface Design: Golden rules, User interface analysis and design, interface analysis, interface design steps.

Pattern Based Design: Design patterns, Pattern based software design, Architectural patterns, Component level design patterns, User interface design patterns.

Web app design: Interface design, Content design, Navigation design.

Mobile app Design: Developing mobile app, design practices.

UNIT-IV

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

Testing Conventional Applications: Software testing fundamentals, White-Box testing: Basis path testing, condition (predicate) testing, data flow testing, loop testing, Black box testing: Equivalence partitioning, Boundary value analysis, Graph based testing methods.

Testing Object Oriented Applications: OO testing methods, Testing methods applicable at class level, Interclass test case design.

Testing Web Applications and Mobile Applications, Security Engineering and risk analysis, Security assurance.

UNIT-V

Umbrella Activities:

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

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

TEXT BOOKS:

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

REFERENCES:

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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

II MCA II- Semester

L	T	P	C
3	1	0	3

(16MC820) WEB TECHNOLOGIES

Course Objectives

- To understand the basics of java bean
- To understand the web server and the server-side programming
- To understand the DB connections and MVC architecture, JSP
- To understand the concepts of PHP and AJAX

Course Outcomes

Student be able to:

- Do the server side programming, maintain sessions.
- Establish the DB connections and access the data.
- 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, 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 by Herbert Schildt. TMH.
2. HTML5 and CSS3, 7th edition, by Elizabeth Castro and Bruce Hyslop.
3. Beginning Web Programming-Jon Duckett, WROX.
4. Ajax: A beginner’s guide by stevenHolzner

REFERENCES:

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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

II MCA II- Semester

L	T	P	C
3	1	0	3

(16MC821) COMPUTER GRAPHICS

Course Objectives:

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

Course Outcomes:

- Gain proficiency in 3D computer graphics API programming
- Enhance the perspective of modern computer system with modeling, analysis and interpretation of 2D and 3D visual information

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*, second edition in C, Foley, VanDam, Feiner and Hughes, Pearson Education.

REFERENCES:

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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

II MCA II- Semester

L	T	P	C
3	1	0	3

(16MC822) BIG DATA ANALYTICS
(DEPARTMENT ELECTIVE – I)

Course Objectives:

- To explore the fundamental concepts of big data analytics
- To learn to analyze the big data using intelligent techniques.
- To understand the various search methods and visualization techniques.
- To learn to use various techniques for mining data stream.
- To understand the applications using Map Reduce Concepts.

Course Outcomes:

The students will be able to:

- Work with big data platform
- Analyze the big data analytic techniques for useful business applications.
- Design efficient algorithms for mining the data from large volumes.
- Analyze the HADOOP and Map Reduce technologies associated with big data analytics
- Explore on Big Data applications Using Pig and Hive
- Understand the fundamentals of various bigdata analysis techniques

UNIT-I

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

REFERENCES:

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

II MCA II- Semester

L	T	P	C
3	1	0	3

UNIT-I

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

REFERENCES:

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

II MCA II- Semester

L	T	P	C
3	1	0	3

(16MC824) DISTRIBUTED SYSTEMS
(DEPARTMENT ELECTIVE – I)

Course Objectives:

- 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;
- 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;
- 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;
- 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;
- To build distributed system software using basic OS mechanisms as well as higher - level middleware and languages

Course Objectives:

After completion of this course, the student is:

- 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;
- 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;
- 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;
- 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

UNIT-I

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.

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
(AUTONOMOUS)

II MCA II- Semester

L	T	P	C
3	1	0	3

(16MC825) SERVICE ORIENTED ARCHITECTURE
(DEPARTMENT ELECTIVE – I)

Course Objectives:

- To provide fundamental concepts of Service Oriented Architecture..
- To gain knowledge about SOAP, UDDI and XML to create web services.
- To know about the Cloud Computing architecture and services.

Course Outcomes:

- Known about the basic principles of service oriented architecture , its components and techniques
- Understand the architecture of web services
- Able to design and develop web services using protocol
- Understand technology underlying the service design
- Acquire the fundamental knowledge of cloud computing

UNIT-I

SOA Basics: Roots of SOA, Characteristics of SOA, Comparing SOA to client-server and distributed internet architectures, Anatomy of SOA, How components in an SOA interrelate, Principles of service orientation, Service Layers.

UNIT-II

Xml and Web Services: XML structure, Elements, Creating Well-formed XML, Name Spaces, Schema Elements, Types, Attributes, XSL Transformations, Parser, WebServices Overview, Architecture.

UNIT-III

WSDL, SOAP and UDDI: WSDL, Overview of SOAP, HTTP, XML-RPC

SOAP: Protocol, Message Structure, Intermediaries – Actors – Design Patterns And Faults, SOAPWith Attachments – UDDI.

UNIT-IV

SOA in J2EE and .NET: SOA platform basics – SOA support in J2EE – Java API for XML-based web services(JAX-WS), Java architecture for XML binding (JAXB) – Java API for XML Registries(JAXR), Java API for XML based RPC (JAX-RPC) – JAX-RS SOA support in .NET – ASP.NET webservices.

UNIT-V

Introduction to Cloud Computing: Vision of Cloud computing, Cloud Definition, Characteristics and Benefits, Virtualization, Cloudcomputing Architecture, Cloud Reference Model, Types of Clouds, Cloud Platforms in Industry.

TEXT BOOKS:

1. Thomas Erl, "*Service-Oriented Architecture: Concepts, Technology*, and Design", Pearson Education, 2006.
2. Heather Williamson, "*XML, The Complete Reference*", McGraw Hill Education, 2012

REFERENCES:

1. Frank. P. Coyle, "*XML, Web Services And The Data Revolution*", Pearson Education, 2002.
2. SandeepChatterjee, James Webber, "*Developing Enterprise Web Services. An Architect's Guide*", Pearson Education, 2005.
3. Newcomer, Lomow, "*Understanding SOA with Web Services*", Pearson Education, 2005.
4. Dan woods and Thomas Mattern, "*Enterprise SOA designing IT for Business Innovation*", O'REILLY, First Edition, 2006.
5. KK RajkumarBuyya, Christian Vecchiola, S. ThamaraiSelvi, "*Mastering Cloud Computing*", McGraw Hill Education, 2013.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

II MCA II- Semester

L	T	P	C
3	1	0	3

(16MC826) INFORMATION SECURITY
(DEPARTMENT ELECTIVE-II)

Course Objectives:

- To understand the basics of cryptography
- learn to find the vulnerabilities in programs and to overcome them,
- know the different kinds of security threats in networks and its solution
- know the different kinds of security threats in databases and solutions available
- learn about the models and standards for security

Course Outcomes:

- Apply cryptographic algorithms for encrypting and decryption for secure data transmission
- Understand the importance of Digital signature for secure e-documents exchange
- Understand the program threats and apply good programming practice
- Get the knowledge about the security services available for internet and web applications
- Gain the knowledge of security models and published standards

UNIT-I

Security Goals, Security Attacks (Interruption, Interception, Modification and Fabrication), Security Services (Confidentiality, Authentication, Integrity, Non-repudiation, access Control and Availability) and Mechanisms, A model for Internetwork security.Internet Standards and RFCs.

Conventional Encryption Principles & Algorithms(DES, AES, RC4), Block Cipher Modes of Operation, Location of Encryption Devices, Key Distribution.

UNIT-II

Public key cryptography principles, public key cryptography algorithms (RSA, RABIN, ELGAMAL, Diffie-Hellman, ECC), Key Distribution. Approaches of Message Authentication. Secure Hash Functions(SHA-512, WHIRLPOOL) and HMAC Digital Signatures. Comparison, Process- Need for Keys, Signing the Digest, Services, Attacks on Digital Signatures, Kerberos, X.509 Directory Authentication Service.

UNIT-III

Email Security: Pretty Good Privacy (PGP) and S/MIME.

IP Security Overview, IP Security Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations and Key Management

UNIT-IV

Web Security Requirements, Secure Socket Layer (SSL) and Transport Layer Security (TLS), Secure Electronic Transaction (SET).

UNIT-V

Basic concepts of SNMP, SNMPv1 Community facility and SNMPv3. Intruders, Viruses and related threats, Virus Counter measures. Firewall Design principles, Trusted Systems, Intrusion Detection Systems

TEXT BOOKS:

1. *Network Security Essentials (Applications and Standards*) by William Stallings Pearson Education, 2008.
2. *Cryptography & Network Security* by Behrouz A. Forouzan, TMH 2007.
3. *Cryptography and Network Security* by William Stallings, Fourth Edition,Pearson Education 2007.

REFERENCES:

1. *Information Security* by Mark Stamp, Wiley – INDIA, 2006.
2. *Fundamentals of Computer Security*, Springer.
3. *Network Security: The complete reference*, Robert Bragg, Mark Rhodes, TMH
4. *Computer Security Basics* by Rick Lehtinen, Deborah Russell &G.T.Gangemi Sr., SPD O'REILLY 2006.
5. *Modern Cryptography* by Wenbo Mao, Pearson Education 2007.
6. *Principles of Information Security*, Whitman, Thomson.
7. *Information Systems Security*,Godbole,Wiley Student Edition.
8. *Cryptography and Information Security*,V.K.Pachghare,PHI.
9. *Network Security and Cryptography*,B.Menezes,Cengage Learning

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

II MCA II- Semester

L	T	P	C
3	1	0	3

(16MC827) SOCIAL NETWORKS AND SEMANTIC WEB (DEPARTMENT ELECTIVE – II)

Course Objectives:

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

Course Outcomes:

- Understand semantic web basics, architecture and technologies
- Able to represent data from a chosen problem in XML with appropriate semantic tags obtained or derived from the ontology
- Able to understand the semantic relationships among these data elements using Resource Description Framework (RDF)
- Able to design and implement a web services application that “discovers” the data and/or other web services via the semantic web
- 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 (AUTONOMOUS)

II MCA II- Semester

L	T	P	C
3	1	0	3

(16MC828) GEOLOGICAL INFORMATION SYSTEMS (DEPARTMENT ELECTIVE – II)

Course Objectives:

- Understand the basic concepts of Geological information systems.
- To provide an exposure to spatial database structures and their utility in GIS.
- Understand the process of scanning, digitizing and georeferencing.
- To introduce the raster and vector geoprocessing capabilities of GIS.

Course Outcomes:

- Understand GIS concepts and spatial data representation
- Able to design spatial data input in raster form as well as vector form
- Understand vector data analysis and output functions
- Understand raster data geo processing
- Able to design a GIS model for real world problem

UNIT-I

Spatial data representation: GIS – Definition and related terminology, Components of GIS: Data, Technology, Application– digital representation of geospatial data – raster, vector, object oriented, geo database model-analysis.

UNIT-II

Data Digitization and Preparation: Characteristics of raster data processing, raster File format, Acquiring and handling raster Data, Georeferencing, Preprocessing, mosaicking, Linking digital databases: ODBC – GPS data integration. Characteristics of Vector Geoprocessing, Vector Data Input Digitizer: Principles, Co-ordinate transformation, Graphical data editing, Scanner: Principles, On Screen Digitization, post scanning, importing, data editing

UNIT-III

Raster Data Analysis: Raster Geospatial Data Analysis-Local operations: Reclassification, Logical and Arithmetic overlayoperations. Neighbourhood operations: Aggregation, Filtering, Slope and Aspect map. Extendedneighbourhood operations: Statistical Analysis, Proximity, Connectivity operations, Buffering,View shed analysis. Regional operations: Area, Perimeter, Shape, Identification of region andClassification-output functions of Raster geoprocessing

UNIT-IV

Vector Data Processing: Non-topological analysis: Attribute database query, SQL, Summary statistics, statistical computation calculation, quantification. Address geocoding, Topological analysis Feature based topological functions, overlay-buffering. Layer based topological function, Reclassification, Aggregation, Overlay analysis- Point-in-polygon, Line-in-polygon, Polygon-on-polygon: Clip, Erase, Identity, Union, Intersection. Network based Geoprocessing, Output functions.

UNIT-V

GIS Modeling and Applications: Spatial modeling – External, Conceptual, Logical, Internal. GIS Modeling with case study- spatial data mining-DEM- introduction and applications

TEXT BOOKS:

1. Lo, C.P. and Yeung, Albert K.W., *Concepts and Techniques of Geographic Information Systems*, Prentice Hall, 2/E,2009
2. Peter A. Burrough, Rachael A. McDonnell, *Principles of GIS*, Oxford University Press, 2000

REFERENCES:

1. Kang-Tsung Chang ,*Introduction to Geographic Information Systems*, McGraw-Hill Higher Education, 2006
2. Robert Laurini and Derek Thompson, *Fundamentals of Spatial Information Systems*, Academic Press, 1996
3. Paul A. Longley, Mike Good child , David J. Maguire, *Geographic Information Systems and Science*, JohnWiley & Sons Inc ,2011

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

II MCA II- Semester

L	T	P	C
3	1	0	3

(16MC829) .NET TECHNOLOGIES
(DEPARTMENT ELECTIVE – II)

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 App Domains - 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 – Data Tables - Controlling table views with Data Views and Data Relation 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 - Implementing Security - 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 - XML and ADO.NET

Web Services:XML Serialization in the .NET Framework -SOAP Fundamentals- Using SOAP with the .NET Framework.-Web Services protocol and standards – WSDL- Documents - Overview of UDDI - Calling a Web Service from a Browser - Calling a Web -Service by Using a Proxy - Creating a simple web service - Developing a WCF Web Service - Creating and Consuming AJAX-Enabled Web Services - Introducing REST and JSON

UNIT-V

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

TEXT BOOKS:

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

REFERENCES:

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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

II MCA II- Semester

L	T	P	C
0	0	4	2

(16MC830) SOFTWARE ENGINEERING LAB

Course Objectives:

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

Course Outcomes:

- Able to prepare various phases of Spiral model.
- Able to draw E-R diagram, DFD, UML diagrams for the project
- Able to develop PERT and CPM project schedule methods.
- 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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

II MCA II- Semester

L	T	P	C
0	0	4	2

(16MC831) WEB TECHNOLOGIES LAB

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.

1. Create a Cookie and add these four user id and passwords to this Cookie.
2. 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 get Init Parameters() 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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

II MCA II- Semester

L	T	P	C
0	0	4	2

(16MC832) COMPUTER GRAPHICS LAB

Course Objectives:

The Objective of this course is to make the students

- To understand the basic concepts of computer graphics.
- To design scan conversion problems using C programming.
- To apply clipping and filling techniques for modifying an object.
- To understand the practical implementation of modeling, Translation, rendering, viewing of objects in 2D

Course Outcomes:

- Able to analyze the basic concepts of computer graphics.
- Able to design scans conversion problems using C programming.
- Able to implement Line, circle and ellipse algorithms
- Able to implement clipping and filling techniques for modifying an object.
- Able to analyze and implement the concepts of different typeof geometric transformation operations on objects in 2D and 3D.
- Able to implement the practical programs of modeling, viewing of objects in 2D

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

1. Write a program to draw a Line using Mid-Point Bresenham’s algorithm.
2. Write a program to draw a Line using DDA algorithm.
3. Write a program to draw a Circle using Bresenham’s algorithm.
4. To draw an ellipse using Mid-Point ellipse drawing algorithm.
5. Write a program to implement Boundary-fill & Flood-fill algorithm.
6. Write a program to implement Scan-line Polygon-fill algorithm.
7. Write a program to implement line clipping concepts using Cohen-Sutherland algorithm.
8. Write a program to implement line clipping concepts using Liang-Barsky algorithm.
9. Write a program to implement Polygon clipping concepts using Sutherland-Hodgeman algorithm.
10. Write a program to implement 2D Translation operations.
11. Case Study:Design a simple animation program.

REFERENCES:

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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

II MCA II- Semester

L	T	P	C
3	0	0	0

(16HS617) APTITUDE PRACTICE-II

Course Objectives:

After thorough learning of Quantitative Aptitude and Reasoning, a student:

- Will be able to critically evaluate various real life situations by resorting to analysis of key issues and factors.
- Will be able to read between the lines and understand various language structures.
- Will be able to demonstrate various principles involved in solving mathematical problems and thereby reducing the time taken for performing job functions.

Course Outcomes:

At the end of the course, students would be expected to:

- Have developed the subtle way of approaching in the candidate.
- Have acquired the decision making with in no time.
- Have acquired logical thinking during professional tenure.
- Have obtained quick decision making skills.

UNIT-I

Permutations and Combinations: Definition of permutation, Problems on Permutations, Definition of Combinations, Problems on Combinations.

Probability: Definition of Probability, Problems on coins, Dice, Deck of Cards.

UNIT-II

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

UNIT-III

Cubes: Basics of a cube, Formulae for finding volume and surface area of a cube, finding the minimum number of cuts when the number of identical pieces are given, Finding the maximum number of pieces when cuts are given, Problems on painted cubes of same and different colors, Problems on cuboids, Problems on painted cuboids, Problems on diagonal cuts.

UNIT-IV

Number and letter series: Difference series, Product series, Squares series, Cubes series, Alternate series, Combination series, miscellaneous series, Place values of letters

Number and Letter Analogies: Definition of Analogy, Problems on number analogy,

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

UNIT-V

Coding and decoding: Coding using same set of letters, Coding using different set of letters, Coding into a number, Problems on R-model.

Directions: Solving problems by drawing the paths, finding the net distance travelled, finding the direction, Problems on clocks, Problems on shadows, Problems on damaged compass, Problems on direction sense using symbols and notations.

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

TEXT BOOKS:

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

REFERENCE BOOKS:

1. *Quantitative Aptitude*, R S Agarwal, S Chand,
2. *Quantitative Aptitude* - G. L BARRONS
3. *Quantitative Aptitude* – Abhijit Guha McGraw Hills
4. *Magical Book on Quicker Maths*, Tyra, BSC publishing company

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

III MCA I- Semester

L	T	P	C
3	1	0	3

(16MC833) SOFTWARE TESTING

Course Objectives:

- To know the behavior of the testing techniques to detect the errors in the software
- To understand standard principles to check the occurrence of defects and its removal.
- To learn the functionality of automated testing tools
- To understand the models of software reliability.

Course Outcomes:

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

UN IT-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 (AUTONOMOUS)

III MCA I- Semester

L	T	P	C
3	1	0	3

(16MC834) MOBILE APPLICATION DEVELOPMENT

Course Objectives:

- Describe those aspects of mobile programming that make it unique from programming for other platforms
- Critique mobile applications on their design pros and cons
- Utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces
- Program mobile applications for the Android operating system that use basic and advanced phone features
- Deploy applications to the Android marketplace for distribution

Course Outcomes:

- Students understood the aspects of mobile programming that make it unique from programming for other platforms
- Students program mobile applications for the Android operating system by use basic and advanced phone features
- Also 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*, 4th edition, Addision Wesley by Joseph Annuzzi Jr., Lauren Darcey, Shane Conder.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

III MCA I- Semester

L	T	P	C
3	1	0	3

(16MC835) SOFTWARE PROJECT MANAGEMENT

Course Objectives:

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

Course Outcomes:

- Understand the activities during the project scheduling of any software application.
- Learn the risk management activities and the resource allocation for the projects.
- Can apply the software estimation and recent quality standards for evaluation of the software projects.
- Acquire knowledge and skills needed for the construction of highly reliable software project.
- 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. Richard H. Thayer: "*Software Engineering Project Management*", 1997, IEEE Computer Society.
2. Shere K. D: "*Software Engineering and Management*", 1998, PHI.
3. S. A. Kelkar, "*Software Project Management: A Concise Study*", PHI.
4. Hughes Cotterell, "*Software Project Management*", 2e, TMH.
5. Kaeron Conway, "*Software Project Management from Concept to Development*", Dream Tech.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

III MCA I- Semester

L	T	P	C
3	1	0	3

(16MC836) CYBER SECURITY
(DEPARTMENT ELECTIVE – III)

Course Objectives:

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

Course Outcomes:

After learning the course the students should be able to:

- Understand cyber-attack, types of cybercrimes, cyber laws and also how to protect them self and ultimately society from such attacks

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 VsStateful 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, Open SSL 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

REFERENCES:

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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

III MCA I- Semester	**L**	**T**	**P**	**C**
	3	**1**	**0**	**3**

(16MC837) IMAGE PROCESSING (DEPARTMENT ELECTIVE – III)

Course Objectives:

- The fundamentals of digital image processing
- Understand image transform used in digital image processing
- Understand image enhancement techniques used in digital image processing
- Understand image restoration techniques and methods used in digital image processing
- Understand image compression and Segmentation used in digital image processing

Course Outcomes:

- Able to enhance images using enhancement techniques.
- Able to restore images using restoration techniques and methods used in digital image processing
- Able to compress images using compression techniques used in digital image processing

UNIT-I

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
(AUTONOMOUS)

III MCA I- Semester

L	T	P	C
3	1	0	3

(16MC838) ARTIFICIAL INTELLIGENCE
(DEPARTMENT ELECTIVE – III)

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

REFERENCES:

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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

III MCA I- Semester

L	T	P	C
3	1	0	3

(16MC839) INFORMATION RETRIEVAL SYSTEMS
(DEPARTMENT ELECTIVE – III)

Course Objectives:

- Learn the important concepts, algorithms, and data/file structures that are necessary to specify, design, and implement Information Retrieval (IR) systems.
- Recognize the Boolean Model, Vector Space Model, and Probabilistic Model.
- Understand retrieval utilities.
- Understand different formatting tags.
- Understand cross-language information retrieval.
- Determine the efficiency.

Course Outcomes:

- Use different information retrieval techniques in various application areas
- Apply IR principles to locate relevant information large collections of data
- Analyse performance of retrieval systems when dealing with unmanaged data sources
- Implement retrieval systems for web search tasks.

UNIT-I

Introduction: Definition, Objectives, Functional Overview, Relationship to DBMS, Digital libraries and Data Warehouses. Information Retrieval System Capabilities **-** Search, Browse, Miscellaneous.

UNIT-II

Cataloging and Indexing: Objectives, Indexing Process, Automatic Indexing, Information Extraction. Data Structures: Introduction, Stemming Algorithms, Inverted file structures, N-gram data structure, PAT data structure, Signature file structure, Hypertext data structure. Automatic Indexing: Classes of automatic indexing, Statistical indexing, Natural language, Concept indexing, Hypertext linkages. Document and Term Clustering: Introduction, Thesaurus generation, Item clustering, Hierarchy of clusters

UNIT-III

User Search Techniques: Search statements and binding, Similarity measures and ranking, Relevance feedback, Selective dissemination of information search, weighted searches of Boolean systems, Searching the Internet and hypertext. Text Search Algorithms: Introduction, Software text search algorithms, Hardware text search systems.

UNIT-IV

Information System Evaluation: Introduction, Measures used in system evaluation, Measurement example – TREC results. Parallel and Distribute IR - Parallel Computing, Performance Measures, Parallel IR - MIMD and SIMD Architectures, Distributed IR –

Collection Partitioning, Source Selection, Query Processing, Web Issues, Trends and Research Issues.

UNIT-V

Multimedia Information Retrieval: Models and Languages, Data Modeling, Query Languages, Indexing and Searching. Libraries and Bibliographical Systems**:** Online IR Systems, OPACs, Digital Libraries.

TEXT BOOKS:

1. *Information Storage and Retrieval Systems: Theory and Implementation* By Kowalski,Gerald, Mark T Maybury Kluwer Academic Press, 2000.
2. *Modern Information Retrival* by Ricardo Baeza-Yates, Pearson Education, 2007.

REFERENCES:

1. *Information Retrieval Data Structures and Algorithms*By William B Frakes, Ricardo Baeza-Yates, Pearson Education, 1992.
2. *Information Storage &Retieval* by Robert Korfhage – John Wiley & Sons.
3. *Introduction to Information Retrieval* by Christopher D. Manning andPrabhakarRaghavan, Cambridge University Press, 2008.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

III MCA I- Semester

L	T	P	C
3	1	0	3

(16MC840) M-COMMERCE (DEPARTMENT ELECTIVE – IV)

Course Objectives:

- To understand the E-commerce strategies and value chains
- To understand the M-commerce services
- To understand M-commerce infrastructure and applications.
- To know the availability of latest technology and applications of M- commerce in various domains.
- To apply mobile commerce in business-to-business application.

Course Outcomes:

- Able to apply E – commerce principles in market place.
- Able to apply M – commerce principles to various business domains
- Understand the theory and applications of M-commerce in business domain
- Get an exposure to current technological advancements in M-commerce

UNIT-I

Electronic Commerce: Introduction, The e-commerce environment, the e-commerce marketplace, Focus on portals, Location of trading in the marketplace, Commercial arrangement for transactions, Focus on auctions, Business models for e-commerce, Revenue models - Focus on internet start-up companies, the dot-com, E-commerce versus E-business.

UNIT-II

Mobile Commerce: Introduction, Infrastructure of M-Commerce, Types of Mobile Commerce Services, technologies of Wireless Business, Benefits and Limitations, Support, Mobile Marketing & Advertisement, Non–Internet Applications In M–Commerce, Wireless/Wired Commerce Comparisons

UNIT-III

Mobile Commerce - Technology: A Framework for The Study Of Mobile Commerce, NTT Docomo's I–Mode, Wireless Devices for Mobile Commerce, Towards A Classification Framework For Mobile Location Based Services, Wireless Personal and Local Area Networks, The Impact of Technology Advances on Strategy Formulation In Mobile Communications Networks

UNIT-IV

Mobile Commerce - Theory and Applications: The Ecology of Mobile Commerce, The Wireless Application Protocol, Mobile Business Services, Mobile Portal – Factors Influencing The Adoption of Mobile Gaming Services – Mobile Data Technologies And Small Business

Adoption and Diffusion, M–Commerce In The Automotive Industry, Location-Based Services: Criteria For Adoption And Solution Deployment, The Role of Mobile advertising in building a Brand, M–Commerce Business Models

UNIT-V

Business-to-Business Mobile E-Commerce: Enterprise Enablement – Email And Messaging, Field Force Automation (Insurance, Real Estate, Maintenance, Healthcare), Field Sales Support (Content Access, Inventory), Asset Tracking and Maintenance/Management, Remote IT Support, Customer Retention (B2C Services, Financial, Special Deals), Warehouse Automation, Security.

TEXT BOOKS:

1. Dave Chaffey, "*E-Business and E-Commerce Management*", Third Edition, 2009, Pearso Education
2. Brian E. Mennecke, Troy J. Strader, "*Mobile Commerce: Technology, Theory and Applications*", Idea Group Inc., IRM press, 2003.

REFERENCES:

1. P. J. Louis, "*M-Commerce Crash Course*", McGraw- Hill Companies February 2001.
2. Paul May, "*Mobile Commerce: Opportunities, Applications, and Technologies of Wireless Business*" Cambridge University Press March 2001.
3. Michael P. Papazoglou, Peter M.A. Ribbers, '*e-business organizational and Technical foundation'*, Wiley India 2009
4. Dr.Pandey, Saurabh Shukla *E-commerce and Mobile commerce Technologies*, Sultan chand, 2011

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

III MCA I- Semester

L	T	P	C
3	1	0	3

(16MC841) CLOUD COMPUTING
(DEPARTMENT ELECTIVE – IV)

Course Objectives:

- To introduce the broad perceptive of cloud architecture and model
- To understand the concept of Virtualization and design of cloud Services
- To be familiar with the lead players in cloud.
- To understand the features of cloud simulator
- To apply different cloud programming model as per need.
- To learn to design the trusted cloud Computing system

Course Outcomes:

- Compare the strengths and limitations of cloud computing
- Identify the architecture, infrastructure and delivery models of cloud computing
- Apply suitable virtualization concept.
- Choose the appropriate cloud player, Programming Models and approach.
- Address the core issues of cloud computing such as security, privacy and Interoperability
- 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*”.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

III MCA I- Semester

L	T	P	C
3	1	0	3

(16MC842) DESIGN PATTERNS
(DEPARTMENT ELECTIVE – IV)

Course Objectives:

- Indicate which underlying object oriented design principle(s) it is based on.
- Explain what specific object oriented design problem the pattern solves.
- Provide a specific context for each pattern in which it can be applied.
- Draw a high level class diagram in UML for each pattern.
- Explain how the different components of the pattern collaborate with each other.
- List the consequences of applying each pattern to the overall software quality of a system.
- Implement this pattern in Java to a real world problem

Course Outcomes:

- Students demonstrate a thorough understanding of patterns and their underlying principles
- Students know what design pattern to apply to a specific problem
- Students demonstrate what tradeoffs need to be made when implementing a design pattern
- 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* By Erich Gamma, Pearson Education

REFERENCES:

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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

III MCA I- Semester

L	T	P	C
3	1	0	3

(16MC843) COGNITIVE COMPUTING
(DEPARTMENT ELECTIVE – IV)

Course Objectives:

- To introduce the broad perceptive of Cognitive Computing
- To understand the concept of Analytics in Cognitive computing
- To be familiar with the IBMs Watson
- To understand the future applications of Cognitive computing

Course Outcomes:

- Understand the broad perceptive of Cognitive Computing
- Understand the concept of Analytics in Cognitive computing
- Using the IBMs Watson
- Designing the applications in Cognitive computing

UNIT-I

The Foundation of Cognitive Computing, Design principles of Cognitive Computing – components of cognitive system, building the corpus, bringing data into cognitive system, machine learning, hypothesis generation and scoring, presentation and visualization services. Natural language processing in support of cognitive system – the role of NLP in a cognitive system, semantic web, applying natural language technologies to business problems.

UNIT-II

Relation between Big data and cognitive computing – dealing with human generated data, defining big data, architectural foundation for big data, analytical data warehouses, hadoop, data in motion and streaming data, integration of big data with traditional data.

Representing knowledge in taxonomies and ontologies – representing knowledge, defining taxonomies and ontologies, explaining how to represent knowledge, models for knowledge representation.

Applying advanced analytics to cognitive computing – path to cognitive computing, key capabilities in advanced analytics, using advanced analytics to create value.

UNIT-III

Role of cloud and distributed computing in cognitive computing – leveraging distributed computing for shared resources, characteristics of cloud computing, cloud computing models, delivering models of cloud, managing workloads, security and governance, data integration and management in the cloud. Business implications of cognitive computing.

UNIT-IV

IBMs Watson as a cognitive system – Watson defined, advancing the research, preparing Watson for commercial applications, the components of Deep QA architecture. The process of building a cognitive application – emerging cognitive platform, defining the objective and domain, understanding the intended users and defining their attributes, defining questions and exploring insights, creating and refining corpora, training and testing the application.

UNIT-V

Emerging cognitive computing areas – characteristics of ideal markets for cognitive computing, vertical markets and industries.
Future applications for cognitive computing – requirements for the next generation, technical advancements, the future, emerging innovations.
Case study – Cognitive computing in government, Cognitive healthcare application

TEXT BOOKS:

1. *Cognitive Computing and Big Data Analytics* by Judith S. Hurwitz, Marcia Kaufman, Adrian Bowles, Wiley publications 2015.

REFERENCES:

1. *www.ibm.com*

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

III MCA I- Semester

L	T	P	C
0	0	4	2

(16MC844) SOFTWARE TESTING LAB

Course Objectives:

- To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
- To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
- To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
- To expose the advanced software testing topics, such as object-oriented software testing methods, and component-based software testing issues, challenges, and solutions.
- To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.
- To understand software test automation problems and solutions.
- To learn how to write software testing documents, and communicate with engineers in various forms.
- To gain the techniques and skills on how to use modern software testing tools to support software testing projects.

Course Outcomes:

By the end of the course, you should:

- Have an ability to apply software testing knowledge and engineering methods.
- Have an ability to design and conduct a software test process for a software testing project.
- Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.
- Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.
- Have an ability to use various communication methods and skills to communicate with their teammates to conduct their practice-oriented software testing projects.
- Have basic understanding and knowledge of contemporary issues in software testing, such as component-based software testing problems
- Have an ability to use software testing methods and modern software testing tools for their testing projects.

Exercises:

1. To learn to use the testing tools to carry out the functional testing.
2. To learn to use the testing tools to carry out the load/stress testing.
3. To learn to use the Win Runner/QTP for functional testing.
4. To learn to use the Load Runner for Load/Stress testing.

5. To learn to use the Test Director for test management.
6. **Case Study**: Design a simple application and perform the testing and design the test cases for the testing process.

REFERENCES:

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

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

III MCA I- Semester

L	T	P	C
0	0	4	2

(16MC845) MOBILE APPLICATION DEVELOPMENT LAB

Course Objectives:

The student should be made to:

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

Course Outcomes:

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

- Design and implement various mobile applications using emulators.
- 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.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

III MCA I- Semester

L	T	P	C
0	0	4	2

(16MC846) SOFTWARE PROJECT MANGEMENT LAB

Course Objective

- The main purpose of conducting this LAB is to give an idea about the working of MS Project specially related with Tasks creation, Resources allocation, formatting and printing of a project plan, tracking the progress on tasks in a project.

Course Outcomes

Students will get familiar with HOW TO:

- Start MS Project, Create a Project Plan from a template, Switch to a different view, View a report, Create a visual report.
- Create a new project plan & its start date, set working & non-working time, enter properties about a project plan, enter new tasks in the project, set duration for each task & to create, a milestone task, organizing tasks into phases, link adjacent and non-adjacent tasks, enter a task note, enter a task hyper link, check a Project plan's duration and other statistics, display project's entire duration in Gantt Chart View.
- Setup work (people and equipment) resources, material resources, cost resources, Enter work (people & material) resource pay rates, to make a onetime adjustment to an individual resource's working time, to edit regular work week for an individual resource, to document resources with resource notes.

<u>Programs</u>:

1. Introduction to MS Project
2. Creating task list
3. Linking adjacent and non-adjacent tasks
4. Setting-up resources
5. Resource allocation to tasks
6. Formatting and printing project plan
7. Tracking progress on tasks

REFERENCES:

1. Microsoft Project 2013 Step by Step, by Chafield.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

III MCA I- Semester

L	T	P	C
-	-	4	-

(16HS615) ADVANCED ENGLISH LANGUAGE AND COMMUNICATION SKILLS LAB

Course Description:

The introduction of the Advanced Professional Communication Skills Lab is considered essential at 3rd year level. At this stage the students need to prepare themselves for their careers which may require them to listen to, read, speak and write in English both for their professional and interpersonal communication in the globalized context.

Course Objectives:

This Lab focuses on using computer-aided multimedia instruction for language development to meet the following targets:

- To improve the students’ fluency in English, through a well-developed vocabulary
- To enable them listening spoken English at normal conversational speed by educated English speakers
- To respond appropriately in different social-cultural and professional contexts
- To develop drafting skills among the students.

Course Outcomes:

- Flair in Writing and felicity in written expression
- To enhance job prospects
- Improving Effective Speaking Abilities
- To prepare effective Interview techniques

UNIT-I:

COMMUNICATIVE COMPETENCY:

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

UNIT-II:

TECHNICAL WRITING

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

UNIT-III:

PRESENTATIONAL SKILLS

1. Oral presentation
2. Power point presentation
3. Poster presentation
4. Stage Dynamics

UNIT-IV:

CORPORATE SKILLS

1. Dress code
2. Telephonic skills
3. Net-etiquettes
4. Video conferencing and Chairing Session

UNIT-V:

GETTING READY FOR JOB

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

Minimum Requirements for Advanced Professional Communication Skills Lab:

The English Language Lab shall have two parts:

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

System Requirement (Hardware component):

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

Specifications:

i) P- IV Processor
   - a) Speed 2.8 GHZ
   - b) RAM – 512 MB Minimum
   - c) Hard Disk – 80 GB

ii) Headphones of High quality

Suggested Software:

i) Clarity Pronunciation Power – Part I (Sky Pronunciation)
ii) Clarity Pronunciation Power – Part II
iii) K – Van Advanced Communication Skills
iv) Walden Info Tech Software.

REFERENCES:

1. *Effective Tech Communication*, Rizvi, Tata McGraw – Hill Education, 2007.
2. *Communication skills*, Sanjay Kumar &Pushpalatha, Oxford University Press, 2012.
3. *Writing Tutor*. *Advanced English Learners' Dictionary*, 9th Edition, Oxford Universty Press, 2015.
4. *Powerful Vocabulary Builder*, Anjana Agarwal, New Age International Publishers, 2011.
5. *Listening Extra*, Miles Craven, Cambridge University Press, 2008.
6. *Reading Extra*, Liz Driscoll, Cambridge University Press, 2004.
7. *Writing Extra*, Graham Palmer, Cam Cambridge University Press, 2004.
8. *Speak Well*, Jayashree Mohanraj et al, Orient Blackswan, 2013.

Mode of Evaluation: Written Examination, Day-to-day Assessment