MADANAPALLE INSTITUTE OF TECHNOLOGY &SCIENCE, Madanapalle

(UGC-AUTONOMOUS)

www.mits.ac.in

Estd 1998

DEPARTMENT OF COMPUTER APPLICATIONS

COURSE STRUCTURE
AND
DETAILED SYLLABI

For the students admitted to
Master of Computer Applications from the academic year 2016-17 batch onwards

VISION AND MISSION OF THE INSTITUTION

Vision

Become a globally recognized research and academic institution and thereby contribute to technological and socio-economic development of the nation.

Mission

To foster a culture of excellence in research, innovation, entrepreneurship, rational thinking and civility by providing necessary resources for generation, dissemination and utilization of knowledge and in the process to create an ambience for practice-based learning to the youth for success in their careers.

Quality Policy

Madanapalle Institute of Technology & Science is committed to bring out and nurture the talents and skills of youth in the fields of Engineering and Management to cater to the challenging needs of society and industry by

- Contributing to the academic standing and overall knowledge development of the students
- Providing excellent infrastructure and conductive learning environment.
- Enhancing the competence of faculty and promoting R&D Programs
- Collaborating with institutions and industries.
- Ensuring continual improvement of Quality Management System.

VISION AND MISSION OF THE DEPARTMENT

Vision

To establish a centre of excellence in technical education using the state of the art infrastructure and best of the faculty to produce graduates industry ready.

Mission

- To create professionals with profound knowledge and skill set to dedicate the services for the betterment of the society
- To impart quality education using the best in faculty with balanced curriculum
- To develop moral and ethical values

Curriculum – Regular MCA

Breakup of Courses

Sl. No.	Category	No. of Theory Courses	No. of Practical Courses	Project Works	Seminars	Curriculum Credits	Weightage (%)
1	Programme Core Courses	15	17	2	1	116	75.3
2	Discipline Electives	4	3	-	-	22	14.3
3	Open Electives	4	-	-	-	16	10.4
4	Audit Courses	2	-	-	-	-	-
	Total	25	20	2	1	154	100

COURSE STRUCTURE AND DETAILED SYLLABI
Master of Computer Applications

MCA - I Year I Semester

Course Code	Course Name	L	P	C
14ENG104	English for Communication	4		4
16MCA101	Introduction to Computing	4		4
16MCA102	Programming in C	4		4
16MCA103	Computer Organization	4		4
14ENG201	Communication Skills Practical		3	2
16MCA201	Office Automation Practical		3	2
16MCA202	Programming in C Practical		3	2
	Software Lab – I		**3**	**2**
	Sub Total			**24**

MCA - I Year II Semester

Course Code	Course Name	L	P	C
16MCA104	Database Management Systems	4		4
16MCA105	Data Structures through C++	4		4
16MCA106	Operating Systems	4		4
14MAT105	Discrete Mathematics	4		4
16MCA204	Database Management Systems Practical		3	2
16MCA205	Data Structures through C++ Practical		3	2
16MCA206	Unix and Shell Programming Practical		3	2
	Software Lab – II		3	2
	Audit Course - I			
	Sub Total			**24**

MCA - II Year I Semester

Course Code	Course Name	L	P	C
16MCA107	Data Structures through Python	4		4
16MCA108	Computer Networks	4		4
16MCA109	Software Engineering	4		4
16MCA110	Design & Analysis of Algorithms	4		4
	Open Elective – I	4		4
16MCA207	Data Structures through Python Practical		3	2
16MCA208	Computer Networks Practical		3	2
16MCA209	SE & UML Practical		3	2
16MCA210	Design & Analysis of Algorithms Practical		3	2
	Audit Course - II			
	Sub Total			**28**

MCA - II Year II Semester

Course Code	Course Name	L	P	C
16MCA111	JAVA Programming	4		4
	Discipline Elective – I (16MCA401 – 16MCA405)	4		4
	Discipline Elective – II (16MCA406 – 16MCA410)	4		4
	Open Elective – II	**4**		**4**
	Open Elective – III	**4**		**4**
16MCA211	JAVA Programming Practical		3	2
	Discipline Elective – I (Practical) (16MCA214-16MCA218)		3	2
	Discipline Elective – II (Practical) (16MCA219-16MCA223)		3	2
	Software Lab – III		**3**	**2**
	Sub Total			**28**

MCA - III Year I Semester

Course Code	Course Name	L	P	C
16MCA601	Internship – I			2
16MCA112	Data Warehousing and Data Mining	4		4
16MCA113	Cryptography & Network Security	4		4
	Discipline Elective – III (16MCA411 – 16MCA415)	**4**		**4**
	Discipline Elective – IV (16MCA416 – 16MCA420)	**4**		**4**
	Open Elective – IV	**4**		**4**
16MCA212	Data Warehousing and Data Mining Practical		3	2
16MCA213	Cryptography & Network Security Practical		3	2
	Discipline Elective – III (Practical) (16MCA224-16MCA228)		3	2
	Software Lab – IV		**3**	**2**
	Sub Total			**30**

MCA - III Year II Semester

Course Code	Course Name	L	P	C
16MCA501	Seminar			2
16MCA602	Major Project			18
	Sub Total			**20**

List of Discipline Core Courses
(All Courses Carry Equal Marks (100)

Sl.No.	Course Code	Course Name	Credits
	Theory Course		
1.	14ENG104	English for Communication	4
2.	16MCA101	Introduction to Computing	4
3.	16MCA102	Programming in C	4
4.	16MCA103	Computer Organization	4
5.	16MCA104	Database Management Systems	4
6.	16MCA105	Data Structures through C++	4
7.	16MCA106	Operating Systems	4
8.	14MAT105	Discrete Mathematics	4
9.	16MCA107	Data Structures through Python	4
10.	16MCA108	Computer Networks	4
11.	16MCA109	Software Engineering	4
12.	16MCA110	Design & Analysis of Algorithms	4
13.	16MCA111	JAVA Programming	4
14.	16MCA112	Data Warehousing and Data Mining	4
15.	16MCA113	Cryptography & Network Security	4
		Total	**60**
	Practical Courses		
1	14ENG201	Communication Skills	2
2	16MCA201	Office Automation	2
3	16MCA202	Programming in C	2
4	16MCA204	Database Management Systems	2
5	16MCA205	Data Structures through C++	2
6	16MCA206	Unix and Shell Programming	2
7	16MCA207	Data Structures through Python	2
8	16MCA208	Computer Networks	2
9	16MCA209	SE & UML	2
10	16MCA210	Design and Analysis of Algorithms	2
11	16MCA211	JAVA Programming	2
12	16MCA212	Data Warehousing and Data Mining	2
13	16MCA213	Cryptography & Network Security	2
14		Discipline Elective – I (16MCA214-16MCA218)	2
15		Discipline Elective – II (16MCA219-16MCA223)	2
16		Discipline Elective – III (16MCA224-16MCA228)	2
17	16MCA291	Software Lab-1(16MCA291-16MCA298)	2
18	16MCA292	Software Lab-II(16MCA291-16MCA298)	2
19	16MCA293	Software Lab-III(16MCA291-16MCA298)	2
20	16MCA294	Software Lab-IV(16MCA291-16MCA298)	2
		Total	**40**

Internship & Seminar

1	16MCA601	Internship / Mini Project – 1	2
2	16MCA602	Major Project – 2	18
3	16MCA501	Seminar	2
		Total	**22**

List of Discipline Electives

Discipline Elective-1

Sl. No.	Course Code	Course Name	Credit
1.	16MCA401	Web Programming through PHP	04
2.	16MCA402	Web Technologies through JAVA	
3.	16MCA403	User Interface Design	
4.	16MCA404	Multimedia Systems	
5.	16MCA405	Advanced Data Structures	
		Total	**04**

Discipline Elective Practical-1

Sl. No.	Course Code	Course Name	Credit
1.	16MCA214	Web Programming through PHP Practical	02
2.	16MCA215	Web Technologies through JAVA Practical	
3.	16MCA216	User Interface Design Practical	
4.	16MCA217	Multimedia Systems Practical	
5.	16MCA218	Advanced Data Structures Practical	
		Total	**02**

Discipline Elective-II

Sl. No.	Course Code	Course Name	Credit
1.	16MCA406	Software Testing	04
2.	16MCA407	Network Security Essentials & Standards	
3.	16MCA408	Computer Graphics	
4.	16MCA409	Parallel Computing	
5.	16MCA410	Pattern Recognition	
		Total	**04**

Discipline Elective Practical-II

Sl. No.	Course Code	Course Name	Credit
1.	16MCA219	Software Testing Practical	02
2.	16MCA220	Network Security Essentials & Standards Practical	
3.	16MCA221	Computer Graphics Practical	
4.	16MCA222	Parallel Computing Practical	
5.	16MCA223	Pattern Recognition Practical	
		Total	**02**

Discipline Elective-III

Sl. No.	Course Code	Course Name	Credit
1.	16MCA411	Big Data & Analytics	04
2.	16MCA412	Wireless Sensor Networks	
3.	16MCA413	Mobile Application Development using Android	
4.	16MCA414	Knowledge Based Systems and Design	
5.	16MCA415	Semantic Web Architecture	
		Total	**04**

Discipline Elective Practical-III

Sl. No.	Course Code	Course Name	Credit
1.	16MCA224	Big Data & Analytics Practical	02
2.	16MCA225	Wireless Sensor Networks Practical	
3.	16MCA226	Mobile Application Development using Android Practical	
4.	16MCA227	Knowledge Based Systems and Design Practical	
5.	16MCA228	Semantic Web Architecture Practical	
		Total	**02**

Discipline Elective-IV

Sl. No.	Course Code	Course Name	Credit
1.	16MCA416	Distributed Operating Systems	04
2.	16MCA417	Service Oriented Architecture	
3.	16MCA418	Cloud Infrastructure & Services	
4.	16MCA419	Internet of Things	
5.	16MCA420	Middleware Technologies	
		Total	**04**

Software Labs I / II / III / IV

Sl.No.	Course Code	Course Name	Credit
1	16MCA289	Machine Learning Practical	02
2	16MCA290	Advanced Python Practical	
3	16MCA291	Selenium Testing Tool	
4	16MCA292	IBM Cognos	
5	16MCA293	Monkey Runner	
6	16MCA294	Quick Test Professional	
7	16MCA295	CloudSim	
8	16MCA296	Data Stage	
9	16MCA297	Android Studio	
10	16MCA298	R - Tool Practical	
11	16MCA299	Multimedia Practical	
		Total	02

List of Open Electives

Open Elective-1

Sl. No.	Course Code	Course Name	Credit
1.	16MBA101	Management & Organizational Behaviour	04
2.	16MBA102	Managerial Economics	
3.	16MBA103	Business Environment	
4.	16HUM403	Financial Accounting for Managers	
5.	16MBA108	Management Information Systems	
		Total	**04**

Open Elective-II

Sl. No.	Course Code	Course Name	Credit
1.	16MBA114	Operations Research	04
2.	16MBA107	Financial Management	
3.	16MBA109	Marketing Management	
4.	16MBA111	Human Resource Management	
5.	16MBA113	Statistical Methods for Manager with SPSS	
		Total	**04**

Open Elective-III

Sl. No.	Course Code	Course Name	Credit
1.	16MBA115	Business Analytics	04
2.	16MBA112	Business Research Methods	
3.	16MBA403	Financial Institutions Markets & Services	
4.	16MBA421	International Human Resource Management	
5.	16MBA439	E – Business	
		Total	**04**

Open Elective-IV

Sl. No.	Course Code	Course Name	Credit
1.	16MBA116	Strategic Management	04
2.	16MBA118	Entrepreneurship Development &Project Management	
3.	16MBA411	Financial Engineering	
4.	16MBA423	Organizational Change & Development	
5.	16MBA442	Management of Software Projects	
		Total	**04**

List of Audit Courses
(No Credits & End Exam – Only Internal Evaluation)
Refer UG Regulations Clause: 6.4

Audit Course – I				
Sl. No.	**Course Code**	**Course Name**	**Offered by the Dept. of**	**Prerequisite Course Code / None**
1.	14ENG301	Effective Public Speaking	English	None
2.	14ENG302	Creative Writing	English	None
3.	14HUM301	Entrepreneurship Development	Humanities	None
4.	14HUM302	Introduction to Intellectual Property Rights	Humanities	None
5.	14CSE301	Data Analysis Using R	CSE	None

Audit Course – II				
Sl. No.	**Course Code**	**Course Name**	**Offered by the Dept. of**	**Prerequisite Course Code / None**
1.	14ENG303	Phonetics and Spoken English	English	None
2.	14ENG304	Introductory Psychology	English	None
3.	14CSE302	Ethical Hacking	CSE	None
4.	16MBA301	Business Ethics and Corporate Governance	Management Studies	None
5.	14HUM303	National Service Scheme (NSS)	Humanities	None

Programme Core Courses

MCA I Year - I SEM

14ENG104 ENGLISH FOR COMMUNICATION

L	T	C
4	0	4

Course Objectives:

1. The Language Lab focuses on recognizing and production practice of sounds of language and familiarizes the students with the use of English in everyday situations and contexts.
2. To expose the students to a variety of self-instructional, learner-friendly modes of language learning.
3. To help the students cultivate the habit of reading passages from the computer monitor, thus equip them with the required facility to face computer-based competitive exams such GRE, TOEFL, GMAT etc.
4. To enable them learn better pronunciation through stress on word accent, intonation, and rhythm.
5. To train them in using language effectively to face interviews, group discussions, public speaking.
6. To expose them to different techniques in resume preparation, report writing, format-making etc.

UNIT-I

VOCABULARY BUILDING-Synonyms and Antonyms, Word roots, One-word substitutes, Prefixes and Suffixes, Idioms and phrases – Grammar- use of articles, prepositions, tenses & Subject-Verb agreement – Common Errors

UNIT-II

NON-VERBAL COMMUNICATION (BODY LANGUAGE - KINESICS, PROXEMICS ETC) - Verbal Communication - Informal Vs Formal conversation – Features of and barriers to effective communication – Process and flow of communication

UNIT-III

COMMUNICATION SKILLS – Listening (barriers to & features of effective listening), Speaking (Jargons - rate of speech, pitch, tone - Clarity of voice etc.), Reading (techniques) and Writing (features of effective writing such "as clarity, brevity, appropriate tone, balance etc) - differences between spoken and written communication

UNIT-IV

TECHNICAL PRESENTATIONS - types of presentation –video conferencing-- participation in meetings - chairing sessions.

INTERVIEWS – planning - interviewing in different settings and for different purposes, interview through tele and video conferencing, recruiting.

UNIT-V

LETTER-WRITING - business letters - format - style – effectiveness, promptness- e-mail, fax –

TECHNICAL REPORT WRITING – Types of reports - progress reports, routine reports - Annual reports - formats - Analysis of sample reports from industry - Synopsis and thesis writing

Course Outcomes:

Students will get the required training in documentation, presentation, discussions, facing interviews and develop communicative competence.

References:

1. Technical Communication by Meenakshi Raman and Sangeetasharma, Oxford Univ.Press.
2. Effective Technical Communication, M Ashraf Rizvi, Tata Mc.Graw-Hill Pub,company Ltd.
3. Communication Skills,Sanjay Kumar &PushpLata, Oxford Univ.Press.
4. Basic Communication Skills for Technology, Andrea J. Rutherford: Pearson Education Asia, New Delhi.
5. GRE and TOEFL; Kaplan and Baron's English in Mind, Herbert Puchta and Jeff Stranks, Cambridge
6. Communication Skills, LenneSen, Prentice –Hall of India Pvt. Ltd., New Delhi.
7. Communicating at work, Ronald B. Adler, Seanne Marquardt Elmhurst ,McGraw Hill international editions.

MCA I Year - I SEM

16MCA101 INTRODUCTION TO COMPUTING

L	T	C
4	0	4

Course Prerequisite: None

Course Description:

This course will discuss the basics of computer concepts in hardware, software programming, networking, database concepts, and other latest applications. Students were exposed to social, legal, ethical issues and privacy. Widely used applications including word processing, spreadsheets, presentation.s

Course Objectives:

1. To impart the knowledge for a student about the evolution of computers, its input and output devices and its architecture.
2. To elucidate the student about the memory organization and processor description.
3. To teach students about the operating systems, its evolution and its different types.
4. To develop the student to manage the data using DBMS.
5. To illuminate the student about the concepts of the networks, crimes and laws associated with it.

UNIT- I

COMPUTER GENERATION & CLASSIFICATIONS: First Generation of Computers, The Second Generation, The Third Generation, The Fourth Generation, The Fifth Generation, Moore's Law, Classification of computers, Distributed Computer System, parallel computers.

DATA REPRESENTATION: Representation of Characters in computers, Representation of Integers, Representation of Fractions, Hexadecimal Representation of Numbers, Decimal to Binary Conversion, Error-detecting codes.

INPUT & OUTPUT DEVICES: Description of Computer Input Units, Other Input Methods, Computer Output Units (Printers, Plotters)

COMPUTER BASICS: Algorithms, A Simple Model of a Computer, Characteristics of Computers, Problem-solving Using Computers.

UNIT -II

COMPUTER MEMORY: Memory Cell, Memory Organization, Read Only Memory, Serial Access Memory, Physical Devices Used to Construct Memories, Magnetic Hard Disk, floppy Disk Drives, Compact Disk Read Only Memory, Magnetic Tape Drives.

PROCESSOR: Structure of Instructions, Description of a Processor, Machine Language and Instruction set. Processors used in desktops and laptops. Specification of a desktop and Laptop computer currently available in the market (Specifications of processor, motherboard &chipset, memory, interface & capacity of hard disk & DVD drives, I/O ports)

UNIT -III

COMPUTER ARCHITECTURE : Interconnection of Units, Processor to Memory communication, I/O to Processor Communication, Interrupt Structures, Multiprogramming, Processor Features, Reduced Instruction, Set Computers (RISC), Virtual Memory.

SOFTWARE CONCEPTS: Types of Software, Programming Languages, Software (Its Nature & Qualities), Programming Languages.

UNIT -IV

OPERATING SYSTEMS: History and Evolution. Main functions of OS Multitasking, Multiprocessing, Time Sharing, Real Time OS with Examples,

DATABASE MANAGEMENT SYSTEM: Purpose and Organization of Database, Introduction to Data Models.

UNIT -V

COMPUTERS & COMMUNICATIONS: Introduction to Computer Communications, Introduction to Computer Networks, Types of Networks, OSI/TCP Model, LAN technologies (fast Ethernet &Gigabit Ethernet), How LAN works, Brief survey of active and passive LAN components.

INTERNET: Network, Client and Servers, Host & Terminals, TCP /IP, World Wide Web, Hypertext, Uniform Resource Locator, Web Browsers, IP Address, Domain Name, Internet Services Providers, Internet Security, Internet Requirements, Web Search Engine, Net Surfing, Internet Services, Case Study, Intranet.

CYBER LAWS: Introduction to Cyber Laws, Cyber-crime, Cyber contract, Cyber privacy, IT Act

Course Outcomes:

1. The student will acquire knowledge about the system in detail.
2. The student will be able to store, retrieve and manage the data.
3. Student will be aware of the crimes and laws related to networks.

Text Books:

1. P .K. Sinha, Fundamentals of Computers, BPB Publications
2. V. Rajaraman, Fundamentals of Computers, 3rdEdition, PHI Publications

MCA I Year - I SEM

16MCA102 PROGRAMMING IN C

L	T	C
4	0	4

Course Prerequisite:

General knowledge about computer hardware and software. Basic knowledge of decision, and repetition structures in C language.

Course Description:

Learn the C programming language and its fundamental programming concepts. Gain the knowledge to write simple C language applications and undertake future courses that assume some background in computer programming. Topics include variables, data types, functions, control structures, pointers, strings, arrays and dynamic allocation principles. You need access to any programming environment that allows you to write, edit, compile, link and debug a C program.

Course Objectives:

1. To understand the various steps in Program development.
2. To understand the basic concepts in C Programming Language.
3. To learn how to write modular and readable C Programs
4. To learn to write programs (using structured programming approach) in C to solve problems.

UNIT- I

Flow charts, Tracing flow charts, Problem solving methods, Need for computer Languages,Sample Programs written in C. C character set, Identifiers and keywords, Data types, Declarations, Expressions, statements and symbolic constants.

UNIT- II

getchar, putchar, scanf, printf, gets, puts, functions, #include, #define, #ifdef, Arithmetic, unary, logical, bit-wise, assignment and conditional operators.

UNIT –III

While, do-while, for statements, nested loops, if else, switch, break, Continue, and goto statements, comma operator.

UNIT- IV

Automatic, external, register and static variables.Defining and accessing, passing arguments, Function prototypes, Recursion, Library functions, Static functions, Defining and processing, Passing arrays to a function, Multi-dimensional arrays. Defining and operations on strings. Declarations, passing pointers to a function, Operations on pointers, Pointer Arithmetic, Pointers and arrays, Arrays of pointers function pointers.

UNIT –V

Defining and processing, Passing to a function, Unions, typedef, array of structure, and pointer to structure File operation: creation, copy, delete, update, text file, binary file.

CourseOutcomes:

A student who successfully completes the course will have the ability to:

1. understand the basic terminology used in computer programming
2. write, compile and debug programs in C language.
3. use different data types in a computer program.
4. design programs involving decision structures, loops and functions.
5. explain the difference between call by value and call by reference
6. understandthe dynamics of memory by the use of pointers.
7. Usedifferent data structures and create/update basic data files.

Text Books:

1. A Structured Programming Approach Using C, B.A.Forouzan and R.F. Gilberg, Third Edition,Cengage Learning.
2. Programming in C. P. Dey and M Ghosh, Oxford University Press.

Reference Books:

1. C& Data structures – P. Padmanabham, Third Edition, B.S. Publications.
2. Problem Solving and Program Design in C, J.R. Hanly and E.B. Koffman, 7th Edition, Pearsoneducation.
3. Programming in C,Ajay Mittal, Pearson.
4. Programming with C,B.Gottfried, 3rd edition, Schaum's outlines, TMH.
5. Problem solving with C,M.T.Somasekhara, PHI
6. Programming with C,R.S.Bickar, Universities Press.
7. Computer Programming & Data Structures, E.Balagurusamy, 4th edition, TMH.
8. Programming in C – Stephen G. Kochan, III Edition, Pearson Education.
9. The C Programming Language, B.W. Kernighan and Dennis M.Ritchie, PHI.
10. C Programming with problem solving, J.A. Jones & K. Harrow, Dreamtech

MCA I Year - I SEM

16MCA103 COMPUTER ORGANIZATION

L	T	C
4	0	4

Course Objectives:

1. The main objective of computer organization course is to introduce the main concepts and components of computer organization and architecture.
2. Understand the architecture of a modern computer with its various processing units.
3. Understand the performance measurement of the computer system
4. Understand the Cache memory and its importance.
5. Students can calculate the effective address of an operand by addressing modes

UNIT- I

INTRODUCTION: Overview of basic digital building blocks; basic structure of a digital computer. Number system and representation of information, arithmetic and logical operation, hardware implementation, Real numbers - fixed and floating point, IEEE754 representation. logic gates, Sequential and combinational circuits, flip flop, adders, multiplexers, encoders, decoders

UNIT- II

BASIC BUILDING BLOCKS FOR THE ALU: Adder, Subtractor, Shifter, Multiplication and division circuits. CPU: Data path - ALU, Registers, CPU buses; Control path - microprogramming (only the idea), hardwired logic, RISC, CISC;

UNIT- III

Instruction format, Bus, Common Bus, hand shaking, external interface. Various addressing modes. Concept of sub-routine and sub-routine call. Use of stack for handling sub-routine call and return, instruction interpretation and execution. Bus Architecture

UNIT- IV

MEMORY : Memory organization; concepts of semi-conductor memory, CPU memory interaction, organization of memory modules, cache memory and related mapping and replacement policies, virtual memory

UNIT- V

I/O Sub block: I/O techniques - interrupts, polling, DMA; Synchronous vs. Asynchronous I/O; Controllers. – Interface circuits – Standard I/O Interfaces (PCI, SCSI, USB), I/O devices and processors. Introduction to VHDL concepts: examples to be taken up from the rest of the course for implementation.

Course Outcomes:

1. Successful completion of this course should lead to the following learning outcomes:
2. Understand the combinational and sequential logical circuits.
3. Understand the basics of assembly language.
4. Understand the main concepts of computer architecture.
5. Design and analyze the main functional units of a computer.
6. Implement assembly programs that accomplish basic computational and I/O operations.
7. Discuss and work in a group in order to design the main functional units of a computer.
8. Discuss and work in a group in order to design and implement assigned programming tasks in Assembly.
9. Demonstrate designed functional units as well as implemented assembly programs.

Text Book:

Computer Organization, Hamachar, Vranesic&Zaky.

MCA I Year - I SEM

14ENG201 COMMUNICATION SKILLS PRACTICAL

L	P	C
0	3	2

Course Objectives:

1. The Language Lab focuses on recognizing and production practice of sounds of language and familiarizes the students with the use of English in everyday situations and contexts.
2. To expose the students to a variety of self-instructional, learner-friendly modes of language learning.
3. To help the students cultivate the habit of reading passages from the computer monitor, thus equip them with the required facility to face computer-based competitive exams such GRE, TOEFL, GMAT etc.
4. To enable them learn better pronunciation through stress on word accent, intonation, and rhythm.
5. To train them in using language effectively to face interviews, group discussions, public speaking.
6. To expose them to different techniques in resume preparation, report writing, format-making etc.

Exercise 1: Phonetics – English pronunciation– basics in phonetics- introduction to sounds of English – vowels – diphthongs – consonants

Exercise 2: Techniques to develop effective word accent- various stress patterns– developing voice, quality and tone– intonation– rhythm

Exercise 3: Fundamentals of interpersonal communication– starting a conversation- responding appropriately and relevantly - Dialogues- Formal and informal– using the right body language

Exercise 4: Role play in different situations.

Exercise 5: Importance of Oral Presentations- developing and organizing the presentations– verbal and visual support in presentations– delivering the presentation

Exercise 6: Informative, group and special occasion presentations– persuasive presentations

Exercise 7: Formal and Informal debate– theory for debating– art of debating

Exercise 8: Debate on various topics

Exercise 9: Nature of group discussion– characteristics of successful GD‟s– strategies– techniques for individual contribution- intervention, summarizing, modulation of voice, body language, relevance, fluency and coherence.

Exercise 10: Organizing Group Discussions

Exercise 11: Interview Skills– concept and process, pre-interview planning, opening strategies, answering strategies, projecting a positive image, interview through tele and video-conferencing.

Exercise 12: Organizing mock interviews

Exercise 13: Resume design– structure and presentation, planning, defining the career objective, projecting one's strengths and skill-sets, summary.

Exercise 14: Resume styles– job application letters

Minimum Requirements Computer aided multimedia language lab equipped with Computer systems in LAN facility.

Conventional Language Lab. with audio and video systems, speakers, headphones and a teacher console so as to accommodate at least 60 students.

Prescribed Software: Young India - Clarity SNET

Suggested Software:

1. Cambridge Advanced Learners' Dictionary with exercises
2. The Rosetta Stone English Library
3. Clarity Pronunciation Power
4. Mastering English in Vocabulary

Course Outcomes:

1. Students can notice different accents of English and improve their fluency and common skills through various activities.
2. Students are able to participate in every co-curricular activity and are able to face TOEFL, GRE and IELTS confidently and appear for any competitive exams.
3. Students are able to participate in Group Discussions and face interviews.

The following course content is prescribed for the English Language Laboratory Practice

1. Phonetics- Introduction to the Sounds of English – vowels, Diphthongs and consonants
2. Introduction to Stress, Accent, Intonation and Rhythm
3. Interpersonal communications - Role play
4. Oral Presentations/Public speaking
5. Debate
6. Group Discussions
7. Facing interviews
8. Resume preparation

Reference Books:

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. **A Course in Communication Skills**, Kiranmai, Dutt& Co Foundation Books, 2012.
4. **Current English grammar and usage,** S M Guptha, PHI, 2013.
5. **Powerful Vocabulary Builder,**Anjana Agarwal, New Age International Publishers, 2011.

Speak Well, JayashreeMohanraj et al, Orient Blackswan.

MCA I Year - I SEM

16MCA201 OFFICE AUTOMATION PRACTICAL

L	P	C
0	3	2

Course Prerequisite: None

Course Description:
This course gives hands on experience to learn widely used applications including word processing, spreadsheets, databases, presentation.

Course Objectives:

1. To provide the student on hands-on experience and technical training in Word processors, Spreadsheets, Presentations.
2. To provide the knowledge about the Database.

List of Experiments:

1. Create a document which includes formatting and organizing the data, including images.
2. Create a time table depicting the properties of table.
3. Create documents with hyperlinks and bookmarks.
4. Create a mail merge.
5. Create, Record and implement a Macro.
6. Create the slides which includes-Selecting the style for slides, formatting the slides with different fonts, colors, creating charts and tables, inserting and deleting text, graphics, bulleting and numbering.
7. Create slides that include hyperlinks, animations, running the slideshow and setting the timing for slideshow.
8. Create a Spread sheet applying formulas and functions for generating a marks memo of a student.
9. Apply sort and filters for the above generated data.
10. Create a Spread sheet applying formulas and functions for generating a salary slip for an employee.
11. Preparecharts for above generated data.
12. Create the database and tables.
13. Perform basic SQL commands which include Create, Insert, Delete, Drop, View and updating the tables.
14. Procedures for installation of different Operating Systems & application S/Ws.

Course Outcomes:
Students are expected to demonstrate:

1. Skills to prepare the Documents effectively using word processors, spread sheets, presentation.
2. To work with database and basic SQL commands.

Mode of Evaluation: Practical, Written Examination

MCA - I Year - I SEM

16MCA202 PROGRAMMING IN C PRACTICAL

L	P	C
0	3	2

Course Objectives:

1. To write programs in C to solve the problems.
2. To implement linear data structures such as lists, stacks, queues.
3. To implement simple searching and sorting methods.

Exercise 1:

1. Write a C program to find the sum of individual digits of a positive integer.
2. A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1.Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program togenerate the first n terms of the sequence.
3. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied bythe user.

Exercise 2:

1. Write a C program to calculate the following Sum:
   Sum=1-x2/2! +x4/4!-x6/6!+x8/8!-x10/10!
2. Write a C program to find the roots of a quadratic equation.

Exercise 3:

1. The total distance travelled by vehicle in‘t’ seconds is given by distance = ut+1/2at2 where ‘u’ and‘a’ are the initial velocity (m/sec.) and acceleration (m/sec2). Write C program to find the distance travelledat regular intervals of time given the values of ‘u’ and ‘a’. The program should provide the flexibility to theuser to select his own time intervals and repeat the calculations for different values of ‘u’ and ‘a’.
2. Write a C program, which takes two integer operands and one operator from the user, performs theoperation and then prints the result. (Consider the operators +,-,*, /, % and use Switch Statement)

Exercise 4:

Write C programs that use 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.

Exercise 5:

1. Write a C program to find the largest integer in a list of integers.
2. Write a C program that uses functions to perform the following:
   i. Addition of Two Matrices
   ii. Multiplication of Two Matrices

Exercise 6:

1. 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.
2. Write a C program to determine if the given string is a palindrome or not

Exercise 7:

1. Write a C program that displays the position or index in the string S where the string T begins, or – 1 ifS doesn't contain T.
2. Write a C program to count the lines, words and characters in a given text.

Exercise 8:

1. Write a C program to generate Pascal's triangle.
2. Write a C program to construct a pyramid of numbers.

Exercise 9:

Write a C program to read in two numbers, x and n, and then compute the sum of this geometricprogression:
$1+x+x^2+x^3+\ldots\ldots\ldots\ldots +x^n$
For example: if n is 3 and x is 5, then the program computes 1+5+25+125.
Print x, n, the sumPerform error checking. For example, the formula does not make sense for negative exponents – if n isless than 0. Have your program print an error message if n<0, then go back and read in the next pair ofnumbers of without computing the sum. Are any values of x also illegal? If so, test for them too.

Exercise 10:

1. 2's complement of a number is obtained by scanning it from right to left and complementing all the bitsafter the first appearance of a 1. Thus 2's complement of 11100 is 00100. Write a C program to find the2's complement of a binary number.
2. Write a C program to convert a Roman numeral to its decimal equivalent.

Exercise 11:
Write a C program that uses functions to perform the following operations:
i. Reading a complex number
ii. Writing a complex number
iii. Addition of two complex numbers
iv. Multiplication of two complex numbers
(Note: represent complex number using a structure.)
Exercise 12:

1. Write a C program which copies one file to another.
2. 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.)

Exercise 13:

1. Write a C program to display the contents of a file.
2. Write a C program 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)

Course Outcomes:
Students are expected

1. To apply the concepts of structure oriented programming approaches
2. To deal the linear and non-linear data structure related problems
3. To solve the problems related to logical operations
4. To solve the various mathematical operations
5. To use the concepts of file handling

Text Books:

1. C programming and Data Structures, P. Padmanabham, Third Edition, BS Publications
2. Computer Programming in C, V. Rajaraman, PHI.
3. C Programming, E.Balagurusamy, 3rd edition, TMH.
4. C Programming, M.V.S.S.N.Prasad, ACME Learning Pvt. Ltd.
5. Mastering C, K.R. Venugopal and S.R. Prasad,

MCA I Year – II Semester

16MCA104 DATABASE MANAGEMENT SYSTEMS

L	T	C
4	0	4

Course Prerequisite: None

Course Description:

This course is designed to investigate how database management system techniques are used to design, develop, implement and maintain database applications in organizations.

Course Objectives:

The students will be able to:

1. Understand data base concepts and applications, data base system structure and data modeling.
2. Introduces relational model, integrity constraints, keys, implementation of relational query languages and Relational Algebra.
3. Provide an over view of schema refinement and Normal Forms.
4. To understand the concept of transaction and its management, concurrency control schemes and crash recovery technique.
5. To construct simple and moderately advanced database queries using Structured Query Language (SQL).

UNIT - I

INTRODUCTION

Database system Applications, Database system Vs File system, View of data, Data abstraction, Instances and Schemas, Database users, Database system structure, Database design and ER diagrams, Beyond - ER Design Entities, Attributes, Entity sets, Relationships and Relationship sets, Additional features of ER model.

UNIT - II

THE RELATIONAL MODEL & RELATIONAL ALGEBRA

Introduction to Relational model- Integrity constraints over the relations, Enforcing integrity constraints, Database Languages, DDL, DML, TCL, basic form of SQL query, Querying relational data, Logical database design, views, Destroying and altering tables/views. Nested queries correlated nested queries, Null values, Relation Algebra- selection, projection, renaming, join, examples.

UNIT - III

SCHEMA REFINEMENT AND NORMAL FORMS

Schema refinement – Problems Caused by redundancy – Decompositions – Problem related to decomposition – reasoning about FDS – FIRST, SECOND, THIRD Normal forms – BCNF -

Lossless-join Decomposition, Dependency- preserving Decomposition – Schema refinement in Data base Design – Multi valued Dependencies – Fourth Normal Form and Fifth Normal form.

UNIT -IV

TRANSACTION MANAGEMENT, CONCURRENCY CONTROL AND CRASH RECOVERY

Overview of Transaction Management: ACID properties, Transactions and Schedules, Concurrent Execution of transactions, Lock Based Concurrency Control, Performance Locking, Transaction Support in SQL, Serializability and Recoverability, Introduction to Lock Management, Lock Conversions, Dealing with Dead Locks, Concurrency without Locking. Crash recovery – Aries Recovery Algorithm.

UNIT - V

SQL

Structured Query Language, data types, logic operators, aggregate functions, character functions, triggers, cursors, exceptional handling, PL/SQL,

Course Outcomes:

Upon successful completion of this course, students should be able to:

1. Apply the basic concepts and advantages of database systems to an organization.
2. Demonstrate the relational data model, implementation of various SQL queries and relational algebra operations.
3. Deal with the different issues involved in the database design using Normal Forms.
4. Explain the basic issues of transaction processing and concurrency control.
5. Execute a relational database system using SQL.

Text Books:

1. Data Base Management Systems, Raghurama Krishnan, Johannes Gehrke, TMH
2. Data Base System Concepts, 6/e, Silberschatz, Korth, TMH.

References:

1. Data Base Management System, 5/e, ElmasriNavathe, Pearson
2. An Introduction to Database Systems, 8th Edition, C J Date, Pearson

Mode of Evaluation: Assignment, Written Examination.

MCA I Year – II SEM

16MCA105 DATA STRUCTURES THROUGH C++

L	T	C
4	0	4

Course Prerequisite: Concepts of Programming Logic and Techniques.

Course Description:
This course gives the concepts of C++. It concentrates on the Linear and Non-Linear data structures and their applications. It also explores various searching and sorting algorithms.

Course Objectives:

Student will be able to:

1. Acquire knowledge on basic constructs of C++.
2. Get familiar with reusability, generic programming of C++ and Data Structures basics.
3. Understand various Linear data Structures.
4. Understand more on linear data structures, hashing and various collision resolution techniques.
5. Understand Non-Linear Data structures and Sorting & Searching algorithms.

UNIT – I

INTRODUCTION TO C++
Class Definition, Objects, Class Members, Access Control, Class Scope, Constructors and destructors, parameter passing methods, Inline functions, static class members, this pointer, friend functions, dynamic memory allocation and deallocation (new and delete), exception handling, Function Over Loading, Operator Overloading.

UNIT –II

INHERITANCE, GENERIC PROGRAMMING AND INTRODUCTION TO DATA STRUCTURES
Inheritance basics, base and derived classes, inheritance types, base class access control, Function and class templates, runtime polymorphism using virtual functions, abstract classes, streams I/O.
Introduction to Data Structures: Information and its meaning: Abstract Data Types, Sequences as Value Definitions, ADT for varying length character Strings, Data Types, Pointers and review of Pointers, Data Structures
Arrays: the Array as an ADT, Using One-dimensional Arrays, Implementing One-Dimensional Arrays, and Arrays as Parameters, Handling of Character Strings and Character Strings

UNIT-III

LINEAR DATA STRUCTURES

Definition and examples, Primitive operations, Example, The stack as an ADT, Representing Stacks, Implementing the pop operation, Testing for exceptional conditions , Implementing the push operations , Examples for infix , postfix, and prefix expressions, Applications of Stacks: Regular Expressions, Expression Evaluations, Recursion etc.

Queues and Lists The queue and its sequential representation, the queue as ADT, Insert operation, Priority queue, and Array implementation of a priority queue.

Linked lists: Inserting and removing nodes from a list, Linked implementations of stacks, get node and Free node operations, Linked implementation of queues, Linked list as a data Structure, Example of list operations, Header nodes, Array implementation of lists, Limitations of array implementation, allocating and freeing dynamic variables, Linked lists using dynamic variables, Non integer and non-homogenous lists,

UNIT-IV

CIRCULAR, DOUBLE LINKED LISTS AND HASHING

Circular lists, Stack as a circular lists, doubly linked lists, Application of Linked Lists, skip list representation, operations insertion, deletion and searching, hash table representation, hash functions, collision resolution-separate chaining, open addressing-linear probing, quadratic probing, double hashing, rehashing, extendible hashing, comparison of hashing and skip lists.

UNIT – V

BINARY SEARCH TREES, SORTING & SEARCHING ALGORITHMS

Tree traversals, Binary Search Tree and Operations, AVL Tree and Operations, Red-Black Tree, Threaded binary trees and operations, B-Trees, B-Tree of order m, height of a B-Tree, insertion, deletion and searching, Comparison of Search Trees.

Sorting &Searching Bubble sort, Quick sort, Selection sort, Tree Sorting: Binary Tree Sort, Heap Sort, and Insertion Sorts: Simple Insertion, Merge and Radix Sort. Basic Search Techniques: Algorithmic Notations, Sequential searching, Searching an ordered table, Indexed sequential search, Binary search, Interpolation search.

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. Understand the Object Oriented principles and applications through C++.
2. Deliberate code reusability, generic programming.
3. Apply Linear Data Structures concepts in computing problems.
4. Learn more on linear data structures & hashing and various collisions resolution techniques.
5. Code for sorting and searching algorithms.

Text Books:

1. Data structures, Algorithms and Applications in C++, S.Sahni, University Press (India) Pvt. Ltd, 2nd edition, Universities Press Orient Longman Pvt. Ltd.
2. 2. Data structures and Algorithms in C++, Michael T. Goodrich, R.Tamassia and .Mount, Wiley student edition, John Wiley and Sons.
3. Data Structures Using C and C++by Yedidyah Langham and Moshe J. Augenstein and
4. Aaron M Tenanbanum, 2nd Edition, Pearson Education Asia, 2002.

References:

1. Data structures and Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education. Ltd., Second Edition.
2. Data structures and algorithms in C++, 3rd Edition, Adam Drozdek, Thomson.
3. Data structures using C and C++, Langsam,Augenstein and Tanenbaum, PHI.
4. Problem solving with C++, The OOP, Fourth edition, W.Savitch, Pearson education.

Mode of Evaluation: Assignment, Seminar, Written Examination.

MCA I Year – II SEM

16MCA106 OPERATING SYSTEMS

L	T	C
4	0	4

Course Prerequisite: 14MCA103

Course Description:
This course introduces the basic concepts of operating systems and their structures. An in-depth working concepts of Memory and Process Management are explained. It also covers an introduction to Unix shell scripting and BASH commands.

Course Objectives:

To study and apply concepts relating to operating systems:

1. Understand the various operating systems, design, services, structure and their organization.
2. Understand the Unix environment and shell scripting.
3. Understand the BOURNE AGAIN SHELL(BASH)
4. Understand Processes, Scheduling and their Synchronization without Dead lock.
5. Recognize the working procedure of Main Memory and Virtual Memory.

UNIT – I

INTRODUCTION
Introduction: Operating Systems objectives and functions, OS Structure, OS Operations, Evolution of Operating Systems - Simple Batch, Multi programmed, time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, Special -Purpose Systems
OS Structure: Operating System services, User OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, OS Structure, Virtual Machines.

UNIT – II

INTRODUCTION TO UNIX SHELL
UNIX Features and Environment, UNIX Structure, Accessing UNIX, **LINUX UTILITIES**-File handling utilities, Security by file permissions, Process utilities, Disk utilities, networking commands. Grep command, Vi Editor **SED** – scripts, operation, addresses, commands, applications.awk – execution, fields and records, scripts, operation, patterns, actions, functions, uses system commands in awk.

UNIT- III

WORKING WITH THE BOURNE AGAIN SHELL (BASH)
Introduction, shell responsibilities, pipes and input Redirection, output redirection, command Execution, command line editing, quotes, command substitution, shell variables, Filters, shell meta

characters, shell programming-control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts.

UNIT – IV

PROCESS

Process: process concept, process scheduling, operations on process, Inter process communication, examples of IPC systems, communication in client server systems.

CPU Scheduling: Basic concepts, scheduling algorithms, multiple process scheduling, thread scheduling,

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

UNIT – V

MEMORY MANAGEMENT

Main 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, memory mapped files, allocating kernel memory.

Course Outcomes:

By the end of the course you should be able to

1. Classify the operating systems, general architecture and structures of operating systems.
2. Recognize various levels of user accesses and to control the environment through shell scripting.
3. Implement functions, quoting mechanisms, control structures, substitutions and interrupt processing using BASH.
4. Schedule the processes without dead locks with inter process synchronization.
5. Explain memory management and kernel management.

Text Books:

1. Operating System Principles , Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 8th Edition, Wiley Student Edition
2. Unix and Shell programming, B.A.Forouzan and R.F.Gilberg, Cengage Learning

References:

1. Modern Operating Systems, Andrew S Tanenbaum, 3rd Edition, Pearson/PHI
2. Operating Systems – Internals and Design Principles, W. Stallings, 6th Edition, Pearson Education.
3. Beginning Linux Programming, 4th Edition, N.Matthew, R.Stones,Wrox, Wiley India Edition, rp-2008.

Mode of Evaluation: Assignment, Seminar, Written Examination.

MCA I Year – II SEM

14MAT105 DISCRETE MATHEMATICS

L	T	C
4	0	4

Course Prerequisite: None

Course Description:
The purpose of this course is to understand and use (abstract) discrete structures that are backbones of computer applications. In particular, this class is meant to introduce logic, proofs, sets, relations, functions, counting, and probability, with an emphasis on applications in computer.

Course Objectives:
1. On completion, the students will be able to explain and apply the basic methods of discrete (noncontiguous) mathematics in Computer Science. They will be able to use these methods in subsequent courses in the design and analysis of algorithms, computability theory, software engineering, and computer systems.
2. In particular, students will be able to
3. Reason mathematically about basic data types and structures (such as numbers, sets, graphs, and trees) used in computer algorithms and systems; distinguish rigorous definitions and conclusions from merely plausible ones; synthesize elementary proofs, especially proofs by induction.
4. Model and analyze computational processes using analytic and combinatorial methods.
5. Apply principles of discrete probability to calculate probabilities and expectations of simple random processes.
6. Work in small teams to accomplish all the objectives above.

UNIT - I

Sets, Relations, Possets, Functions-Mathematical Inductions (Simple and Strong) Principles of counting (Addition and multiplication) Logic and Proof, Sets and Functions – Logic Propositional equivalence, Predicates and Quantities, Nested quantifiers, Methods of Proof, sets, set operations, functions.

UNIT - II

The Integers and Division, Integers and Algorithms, Applications of Number theory, Mathematical reasoning, Induction and Recursion – Proof strategy, Sequences and Summations, Mathematical induction.Recursive definitions and Structural induction, Recursive algorithms, Program correctness.

UNIT - III
The basics of counting, the pigeonhole principle, Permutations and Combinations, Binomial coefficients, Generalized permutations and combinations, Generating permutations and combinations, Recurrence relations, Solving recurrence relations.

Relations – Relations and their properties, n-ary Relations and their applications, Representing Relations, Closures of relations, Equivalence relations, Partial orderings. Languages and Grammers,

Finite state machines with output, Finite state machines with no output, Language recognition, Turing machines.

UNIT - IV

Graphs – Introduction to Graphs, Graph terminology, Representing graphs and Graph isomorphism, Connectivity, Euler and Hamilton Paths, Shortest Path problems, Planar graphs, Graph coloring.

UNIT - V

Recurrence Relations and Generating Functions: Homogeneous and Non-homogeneous recurrence and their solutions-solving recurrences using generating functions

Course Outcomes:

Students will be able to:

1. Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations, functions, and integers.
2. Evaluate elementary mathematical arguments and identify fallacious reasoning (not just fallacious conclusions).
3. Synthesize induction hypotheses and simple induction proofs.
4. Prove elementary properties of modular arithmetic and explain their applications in Computer Science, for example, in cryptography and hashing algorithms.
5. Apply graph theory models of data structures and state machines to solve problems of connectivity and constraint satisfaction, for example, scheduling.
6. Derive closed-form and asymptotic expressions from series and recurrences for growth rates of processes.
7. Problem solve and study in a small team with fellow students.

Text Book:

Rosen K.H. Discrete Mathematics and its Applications, 5th edition, Tata McGraw – Hills, 2003.

Reference Books:

1. Johnson Baugh R, and Carman R, Discrete mathematics, 5th edition, Person Education, 2003.
2. Kolman B, Busoy R.C, and Ross S.C, Discrete Mathematical Structures, 5th edition, Pretitice – Hall, 2004.
3. Mott J.L, Kandel A, and Bake T.P, Discrete Mathematics for Computer Scientists & Mathematicians, 2nd edition, Prentice-Hall of India, 2002.
4. Gary Haggard, John Schlipf and sue Whitesides, Discrete

Mode of Evaluation: Assignment, Seminar, Written Examination.

MCA I Year – II SEM

16MCA204 DATABASE MANAGEMENT SYSTEMS PRACTICAL

L	P	C
0	3	2

Course Prerequisite: None

Course Description:
This course provides an executive understanding and approach to the technical subject of database management system.

Course Objectives:

1. Provide formal foundation on the relational model of data.
2. Present SQL and procedural interfaces to SQL comprehensively
3. Give an introduction to systematic database design approaches covering conceptual and physical design.

Recommended Systems/Software Requirements:
1. Intel based desktop PC
2. Mysql /Oracle latest version Recommended

LIST OF SAMPLE PROBLEMS/EXPERIMENTS

1) Creation, altering and droping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.

2) 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.

3) Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

4) 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)

5)

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.

6) Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.

7) Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE- APPLICATION ERROR.

8) Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.
9) Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.
10) Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.
11) Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
12) Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

Course Outcomes:

At the end of the course the student will be able to:

1. Populate and query a database using SQL DML/DDL commands.
2. Declare and enforce integrity constraints on a database using a state-of-the-art RDBMS.
3. Programming PL/SQL including stored procedures, stored functions, cursors and packages.

References :

1. ORACLE PL/SQL by example. Benjamin Rosenzweig, Elena Silvestrova, Pearson Education 3rd Edition
2. SQL & PL/SQL for Oracle 10g, Black Book, Dr.P.S. Deshpande.
3. Introduction to SQL, Rick F.VanderLans, Pearson Education.
4. Oracle PL/SQL Programming,StevenFeuerstein,SPD.
5. The Database Book, N.Gehani, Universities Press.
6. Database Systems using Oracle: A Simplified Guide to SQL and PL/SQL, Shah, PHI.

Mode of Evaluation: Practical, Written Examination

MCA I Year – II SEM

16MCA205 DATA STRUCTURES THROUGH C++ PRACTICAL

L	P	C
0	3	2

Course Prerequisites: Knowledge on Object Oriented Programming approach and algorithm basics.

Course Description:
This course introduces on the practical part of Data Structure under C++ Environment. This course allows students to implement linear and nonlinear data structures. It also gives practical exposure on Sorting and searching algorithms.

Course Objectives:
1. Design, implement, test, debug, and document programs in C++.
2. Program with arrays, pointers using dynamic memory allocation concept.
3. Build and manipulate linear and non-linear data structures, including stacks, queues, linked lists, trees.
4. Able to choose appropriate data structures for solving typical searching and sorting problems.

List of Experiments:

1. Create a Stack and do the following operations using arrays and linked list.
    a. push
    b. pop
    c. peep
2. Create a queue and do the following operations using arrays and linked list.
    a. add
    b. remove
    c. display queue elements.
3. Implement the operations on singly linked list.
4. Implement the operations on circular linked list.
5. Implement the operations on doubly linked list.
6. Write a c++ program that use non recursive functions to traverse tree in
    a. pre order
    b. in order
    c. postorder
7. Write a program to create BST (Binary Search tree)
8. Write a program to implement sorting methods
    a. Bubble & Selection sort.
    b. quick sort.
    c. merge sort.
9. Write C++ programs that use both recursive and non-recursive functions to perform the following searching operations for a Key value in a given list of integers:
    i) Linear search
    ii) Binary search
10. Write a program to convert postfix expression evaluation.

11. Write a program to convert infix to postfix conversion.

12. Write 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.

Course Outcomes:

At the end of this lab session, the student will

1. Be able to design and analyze the time and space efficiency of the data structure

2. Be capable to identity the appropriate data structure for given problem

3. Have practical knowledge on the application of data structures

MCA I Year – II SEM

16MCA206 UNIX AND SHELL PROGRAMMING PRACTICAL

L	P	C
0	3	2

Course Objectives:

1. This course is designed to give delegates practical experience in developing and writing shell scripts.
2. Most of the built-in unix shell commands are introduced together with the main program control structures.
3. The course also gives practical experience using a range of UNIX tools to manipulate text and incorporate into UNIX shell scripts.
4. To provide the skills needed to develop and customize Unix shell programs
5. To make effective use of a wide range of standard UNIX programming and development tools.

Lab Exercises:

1. Execution of various file/directory handling commands.
2. 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.
3. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
4. 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.
5. 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.
6. 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.
7. Write a shell script to list all of the directory files in a directory.
8. Write a shell script to find factorial of a given integer.
9. Write a shell script to calculate simple arithmetic operator.
10. Write a shell script to find the length of a given string.
11. Write a shell program to find out reverse string of the given string and check the given string is palindrome or not
12. Write a shell program to calculate the Fibonacci series by using function.
13. Write a shell program to check given number is prime or not by using function.
14. Write an awk command to print first field and second field only if third field value is >=50 in the given input file. (input field separator is ":" and output field separator is ",")
15. Consider the marks.txt is a file that contains one record per line(comma separate fields) of the student data in the form of studentid, student name, Telugu marks, English marks, Maths Marks, Science marks, Social Marks. Write an awk script to generate result for every

students in the form of studentid, studentname, Total Marks and result. Result is PASS if marks is >=30 in TELUGU and English, and if marks>=40 in other subjects. Result is fail otherwise.

16. Write an awk program to demonstrate user defined functions and system command.
17. Write an awk script to count the number of lines in a file that do not contain vowels.
18. Write an awk script to find the number of characters, words and lines in a file.
19. Write a grep command that selects the lines from the file1 that have exactly three characters

20. Write a grep command that selects the lines from the file1 that have at least three characters.

21. Write a grep command that selects the lines from the file1 that have three or fewer characters

22. Write a grep command that count the number blank lines in the file1

23. Write a grep command that copy the file to the monitor, but delete the blank lines.

24. Write a grep command that selects the lines from the file1 that have at least two digits without any other characters in between

25. Write a sed command that delete lines that contain **BEGIN** but not **END**

26. Write a sed command that swaps the first and second word in each line in the file

27. Write a sed command to delete character before last character in each line in a file

Course Outcomes:

After completion of the course students will be able to

1. Work confidently in Unix/Linux environment
2. Write shell scripts to automate various tasks
3. Write complex shell scripts.
4. Master the basics of linux administration
5. Students will learn Unix structure, commands, and utilities.Also, students will become versed with regular expressions and shell programming.

MCA II Year - I SEM

16MCA107 DATA STRUCTURES THROUGH PYTHON

L	T	P	C
4	0	0	4

Course Prerequisite: Concepts of Programming Logic and Techniques using Python.

Course Description:

This course gives the concepts of Data structures through python. It concentrates data structures, Python and their applications. It also explores various searching and sorting algorithms.

Course Objectives:

Student will be able to:

1. Acquire knowledge on basic constructs, functions and its modules of Python programming.
2. Get familiar with generic programming and Data Structures basics using Python.
3. Understand Python scripting language with Arrays, Stacks and Queues.
4. Understand more Queues and Linked Lists, implementations in Python.
5. Understand Non-Linear Data structures and Sorting & Searching Techniques and its implementations.

UNIT- I

BASIC PROGRAMMING CONCEPTS

VARIABLES, EXPRESSIONS AND STATEMENTS: Values and types, keywords, Operators Expressions, Interactive mode and script mode, String operations, Comments. **FUNCTIONS & MODULES**: Function calls, Type conversion functions, Math functions, Adding new functions, Definitions and uses, Flow of execution, Parameters and arguments, Random numbers, The time module, The math module. **CONDITIONALS:** Conditional execution, Alternative execution, Chained conditionals, Nested conditionals. Iteration: Multiple assignment, Updating variables, the while statement, break, continue.

UNIT- II

INTRODUCTION TO GENERIC PRORAMMING AND DATA STRUCTURES

Class Definition, Objects, Class Members, Constructors and destructors, Inline functions, static class members, this pointer, friend functions, dynamic memory allocation and deallocation (new and delete), exception handling, Function Over Loading, Operator Overloading.Inheritance basics and its types, Function and class templates, runtime polymorphism using virtual functions, abstract classes, streams I/O.

Introduction to Data Structures: Information and its meaning: Abstract Data Types, Sequences as Value Definitions, ADT for varying length character Strings, Data Types, Pointers and review of Pointers, Data Structures

UNIT- III

ARRAYS, STACKS AND QUEUES

Arrays: the Array as an ADT, Using One-dimensional Arrays, Implementing One-Dimensional Arrays, and Arrays as Parameters, Handling of Character Strings and Character Strings..

Stacks: Primitive operations, the stack as an ADT, Representing Stacks, and Testing for exceptional conditions, Implementing the push and pop operations, Examples for infix, postfix, and prefix expressions, Applications of Stacks: Regular Expressions, Expression Evaluations, etc.

UNIT-IV

QUEUES AND LINKED LISTS

Queues: The queue and its sequential representation, the queue as ADT, Insert operation, Priority queue, and Array implementation of a priority queue.

Linked lists: Inserting and removing nodes from a list, Linked implementations of stacks, get node and Free node operations, Linked implementation of queues, Linked list as a data Structure, Example of list operations, Header nodes, Array implementation of lists, Limitations of array implementation, allocating and freeing dynamic variables, Linked lists using dynamic variables, Circular lists, Stack as a circular lists, doubly linked lists, Application of Linked Lists, skip list representation.

UNIT-V

TREES, SORTING AND SEARCHING

Trees: Tree traversals, Binary Search Tree and Operations, AVL Tree and Operations, Red-Black Tree, Threaded binary trees and operations, B-Trees, B-Tree of order m, height of a B-Tree, insertion, deletion and searching, Comparison of Search Trees.

Sorting and Searching: Bubble sort, Quick sort, Selection sort, Tree Sorting: Binary Tree Sort, Heap Sort, and Insertion Sorts: Simple Insertion, Merge and Radix Sort. Basic Search Techniques: Algorithmic Notations, Sequential searching, Searching an ordered table, Indexed sequential search, Binary search, Interpolation search.

Course Outcomes:

1. Apply the knowledge of basic constructs, functions in Python programming.
2. Apply the concepts of ADR and Data Structures basics using Python.
3. Code in python scripting language with Arrays, Stacks and Queues.
4. Code and apply the concepts of Queues and Linked Lists, implementations in Python.
5. Apply the knowledge ofNon-Linear Data structures and Sorting & Searching Techniques and its implementations.

Text Books:

1. Problem Solving with Algorithms and Data Structures using Python by Bradley N. Miller, David L. Ranum is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.
2. Introducing Python by Lubanovic Bill
3. Data Structure and Algorithmic Thinking with Python by NarasimhaKarumanchi (Author)

MCA II Year – I Semester

16MCA108 COMPUTER NETWORKS

	L	T	P	C
Course Prerequisite: None	**4**	**0**	**0**	**4**

Course Description:

This course introduces the topologies, models, architecture, various components and protocols of computer networks. It narrates the analog and digital transmission related issues in various networking environments. It explains about the domain naming system and network management protocols in detail. It provides basic information about network attacks, mechanisms and services provided by various layers.

Course Objectives:

1. To get familiar with basics of networking concepts functions of various layers in networking architecture
2. To know the basics of transmissions techniques and media used in networking environment.
3. To understand the underlying components used in networking environments and the different versions of Internet Protocol.
4. It explains about DNS and the network management protocols in detail
5. To acquire the knowledge about network vulnerabilities and security management.

UNIT-I

INTRODUCTION

Data communication-networks - network models-network topologies- The Need for a Protocol Architecture-A Simple Protocol Architecture - The OSI and TCP/IP Protocol Architecture - Standardization within a Protocol Architecture - Traditional Internet-Based Applications.

UNIT-II

DATA TRANSMISSION

Concepts and Terminology - Analog and Digital Data Transmission - Transmission Impairments - Channel Capacity. Transmission Media - Digital Data Communication Techniques-Data link control protocols – Multiplexing – Circuit and Packet switching – ATM.

UNIT-III

LAN

Ethernet –Types of Ethernet - Wireless LAN – internet protocols IPv4 and IPv6 - Internetwork operations – Quality of services – role of MPL and MPL operations – Transport protocols – Routing – Congestion Control – High Speed LAN.

UNIT – IV

NAME SPACE AND NETWORK MANAGEMENT

Domain Name System - Remote Logging, Electronic Mail, and File Transfer - WWW and HTTP - Network Management: SNMP –Multimedia-ICMP-IGMP-Unicast and multicast routing.

UNIT-V

NETWORK SECURITY

Overview – Attacks - Symmetric Key and Asymmetric Key Cryptography – Security services – Digital Signature –Message Confidentiality – Message Integrity – IP security – PGP - Firewall and types of Firewalls.

Course Outcomes:

1. Explain the basic networking concepts and importance of layered architecture
2. Understand the various techniques and media used for data transmission process
3. Explore the responsibilities of protocols and components which are supporting networking environment
4. Understand the domain naming system and network management protocols.
5. Design the secured network environment to safeguard the data transmission

Text Books:

1. "DATA AND COMPUTER COMMUNCIATIONS" By WILLAIM STALLINGS, 8th and 9th Edition, Prentice Hall
2. "DATA COMMUNICATIONS AND NETWORKING" BY FOROUZAN, 4th Edition, McGraw Hill

Reference Books:

1. "NetworkSecurityEssentialsApplicationsandStandards",WilliamStallings,PearsonEducation.
2. "Cryptography and Network Security", By FOROUZAN, Special Edition, Tata McGraw Hill.
3. "Cryptography and Network Security", Atulkahate, Tata McGraw Hill.

Mode of Evaluation: Assignment, Seminar, Written Examination.

MCA II Year – I Semester

16MCA109 SOFTWARE ENGINEERING

L	T	P	C
4	0	0	4

Course Prerequisites: None

Course Description:
This course is aimed at helping the students build up an understanding of how to develop a software system from scratch by guiding them through the development process and giving them the fundamental principles of system development with object oriented concepts.

Course Objectives:

Student will be able to understand:

1. The need of software engineering process and various process models.
2. The significance of software requirements, modeling & design concepts
3. The importance of User Interface Design and pattern based design in software development.
4. The role of various testing strategies.
5. The weightage of process metrics, risk management and reengineering.

UNIT -I

SOFTWARE ENGINEERING PROCESS & MODELS

The nature of Software, The unique nature of WebApps, Software engineering- A layered technology, The essence and principles of software engineering practice, Generic process model (framework), Process patterns, Process assessment and improvement, CMMI, Software myths.
Prescriptive process models: The waterfall model, Incremental process models, Evolutionary process models. The Unified process, Aspect oriented software development, Agile development: Agile process, Extreme programming.

UNIT- II

SOFTWARE REQUIREMENTS, MODELING& DESIGN CONCEPTS

Software Requirements: Introduction to functional and non-functional requirements, Requirements engineering activities, Eliciting requirements, Requirements modelling, Requirements validation, Software requirements specification (SRS), Requirements management.
Requirements modeling:Structured view: Data modelling (ERD), Flow-Oriented modelling (DFD), Behavioural modelling, Object models, and structured methods**. Software Project Estimation**: Empirical estimation models.
Design Concepts: Software design quality guidelines and attributes, Design concepts, Design model. Software Architecture: Architecture and its importance, Architectural Styles, Data design, Architectural design. Design: Structured view (Traditional view): Architectural mapping using data flow (Call and return architecture), Interface design, Function based component design.

UNIT – III

USER INTERFACE DESIGN & PATTERN BASED DESIGN

Performing 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.

UNIT – IV

TESTING STRATEGIES, CONVENTIONAL AND OBJECT ORIENTED TESTING

Software testing strategies: A strategic approach to software testing, Test strategies (Unit testing and integration testing) for conventional and object oriented software, 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.

UNIT – V

PROJECT METRICS, RISK MANAGEMENT AND REENGINEERING

METRICS in the process and project, software measurements, metrics for software quality, integrating metrics within the software process, Metrics for small organizations.
RISK MANAGEMENT Reactive vs proactive risk strategies, Software Risks, Risk Identification, Projection and refinement RMMM.
REENGINEERING Business process reengineering, software reengineering, reverse engineering.

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. Understand the role of software engineering process and various process models in software development.
2. Acquire the knowledge of software requirements, modeling & design concepts
3. Know the need of User Interface Design and pattern based design in software development
4. Differentiate which testing techniques can be applied the areas of software
5. Understand the role of project metrics, Risk and Re engineering concepts

Text Books:

1. Roger S. Pressman, “Software Engineering – A practitioner’s Approach”, Seventh Edition, McGraw-Hill International Edition, 2005.
2. Lan Somerville, “Software engineering”, Seventh Edition, Pearson Education Asia, 2007.

Reference Books:

1. Software Engineering, an Engineering approach-James F. Peters, WitoldPedrycz, John Wiely.

2. Software Engineering principles and practice-Waman S Jawadekar, The McGraw-Hill Companies

MCA II Year – I Semester

16MCA110 DESIGN AND ANALYSIS OF ALGORITHMS

L	T	C
4	0	4

Course Prerequisite: Data Structures

Course Description:
This course describes about asymptotic analysis of algorithms. It covers efficient algorithms for sorting, searching, and selection. Various algorithm design techniques such as divide-and-conquer, Greedy methods, Dynamic programming, back tracking and Branch and Bound algorithms are explained. This course also deals with lower bound theory and NP-hard and NP-complete problems.

Course Objectives:

Student will be able to:

1. Analyze asymptotic performance of algorithms.
2. Apply divide and conquer & greedy methods for sorting and searching problems.
3. Apply dynamic programming method for solving real time problems.
4. Explore the applicability of back tracking & branch and bound techniques to solve suitable problems.
5. Understand the NP-hard and NP-complete problems.

UNIT – I

INTRODUCTION
Algorithm Specification, Performance analysis, Space Complexity, Time Complexity, Asymptotic Notation(O,Omega,Theta), Practical Complexities, Performance Measurement, Review of elementary data structure- Heap and Heap Sort, Hashing, Set representation. UNION, FIND.

UNIT – II

DIVIDE AND CONQUER & GREEDY METHOD
The general method, finding maximum, minimum. Merge sort, quick sort and selection sort. Knapsack problem, Optimal Storage on tapes, Job sequencing with deadlines, Optimal merges patterns, Minimum Spanning Trees.

UNIT- III

DYNAMIC PROGRAMMING AND TRAVERSAL TECHNIQUES
Multistage graph, All Pair Shortest Path, Optimal Binary Search trees, 0/1 Knapsack, Reliability Design, Traveling Salesman Problem, Bi connected Components, Breadth First Search and Depth First Search.

UNIT - IV

BACKTRACKING & BRANCH AND BOUND

8-Queens Problem, Graph Coloring, Hamilton cycle, Knapsack Problem, 0/1 Knapsack Problem, Traveling salesperson problem, Lower Bound Theory.

UNIT – V

NP-HARD AND NP-COMPLETENESS

Basic concepts, Cook's theorem, NP-hard Graph problems and scheduling problem, NP-hard code generation problems, Clique Decision problem, Node cover decision problem, NP-hard code generation problem.

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. Determine the time complexities of algorithms using asymptotic analysis.
2. Explain and implement algorithms using divide-and-conquer & greedy methods.
3. Explore the dynamic programming method for solving suitable problems. Acquire the knowledge about different traversal techniques.
4. Implement algorithms using back tracking & branch and bound methods.
5. Understand the problems belongs to the classes of NP-hard and NP-complete.

Text Books:

1. Horowitz E. Sahani S: "Fundamentals of Computer Algorithm", Galgotia Publications.
2. AnanyLevitin, "Introduction to the Design & Analysis, of Algorithms", Pearson Education, 2000.
3. Aho, Hopcroft, Ulman, "The Design and Analysis of Computer Algorithm", Pearson Education, 2000.
4. Parag H. Dave, Himanshu B. Dave "Design and Analysis of Algorithms" Pearson Education, 2008.

Mode of Evaluation: Assignments, I Mid Exam, II Mid Exam and End Exam.

MCA II Year – I Semester

16MCA207 DATA STRUCTURES THROUGH PYTHON PRACTICAL

L	T	P	C
0	0	3	2

Course Objectives:

Students are expected to
1. Apply the Python programming concepts to various problems
2. Understand the differences in Python and other programming approaches.
3. Solve the ADT using Python programmes
4. Apply the Python programmes to solve liner and non-linear data structure
5. Solve the sorting and searching operation using Python programming

List of Experiments:

1. Write a program to implements the array ADT using array capabilities of the ctypes module
2. Implementation of the Matrix ADT using a 2D Array
3. Program for playing the game of Life
4. Implementation of the Set ADT container using a Python list
5. Implementation of Map ADT using a single list
6. Implementation of the MultiArray ADT using a 1-D array
7. Implementation of the Binary Search algorithm
8. Implementation of the Bubble Sort algorithm
9. Implementation of the selection and insert sort algorithms
10. Write a program to implement single linked and double linked list
11. Implementation of the Polynomial ADT using a sorted linked list

Course Outcomes:

Students will able to
1. Understand the Python programming paradigm
2. Apply the Python programming approaches to solve problems related to ADT
3. Solve problems related to linear and non-linear data structures using Python concepts,
4. Solve sorting and searching programs using Python.

MCA II Year – I Semester

16MCA208 COMPUTER NETWORKS PRACTICAL

L	T	P	C
0	0	3	2

Course Prerequisite: Java programming language and basics of networks**.**

Course Description:

Introduction to local, metropolitan, and wide area networks environments and their components. Provides an overview of Internet protocol suite and network tools and networking based commands. Also gives exposure on TCP and UDP based basic socket programming concepts.

Course Objectives:

1. To acquire more knowledge on the terminology and concepts of the OSI reference model and the TCP-IP reference model.
2. To obtain the deep knowledge on the concepts of protocols, network interfaces, and design/performance issues in local area networks and wide area networks,
3. To familiar with TCP/UDP based socket programming.

List of Experiments:

1. Study of different types of network cables and protocols
2. Study of network devices in detail.
3. Study of network IP.
4. Connect the computers in LAN network
5. Study of basic network commands and network configuration commands
6. Study of Network Utilities(ping, tracert, arp, ifconfig, nslookup,) on both platforms(Windows/Linux)
7. Study of Network Utilities(telnet, ftp,wget, curl,netstat) on both platforms(Windows/Linux)
8. Study of IP Addressing, sub-netting.
9. Design of a network in simulator(NS2/Cisco Simulator)
10. Write a program for distance vector algorithm to find suitable path for transmission.
11. Study of wireshark and tcpdump.
12. Analyze packets using Wireshark(Ethernet/W,Http,etc)
13. To study the Transmission Control Protocol (TCP) Connection states, Flags and Flow control.
14. Write a Java program for implement the concept of TCP Date Server/Client
15. Write a Java program for implement the concept of TCP Echo Server/Client
16. Write a Java program for implement the concept of TCP Chat Server/Client
17. Write a Java program for implement the concept of TCP File Server / Client
18. Write a Java program for implement the concept of UDP Echo Server/Client

19. Write a Java program for implement the concept of UDP Chat Server/Client
20. Write a Java program for implement the concept of UDP DNS Server/Client

Course Outcomes:

1. Can acquire more knowledge on the terminology and concepts of the OSI reference model and the TCP-IP reference model.
2. Can obtain the deep knowledge on the concepts of protocols, network interfaces, and Design / performance issues in local area networks and wide area networks,
3. Become familiar with TCP/UDP based socket programming.

Mode of Evaluation: Practical, Written Examination

MCA II Year – I Semester

16MCA209 SE AND UML PRACTICAL

L	T	P	C
0	0	3	2

Course Prerequisite: None

Course Description:
The course will focus on the object-oriented approach for analysis and design. Students will gain the knowledge of doing analysis, design and writing programs. They work in small groups, each group having the responsibility for analysis, design and implementation of a software system.

Course Objectives:
1. Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.

List of Exercises:
To develop a mini-project following the 12 exercises listed below:

1. To develop a problem statement.
2. Develop an IEEE standard SRS document. Also develop risk management and project plan (Gantt chart).
3. Identify Use Cases and develop the Use Case model.
4. Identify the business activities and develop an UML Activity diagram.
5. Identity the conceptual classes and develop a domain model with UML Class diagram.
6. Using the identified scenarios find the interaction between objects and represent them using UML Interaction diagrams.
7. Draw the State Chart diagram.
8. Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.
9. Implement the Technical services layer.
10. Implement the Domain objects layer.
11. Implement the User Interface layer.
12. Draw Component and Deployment diagrams.

Suggested domains for Mini-project:

1. Passport automation system.
2. Book bank
3. Exam Registration
4. Stock maintenance system.
5. Online course reservation system
6. E-ticketing
7. Software personnel management system
8. Credit card processing
9. e-book management system
10. Recruitment system

11. Foreign trading system
12. Conference Management System
13. BPO Management System

Course Outcomes:

Students are expected to demonstrate:

1. How the object-oriented approach differs from the traditional approach in systems analysis and design.
2. The importance of modeling and how the Unified Modeling Language (UML) represents an object-oriented system using a number of modeling views.
3. How to construct various UML models (including use case diagrams, class diagrams, interaction diagrams, state chart diagrams, activity diagrams, and implementation diagrams) using the appropriate notations.

Suggested Software Tools:

ArgoUML, Eclipse IDE, Visual Paradigm, Visual case, Star UML and Rational Suite

Mode of Evaluation: Practical, Written Examination

MCA II Year – I Semester

16MCA210 DESIGN AND ANALYSIS OF ALGORITHMS PRACTICAL

L	P	C
0	3	2

Course Prerequisite:
Knowledge of any Programming Language , Data Structures concepts

Course Description:
Several fundamental principles of algorithm design techniques as Divide and Conquer, Greedy method, Branch & Bound and Backtracking are explained. For computing on graphs, such as how to compute connectivity information and finding shortest paths are covered.

Course Objectives:

The student will be able to

1. To analyze Bestcase,Average case ,worst-case running time of algorithms.
2. To introduce the methods of designing and analyzing algorithms
3. To study about various designing paradigms of algorithms for solving real world problems.

List of Experiments :

1. To implement Insertion sort using array as a data structure and analyse its time complexity.
2. To implement Merge sort using array as a data structure and analyse its time complexity.
3. To implement Quick sort using array as a data structure and analyse its time complexity.
4. To implement Randomized Quick sort using array as a data structure and analyse its time complexity.
5. To implement Bubble sort using array as a data structure and analyse its time complexity.
6. To implement Bucket sort using array as a data structure and analyse its time complexity.
7. To implement Radix sort using array as a data structure and analyse its time complexity.
8. To implement Shell sort using array as a data structure and analyse its time complexity.
9. To implement Counting sort using array as a data structure and analyse its time complexity.
10. To implement Selection sort using array as a data structure and analyse its time complexity.
11. To implement Heap sort using array as a data structure and analyse its time complexity.
12. To implement Exchange sort using array as a data structure and analyse its time complexity.
13. To implement sorting algorithm of your own choice using array as a data structure and analyze its time complexity.
14. To implement linear search and analyze its time complexity.
15. To implement Binary search and analyze its time complexity.
16. To implement Recursive Binary search and analyze its time complexity.
17. To implement Fibonacci search and analyze its time complexity.
18. To implement searching algorithm of your own choice and analyse its time complexity.
19. To implement Matrix Multiplication and analyse its time complexity.
20. To implement Breadth First Search and analyse its time complexity.
21. To implement Depth First Search without backtracking and analyse its time complexity.
22. To implement Depth First Search with backtracking and analyse its time complexity.

23. To implement Best First Search and analyse its time complexity.
24. To implement Kruskal's algorithm and analyse its time complexity.
25. To implement Prim's algorithm and analyse its time complexity.
26. To implement Tower of Hanoi problem and analyse its time complexity.
27. To implement n-Queens problem and analyse its time complexity.
28. To sort both an array and a linked list using any sorting algorithm and compare their time complexity.
29. To implement 8-puzzle problem and analyse its time complexity.
30. To implement Travelling Salesman problem and analyse its time complexity.

Course Outcomes:

1. To specify the complexities of various problems in different domains.
2. To implement methods for analyzing the efficiency and correctness of algorithms (such as exchange arguments, recurrence, induction, and average case analysis)
3. To develop the efficient algorithms for the new problem with suitable designing techniques.

Mode of Evaluation: Observation, Record, Program Execution and Viva.

MCA II Year – II Semester

16MCA111 JAVA PROGRAMMING

L	T	P	C
4	0	0	4

Course Prerequisite: Concepts of Object Oriented Programming.

Course Description:
This course introduces core Java. It covers the history and evolution of java, basic constructs of the programming language. It gives exposure on arrays, string related classes, reusability of code, handling run time errors, multithreaded programming. It also introduces various classes and methods of input/output stream, networking, collections and swings.

Course Objectives:

Student will be able to:

1. Understand the evolution and basic constructs of Java.
2. Acquire knowledge on Arrays, Strings, code reusability and handling exceptions.
3. Understand the role of multithreading and applicability of wrapper classes.
4. Explore the utilities of I/O, Collections and Networking classes.
5. Build GUI through Swings and event handling.

UNIT- I

INTRODUCTION TO JAVA
The Java Language: The History and Evolution of Java, An Overview of Java, Data Types, Variables, Operators, Control Statements, Introducing Classes, A closer Look as Methods and Classes, Overloading Methods, Constructor and Overloading Constructors, Using Object as parameters, Returning Objects, Recursion, Introducing Access Control, Static Keyword, final Keyword, Nested and Inner Classes.

UNIT- II

ARRAYS, STRINGS, INHERITANCE AND EXCEPTION HANDLING
Arrays, String and StringBuffer Classes, Inheritance: Definition, types of Inheritance, Super Keyword, Method Overriding, Using Abstract Class and Final with Inheritance, The Object Class, Interface, Packages.
Exception Handling: Fundamentals, Exception Types, Un Caught Exceptions, Using Try and Catch, Multiple catch Clauses, throw, throws, finally, Java Built-in Exceptions.

UNIT- III

MULTI-THREADING AND WRAPPER CLASSES

Multithreaded Programming: The Java Thread Model, thread Properties, The Main Thread, Life cycle of Thread, Creating Thread, creating Multiple Threads, using isAlive and join method, suspending, resuming and stopping threads. Enumerations, Auto boxing and Annotations (Metadata): Enumerations, Type Wrappers, Auto boxing, Annotations.

UNIT- IV

I/O STREAMS, COLLECTIONS, NETWORKING

Input/Output: The Java I/O Classes and Interfaces, the Stream Classes. The Byte Streams: InputStream, OutputStream, FileInputStream, FileOutputStream, PrintStream, DataInputStream, DataOutputStream. The Character Streams: Reader, Writer, FileReader, FileWriter, CharArrayReader, CharArrayWriter, BufferedReader, BufferWriter,

The Collections Framework: Collections Overview, the Collection Interfaces, the Collection Classes.

Networking: Networking Basics, the Networking Classes and Interfaces, InetAddress, TCP/IP Client Sockets, URL, URLConnection, the URI Class, Datagrams

UNIT- V

SWINGS AND EVENT HANDLING

GUI Programming with Java -Introduction to swings, JLabel, JImageIcon, JTextField, JButton, JToggleButton, JChekcBoxes, JRadioButtons, JTabbedPane, JScrollPane, JList, JComboBox, JTrees, JTable.

Event Handling: Event Handling Mechanisms, The Delegation Event Model: Events, Event Sources, Event Listeners, Event Classes, And Event Listener Interfaces

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. Write java code using basic constructs of java.
2. Write code for arrays, strings and inheritance.
3. Apply multithreaded and wrapper class concepts in developing applications.
4. Use various I/O, Collections and Networking classes appropriately.
5. Write code for GUI components using Swings and event handling.

Text Books:

1) The Complete Reference Java to, Herbert Scheldt, 7th Edition.
2) Java for Programmers, P.J.Deitel and P.J.Deitel, Pearson education / Java: How to Program P.J.Deitel and P.J.Deitel, 8th edition, PHI.

Mode of Evaluation: Assignment, Seminar, Written Examination.

MCA II Year – II Semester

16MCA211 JAVA PROGRAMMING PRACTICAL

L	T	P	C
0	0	3	2

Course Prerequisite: Practical knowledge on OOPs concepts.

Course Description:
In this course, students gain extensive hands-on experience on Java programming. Students learn to create robust console applications using code reusability with multi-threading, applications of exception handling, I/O streams and GUI implementation.

Course Objectives:

1. Implement java programs using basic constructs.
2. Build robust applications using object-oriented features.
3. Read and write data using Java streams.
4. Develop platform-independent GUIs.

List of Experiments:

1. Write a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2 -4ac is negative, display a message stating that there are no real solutions.
2. Fibonacci sequence is defined by the following rule: The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a Java program to print the Fibonacci sequence.
3. Write a Java program to print the individual digits of a given number.
4. Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.
5. Write a Java program to print Multiplication table
6. Write a Java program to find both the largest and smallest number in a list of integers.
7. Write a Java Program Method Overloading, Constructor and Constructor overloading.
8. Write a Java program to Perform Matrix Operations Addition, Subtraction, and Multiplications.
9. Write a Java program to find factorial of given number.
10. Write a java program to perform linear search.
11. write a java program which creates InputStreamReader to read standard input stream until the user press "Q or q" (Q means Quit)
12. Write a Java program to Demonstrate ByteStream Classes (FileInputStream and File OutputStream)
13. Write a java program to generate prime numbers for the given range and store it in a file called "primenumbers.txt".
14. write a java program to read contents of a file "primenumber.txt"(which created in Exp NO.---) and display only even number of that file.
15. Write a java program to copy the contents of a file to another file.

16. Write a java program to read input any ".java" file and display as output.(Java file itself as output).
17. Write a java program to use list () method of File object to list down all the files and directories available in a directory.
18. Write a java program to filter the files by file extensions and show the file names. (lists only files that meet a certain filtering criteria like *.java, *.txt)
19. Write a Java program to perform String Operations.
20. 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.
        i. Write a Java program that checks whether a given string is palindrome or not. Ex: MADAM is a palindrome
21. Write a Java program to demonstrate Inheritance.
22. Write a Java program to demonstrate Interfaces.
23. Write a java program to demonstrate Packages.
24. Write a java program to demonstrate Exception using try and multiple catch blocks.
25. 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
26. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 andNum2. 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 Num2were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.
27. Write a program to build simple Calculator using Java Swings Components.
28. Write a java program to demonstrate all the Swing components textbox, password, radio button, and selection and choices boxes.
29. Write a Java program to demonstrate Swing components like JLabel, JTextField, JPasswordField, JTextArea, JButton, JRadioButton, JList, and JMenuBar.
30. Develop a user interface for New User Sign-Up for E-Mail Creation.

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. Develop programs using basic constructs based on the Java structure.
2. Develop robust applications.
3. Implement programs for managing I/O operations.
4. Code for various functionalities of GUI.

Mode of Evaluation: Practical, Written Examination

MCA III Year – I Semester

16MCA112 DATA WAREHOUSING AND MINING

L	T	P	C
4	0	0	4

Course Prerequisite: Database Concepts

Course Description:
This course introduces data warehousing concepts and various data mining techniques. Data Mining is the process of extracting the hidden predictive information from large data sets. Data warehousing involves data preprocessing, data integration and providing on-line analytical processing (OLAP) tools for the interactive analysis of multidimensional data. It also includes the various processes such as spatial mining, text mining and web mining.

Course Objectives:

Students will be able to:

1. Know the fundamentals of Data mining and Data preprocessing techniques.
2. Understand the data warehouse architecture & OLAP and discover the frequent Item sets for the given data set.
3. Predict the knowledge from data sets using Classification techniques.
4. To understand the various clustering methods.
5. Learn the mining concepts for 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
Mining Frequent Patterns, Associations and Correlations: Basic Concepts, Efficient and Scalable Frequent Item set Mining Methods, Mining various kinds of Association Rules, From Association Mining to Correlation Analysis, Constraint-Based Association Mining

UNIT- III

CLASSIFICATION AND PREDICTION

Issues Regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification, Rule-Based Classification, Classification by Back propagation, Other Classification Methods, Prediction, Accuracy and Error measures, Evaluating the accuracy of a Classifier or a Predictor.

UNIT- IV

CLUSTER ANALYSIS

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 .

UNIT- V

MINING OBJECT, SPATIAL, MULTIMEDIA, TEXT, WEB AND TIME SERIES DATA

Multidimensional Analysis and Descriptive Mining of Complex Data Objects, Spatial Data Mining, Multimedia Data Mining, Text Mining, Mining the World Wide Web, Mining Time Series data.

Course outcomes:

Upon successful completion of this course, students will be able to:

1. Explore the knowledge about fundamentals of Data Mining and implement the methods of data preprocessing.
2. Generate the frequent item sets for the given data set using association rule mining.
3. Extract the knowledge by classification techniques using data mining tool.
4. Perform cluster analysis for the given data sets.
5. Apply the data mining algorithms to advanced database applications.

Text Book:

Data Mining – Concepts and Techniques - Jiawei Han &MichelineKamber, Morgan Kaufmann Publishers, 2nd Edition, 2006.

Reference Books:

1. Data Mining Introductory and advanced topics –Margaret H Dunham, Pearson education
2. Data Mining Techniques – Arun K Pujari,2nd edition, Universities Press.
3. Introduction to Data Mining – Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Pearson education.
4. Insight into Data Mining,K.P.Soman,S.Diwakar, V.Ajay,PHI,2008.
5. Data Warehousing Fundamentals – PaulrajPonnaiah Wiley student Edition
6. Data Mining,V.Pudi and P.Radha Krishna, Oxford University Press.

Mode of Evaluation: Assignments, I Mid Exam, II Mid Exam and End Exam.

MCA III Year – I Semester

16MCA113 CRYPTOGRAPHY AND NETWORK SECURITY

L	T	P	C
4	0	0	4

Course Prerequisite: Knowledge of Computer Networks

Course Description:

It introduces techniques for protecting information and network components against attacks. Highlights the security mechanisms for both wired and wireless networks and their implementation procedure to provide security. Describes about the Wireless LAN Concepts and the security and how to protect Data from different attacks and vulnerabilities. Discusses about the Mobile security in GSM and UMTS. Highlights the concepts of protocol vulnerabilities that includes hijacking and spoofing.Illustrates the network viruses, worms and firewall concepts and firewall issues. Provides information about the Web Services Security and Malware and prevention and detection

Course Objectives:

Student will be able to

1. Understand the types of Attacks, Elementary Ciphers, Security Layers and security protocols
2. Know the concepts of LAN Security and the related protocols, GSM and UMTS security, cellphone security
3. Acquire knowledge related to vulnerabilities of protocols and Access control in OS
4. Understand about the viruses, worms and Firewalls and their security
5. Acquire knowledge on Intrusion Prevention and Detection

UNIT – I

INTRODUCTION – Cyber Attacks. **Defense , Strategies and Techniques,**Guiding Principles. **Basics of Cryptography:** Preliminaries – Secret versus public key cryptography – Types of Attacks – Elementary Substitution Ciphers – Elementary Transposition Ciphers – Other Cipher Properties **– IPSec Security at the Network Layer**- IPSec in Action- Internet Key Exchange Protocol- Security Policy and IPSec- VPN-**Security at Transport Layer-**SSL Handshake protocol- SSL Record Layer Protocol- Open SSL

UNIT – II

IEEE802.11 Wireless LAN Security: Background – Authentication –Pre WEP Authentication – Authentication in WEP – Authentication and key Agreement in 802.11. **Confidentiality and Integrity:** Data Protection in WEP – Data Protection in TKIP and CCMP

Cellphone Security: Preliminaries – Entities Involved – Security Goals – **GSM (2G) Security:** Entity Authentication and Key Agreement – Encryption – Problems and Drawbacks – **Security in UMTS (3G):** Security Enhancements – Authentication and Key Agreement – Integrity Protection and Encryption

UNIT III

Non-Cryptographic Protocol Vulnerabilities: DoS and DdoS, Session Hijackig and Spoofing, Pharming Attacks, Wireless LAN Vulnerabilities- **Software Vulnerabilities:** Phishing, Buffer Overflow, Format String Attacks, Cross Site Scripting (XSS), SQL Injection – **Access Control in the Operating System:** Preliminaries, Discretionary Access Control – Case Studies: Windows / UNIX, Mandatory Access Control, Role Based Access Control, SELinux and Recent Trends

UNIT IV

Viruses , Worms and other Malware- Virus and Worm Features- Internet scanning worms- Topological Worms- Web Forms- Mobile Malware- Botnets. **Firewalls-** Basics- Practical Issues- Personal Firewalls

UNIT V

Intrusion Prevention and Detection: Introduction – Prevention Versus Detection – Types of Intrusion Detection Systems – DdoS Attack Prevention / Detection – Malware Detection. **Web Services Security:** Motivation – Technologies for Web Services – WS Security - SAML – Other Standards

COURSE OUTCOMES:

Upon Successful Completion of this course students will be able to

1. Identify different types of attacks and implementing the security mechanisms in network and transport layer
2. Categorize the WEP protocol functionalities and data protection using different protocols
3. Classify different Non vulnerability protocols and gain knowledge about software vulnerabilities
4. Classify different types of Worms and Viruses, Malware and Firewalls
5. Find the Intrusion detection systems and prevention systems and can learn different types of technologies on web services and standards.

Text Books:

1. Network Security and Cryptography, Bernard Menezes, CENGAGE Learning
2. Cyber Security: NnaGodbole, SunitBelapure, Wiley India

Reference Books:

1. CryptographyandNetworksecurity,AtulKahate,TataMcGraw-HillPubcompanyLtd.,NewDelhi
2. Network Security Hacks: Andrew Lockhart, O'Reilly, SPD
3. Cryptography and Network Security: ForouzanMukhopadhyay, McGraw Hill, 2nd Edition
4. Principles of Computer Security: WM.Arthur Conklin, Greg White, TMH
5. Wireless Security Models, Threats and Solutions: Randall K.Nichols, PanosC.Lekkas, TMH
6. Computer Security: Dieter Gollman, 2nd Edition, Wiley India
7. Computer Evidence: Collection and Prevention, Christopher L.T.Brown, Firewall Media

MCA III Year−I Semester

16MCA212 DATA MINING AND DATA WAREHOUSE PRACTICAL

L	T	P	C
0	0	3	2

Course Prerequisite: Knowledge on Databases

Course Description:

This course introduces Data Warehousing tool (e.g ETL-Informatica) for various data transformations and Data mining tool (e.g WEKA) is used to apply machine learning algorithms.

Course Objectives:

Students will be able to:

1. Get familiar with WEKA and ETL tool environment.
2. Develop skills for using data mining tool to solve the practical problems.
3. Implement data transformations using mapping techniques.

I. Using WEKA tool Implement the following tasks:

Task-1
Introduction to weka.
Task-2
Mine the association rules for the following.
(a) Weather data set using Apriori Algorithm.
(b) Buy's computer data set.

Task-3
Classification Techniques.
(a) Decision tree
(b) REP tree
(c) MLP, Baye's classification
(d) Converting a Decision Tree into If – Then Rules using PART and ONE-R algorithm.

Task-4
Answer the following queries for loan data set using classification technique.
a) List all the categorical (or nominal) attributes and the real-valued (or numeric) attributes separately.
b) What attributes do you think might be crucial in make the credit assessment? Come up with some simple rules in plain English using your selected attributes.
c) 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.
d) 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?

e) Is testing on the training set as you did above a good idea or not?
f) One approach for solving the problem encountered in the previous question is using cross-validation. Describe what is cross validation briefly? Train a decision tree again using cross validation and report your results. Does your accuracy increase/decrease? Why?
g) Check to see if the data shows a bias against "foreign workers", OR "personal status". One way to do this (perhaps rather simple method) 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.
h) 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?
i) 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 tree you obtain? Also report your accuracy using the pruned model. Does your accuracy increase?

Task-5

Clustering

(a) K-means Algorithm.

II. Using Informatica Tool,
Perform the following transformations like
Filter Transformation,
Joiner Transformation,
Expression Transformation,
Ranker Transformation,
Router Transformation etc.

Course Objectives:

Students will be able to:

1. Get familiar and use WEKA and ETL tool environment.
2. Show the skills in using data mining tool to solve the practical problems.
3. Implement data transformations using mapping techniques using tools.

MCA III Year–I Semester

16MCA213CRYPTOGRAPHY AND NETWORK SECURITY PRACTICAL

L	T	P	C
0	0	3	2

Course Prerequisite: Knowledge on any high-level programming language.

Course Description:

This course provides technical grounding in cryptography, network security concepts and algorithms. It deals with the security cryptographic algorithms like Blowfish, Rijndael, Digital Signature etc. In essence it helps to the user to secure the confidentiality of the data.

Course Objectives:

Student will be able to understand:

1. Get knowledge on basic constructs of security programs
2. Acquire knowledge on Advanced DES, AES, RSA and Digital Signature algorithms
3. Experiment cryptographic based encryption and decryption algorithms

List of Programs

1. Write a C program that contains a string (char pointer) with a value 'Hello World', the program should XOR each character in this string with 0 and displays the result.
2. Write a C program that contains a string (char pointer) with a value 'Hello World', the program should AND or XOR each character in this string with 127 and display the result.
3. Write a program to implement the concept of Encryption and Decryption using Ceaser Cipher.
4. Write a program to implement the concept of Encryption and Decryption using Substitution Cipher.
5. Write a program to implement the concept of Encryption and Decryption using Hill Cipher.
6. Write a program to implement DES algorithm logic.
7. Write a program to implement BlowFish Algorithm using Java Cryptography and create your own key using Java keytool.
8. Write a Java program to implement the Rijndael algorithm.
9. Write a program to implement RSA algorithm.
10. Write a program to implement the concept of message digest of a text using the SHA-1 algorithm in Java.
11. Write a program to implement the concept of message digest of a text using the MD5 algorithm in Java.
12. Write a Program to implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. They will be able to write program on basic construct of cryptography & network security programs
2. Familiar with important cryptographic algorithms like DES, BlowFish, Rijndael etc.,
3. Capable of implementing encryption and decryption in simple login forms

Mode of Evaluation: Observation, Record, Program Execution and Viva

DISCIPLINE ELECTIVE-I

MCA II Year – II Semester

16MCA401 WEB PROGRAMMING THROUGH PHP

L	T	P	C
4	0	0	4

Course Prerequisite: Knowledge of HTML tags and any usage of scripting language.

Course Description:

This course enables students to have strong fundamentals in web programming though PHP, including the flavor of Object oriented approach. This course helps, students to understand the various in-depth concepts of web page designing, RDBMS and to establish connection it with back end servers. Makes students to have hands on experience to design and develop client-server architecture using MySQL and PHP.

Course Objectives:

Student will be able to understand:

1. PHP programming and its environment.
2. Functions and core reusability facilities available in PHP.
3. Object oriented approach in PHP.
4. Basic and advanced controls and working mechanism of RDBMS.
5. Connecting PHP with RDBMS for end to end application development.

UNIT- I

BASIC AND ADVANCED PROGRAMMING WITH HTML

Basic tags of HTML, Advanced tags of HTML, Using PHP, Embedding PHP in HTML, Adding dynamic content, Accessing form variables, Understanding identifiers, Creating user declared variables, Examining variable types, Understanding variable scope, Operators, Implementing control structures, Making decisions with conditionals, Repeating actions through iterations, Storing and retrieving data, Arrays, Array operators, Multi-dimensional arrays, Array manipulations, String manipulation and regular expressions.

UNIT- II

WORKING WITH FUNCTIONS AND CODE REUSE

Functions in PHP, Reusing Code, Using functions in PHP, Functions for web page templates, Understanding the scope, Passing by reference and passing by value, Returning from functions and implementing recursion.

UNIT-III

WORKING AND EXTENDING WITH CLASSES

Understanding Object Oriented PHP, Creating classes, attributes and operations in PHP, Controlling access and Inheritance in PHP. Advanced and new object oriented functionality in PHP. Exception handling in PHP.

UNIT- IV

INTRODUCTION TO RDBMS

RDBMS concepts, Database and Web database architecture. How to design own web database. Introducing MySQL, Introducing MySQL's privilege systems, Creating databases and understanding MySQL identifiers. Working with MySQL databases.

UNIT- V

WORKING WITH MYSQL AND PHP

Accessing MySQL database from the Web with PHP.Advanced SQL administration and SQL programming.Implementing authentication with PHP and MySQL.Implementing secure transactions with PHP and MySQL. Generating, parsing and transforming XML, Security in PHP. Application techniques in PHP.

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. Do basic programming and advanced scripting using PHP.
2. Concepts and benefits of functions and code reusability in PHP.
3. Apply object oriented approach in PHP.
4. Advanced controls and working mechanism of RDBMS to connect with PHP.
5. Develop a complete end-to-end web applications using PHP.

Text Books:

1. Programming PHP - Creating dynamic web pages, RasmusLerdorf,KevinTatroe and Peter B. MacIntyre, O'Reilly, 2006, ISBN : 978-0-596-00681-5.
2. PHP and MySQL web development, Luke Welling and Laura Thomson, Sams Publishing, 2005, ISBN : 0-672-32672-8.

References:

1. PHP /MySQL Programming for the Absolute Beginner, Andy Harris, Cengage Learning, 2009, ISBN : 978-1-598-63826-4
2. PHP and MySQL Web Development Developer's Library, Luke Welling, Laura Thomson, Pearson Education, 2008, 978-0-768-68643-2
3. Head First PHP & MySQL, Lynn Beighley, Michael Morrison, "O'Reilly Media, Inc.", 2008, ISBN : 978-0-596-00630-3

Mode of Evaluation: Assignments, I Mid Exam, II Mid Exam and End Exam.

MCA II Year – II Semester

16MCA214 WEB PROGRAMMING THROUGH PHP PRACTICAL

	L	T	P	C
Course Prerequisite: None	**0**	**0**	**3**	**2**

Practical Knowledge on HTML tags and scripting language usage.

Course Description:

This practical course helps students to recall the fundamentals of HTML, basic and advanced PHP commands & controls required for web page designing. Basic and advanced concepts of XML and Parser concepts are also introduced for better understanding of parsing. A complete hands on experience will be gained by the students to create end to end web application developed using MySQL as backend with PHP as front end.

Course Objectives:

Student will be able to understand:

1. Write programs using advanced PHP script and develop web pages using advanced controls of PHP.
2. Implement the advanced working mechanism of RDBMS to connect web pages with server.
3. Write script for end to end web application development using PHP and MySQL..

List of Experiments

1. Write a Program to demonstrate simple and advanced HTML tags.
2. Write a program to embed advanced HTML tags in PHP
3. Write a Program to demonstrate advanced commands in PHP
4. Write a program to demonstrate different controls in PHP
5. Write a program to create Database in MySQL.
6. Write a program to demonstrate connecting MySQL Database with PHP.
7. Write a program to store and retrieve information from MySQL through PHP.
8. Write a program to develop a web page using PHP controls.
9. Write a program for developing End to End Web Application using MySQL and PHP.

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. Do basic and advanced programming using PHP, PHP controls.
2. Create databases and set advanced controls & privileges to it.
3. Develop complete end to end web applications using PHP and MySQL.

Mode of Evaluation: Observation, Record, Program Execution and Viva.

MCA II Year – II Semester

16MCA402 WEB TECHNOLOGIES THROUGH JAVA

	L	T	P	C
Course Prerequisite: 14MCA101, 14MCA105	**4**	**0**	**0**	**4**

Course Description:
The course explores and emphasizes the process to create and modify JAVA-based internet applications. It discusses the process of creation of entire web site by introducing the basic structure of HTML, CSS and XML. It guides development of dynamic, database-driven web applications using the JAVA programming languages. This includes Servlets, JSP and database interactions.

Course Objectives:

Students will able to

1. Understand the process to develop dynamic web pages using HTML, CSS, and Form validation and effective creation of XML documents.
2. Acquire knowledge on Networking Sockets.
3. Understand the concepts of Web Server and Servlets.
4. Understand the concepts of JSP.
5. Understand the working mechanism of JDBC.

UNIT- I

INTRODUCTION TO HTML, CSS, JAVASCRIPT AND XML
Hyper Text Markup Language (HTML): Working with Basic and Advanced Tags. **Cascading Style Sheet (CSS):** Introduction to CSS, Types of Style Sheets .**JavaScript:** Introduction - Core features, Functions, Event Handling- Controlling Windows & Frames and Documents - Form handling and validation **XML**: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX.

UNIT- II

NETWORKING
Networking Basics, The Networking Classes and Interfaces, TCP/IP Client Sockets, URL, Http URL Connection, TCP/IP Server Socket, Datagrams.

UNIT-III

WEB SERVER AND SERVLETS
Tomcat web server, Introduction to Servlets: Lifecycle of a Servelet, JSDK, The Servelet API, The javax.servlet Package, Reading Servelet parameters. The javax. Servlet HTTP package, Handling HTTP Request & Responses, Using Cookies, Session Tracking.

UNIT –IV

JAVA SERVER PAGES

JSP Basics: Introduction, Overview, Setting Up the 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

UNIT- V

JDBC

Database Access: Database Programming using JDBC, Studying Javax.sql.* package, Accessing a Database from a JSP Page, Application – Specific Database Actions

Course outcomes:

After completion of the above course the student will be able to

1. Design dynamic web pages using HTML, CSS, form validation and XML documents.
2. Implement Socket programming.
3. Acquire knowledge to implement request and response mechanism using Servlets.
4. Design web page using JSP which controls the content and appearance of the page.
5. Access the database using JDBC and perform actions using JSP.

Text Books:

1. Web Programming, building internet applications,Chris Bates, Second Edition, WILEY Dreamtech.
2. Harvey & Paul Deitel& Associates, Harvey Deitel and Abbey Deitel, “Internet and World WideWeb - How To Program”, Fifth Edition, Pearson Education, 2011
3. The Complete Reference Java ,Seventh Edition by Herbert Schildt,Tata McGraw-Hill Edition
4. Java Server Pages –Hans Bersten,SPD O’Reilly

Mode of Evaluation: Assignment, Written Examination

MCA II Year − II Semester

16MCA215 WEBTECHNOLOGIES THROUGH JAVA PRACTICAL

L	T	P	C
0	0	3	2

Course Prerequisite: Programming knowledge on Java, basics of HTML.

Course Description:
The course guides to completely design a web site with dynamic web pages using HTML, JavaScript and JSP. It also guides to access the database and perform database actions using JDBC, Servlets and JSP.

Course Objectives:

1. Understand the designing of web pages using basic tags, advanced tags, CSS, Java Script and XML.
2. Handle request and response mechanism using Servlets and JSP.
3. Working with database and establishing the connectivity using JDBC.

Hardware and Software requirements:
1. A working computer system with either Windows or Linux.
2. A web browser either IE or Firefox.
3. Tomcat web server and Apache web server.
4. A database either Mysql or Oracle.
5. JVM (Java virtual machine) must be installed on your system.

List of Experiments:
1. Design a web page using basic and advanced HTML tags.
2. Apply CSS and validate the above created page.
3. Create a "registration form" with the following fields
    a. Name (Text field)
    b. Password (password field)
    c. Sex (radio button)
    d. Phone number (text field)
    e. E-mail id (text field)
    f. Phone number (text field)
    g. Address (text area)
4. Validate the fields using JavaScript for the above designed form
    a. Name (Name should contains alphabets and the length should not be less than 6 characters).
    b. Password (Password should not be less than 6 characters length).
    c. E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com)
    d. Phone number (Phone number should contain 10 digits only).
5. Create a XML page to display the Information of the registered users. Use XML schemas, XSL and CSS. You may use xml editors like XML-spy.

6. Install TOMCAT web server and APACHE. While installation assign port number 4040 to TOMCAT and 8080 to APACHE. Make sure that these ports are available i.e., no other process is using this port.
7. Implement client and server socket program for sending receiving messages.
8. Create a Cookie and add these four user id's and passwords to this Cookie. Read the user id's and passwords entered in the Login form 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) Then display a "Welcome note" by the User Name else display "You are not an authenticated user ".Store the user-names and passwords in the webinf.xml and access them in the Servlet by using the getInitParameters() method.
9. Demonstrate simple Java Servlet with Request, Response mechanism.
10. Install a database (Mysql or Oracle).
    Create a table which should contain at least the following fields: name, password, email-id, phone number (these should hold the data from the registration form).Practice 'JDBC' connectivity.
    Write a java servlet to connect to that database and extract data from the tables and display them. Experiment with various SQL queries.
    Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page.
11. Demonstrate simple JSP page with Request, Response mechanism.
12. Write a JSP which does the following job:
    Insert the details of the 3 or 4 users who register with the web site by using registration form. Authenticate the user when he submits the login form using the user name and password from the database (instead of cookies).

Course Outcomes:

After the completion of this course students will be able to

1. Design both static and dynamic web pages along with validations.
2. Implement Servlets and session management.
3. Access the database and perform actions on the database using JSP.

Mode of Evaluation: Practical, Written Examination

MCA II Year−II Semester

16MCA403 USER INTERFACE DESIGN

L	T	P	C
4	0	0	4

Course Description:

This course help students to learn and design the user interfaces using latest technologies. Various control mechanisms, software tools are implemented. Working with interaction devices and online help tutorials are detailed.

Course Objectives:

Students are expected to

1. Identify the human factors of UID, controlled and psychologically oriented experiments
2. Work with latest software tools, commands and natural languages
3. Design and deploy algorithms/ programs for interaction Devices, Balancing fashions and functions, Online Help and Tutorials
4. Work with Multiple window strategies, Information search and visualization, Hyper media and WWW
5. Design UID using available Interface designing technologies

UNIT – I

Human factors of interactive software goals of system engineering and user-interface design, motivations, accommodation of human diversity goal for out profession. Theories, principles, and guidelines – High-level theories, object-action interface model, Principle 1.2 and 3, guide links for data display and data entry, balance of automation and human control. Managing design processes – Usability, design pillars, development methodologies, ethnographic observation, usability testing, surveys, and continuing assessments – expert reviews, usability testing and laboratories, surveys acceptance tests, evaluation during active use, and controlled psychologically oriented experiments.

UNIT – II

Software tolls – Specification methods, interface- building tools and evaluation and critiquing tools. Direct manipulation and virtnal environments – examples, explanations, programming, visual, thinking and icons Home automation, remote direct manipulation, visual environments. Menu selection, form fillin, and dialong boxes – Task – related organizations item presentation sequence, response time and display rate, fact movement through menus, menu layout, form fill in, and dialog boxes. Command and natural languages – Functionality to support users tasks, command – organization strategies, the benefits of structure, naming and abbreviations, command menus, natural language in computing.

UNIT – III

Interaction Devices – Keyboards and function keys, pointing devices, speech recognition digitization and generation. Image and video displays, printers. Response time and display rate-Theoretical foundations, expectations and attitudes, user productivity, variability. Presentation

styles: Balancing function and fashion – error messages, No anthropomorphic design, display design, color, Printed manuals, Online Help and tutorials – Reading from paper versus form displays, preparation of printed manuals, and preparation of online facilities.

UNIT – IV

Multiple – Window strategies – Individual – Window design, multiple-window design, Coordinator by tightly – coupled windows. Image browsing and tightly –coupled windows, personal role management and elastic windows. Computer-supported cooperative work-goals of cooperation, Asynchronous Interaction: Different time and place, Synchronous Distributed: Different place, same time, face to face: same place, same time, Applying CSCW to Edition, Information search and visualization – Database Query and phrase search in textual documents, multimedia document searches, information visualization. Advanced filtering. Hypermedia and the world wide web (www). Genres and goals and designers, users and their tasks, object-action interface model for web site design.

UNIT – V

Introduction to Dot Net technology VB.Net Language – Control structures – GUI controls – Database GUI Controls and its connectivity to databases – ASP.Net Fundamentals and Web pages Interface designing.

Course Outcomes:
Students will able to

1. Identify and work with the human factors of UID, controlled and psychologically oriented experiments
2. Design and work with latest software tools, commands and natural languages
3. Analyze, Design and deploy algorithms/ programs for interaction Devices, Balancing fashions and functions, Online Help and Tutorials
4. Design and work with Multiple window strategies, Information search and visualization, Hyper media and World Wide Web services
5. Design UID using available Interface designing technologies

Text Book:
Ben Shriderman, "Designing the user Interface, strategies for effective human- Computer introduction" Third Edition, Pearson Education, 2004, (For units I, II, III and IV).
Beginning .NET 2.0 by wrox publications (For Unit V).

Reference Books:
1. Hix, Deborah and Hartgon, H.RR X; Developing use Interfaces, John Wiley, 1993.
2. Galitz, Wilbert O., It's Time to Clear Your Windows: Designing GUIs that Work, John Wiley and Sons, New York(1994)

ASP.NET 2.0 Black Book ,Dreamtech publications.
VB.NET 2.0 Black Book, Dreamtech publications.

MCA II Year − II Semester

16MCA216 USER INTERFACE DESIGN PRACTICALS

L	T	P	C
0	0	3	2

Course Objectives:

1. Understand and implement simultaneous equation solver
2. Understand, design and implement simple harmonic motion
3. Understand and design drawing pad and create calculator applications
4. Understand the usage of ADO control
5. Understand and design Digital clock and web browser applications

1. Write a program to calculate Maximum Number from the given numbers
   Description:
   This program allows the user to enter three hidden numbers and the program can calculate the maximum number among the three numbers. In order to allow the user to enter the password in hidden mode, you have to set the PasswordChar property to alphanumeric symbols such as *

2. Write a program in Visual Basic to evaluate Simultaneous Equations Solver
   Description: Simultaneous equations are equations that involves two or more unknown variables. There must be as many equations as the number of unknown variables in order for us to solve the problem.

3. Write a program in Visual Basic to calculate Prime Number
   Description: This program can test whether a number entered by the user is a prime number or not. A prime number is a number that cannot be divided by other numbers other than by itself. Examples are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37 and more.

4. Write a program in Visual Basic to Simple Harmonic Motion
   Description: Simple harmonic motion is the motion of a simple harmonic oscillator. The motion is periodic, as it repeats itself at standard intervals in a specific manner with constant amplitude. It is characterized by its amplitude, its period(the time for a single oscillation), its frequency(the number of cycles per unit time), and its phase(which determines the starting point on the sine wave). The period, and its inverse the frequency, are constants determined by the overall system, while the amplitude and phase are determined by the initial conditions (position and velocity)of that system.

5. Write a program in Visual Basic to Simple Harmonic Motion Boogle
   Description: Boogle is a type of words puzzle game where the players can form as many words as possible from the characters displayed on anxn square. Words can be formed in many ways, from left to right, from right to left, top to bottom, bottom to top, diagonal, zigzag manner and etc.

6. Write a program in Visual Basic to Drawing pad
   Description: the user needs to fill in all the coordinates and selects a color before proceeding to draw the required shape. If the user forgets to fill in the coordinates or selects a color, he or she will be prompted to do so.

7. Write a program in Visual Basic to Calculator
8. Write a program in Visual Basic to using ADO control
9. Write a program in Visual Basic to create digital clock
10. Write a program in Visual Basic to create Web Browser
11. Write a program to calculate Maximum Number from the given numbers
    Description:
    This program allows the user to enter three hidden numbers and the program can calculate the maximum number among the three numbers. In order to allow the user to enter the password in hidden mode, you have to set the PasswordChar property to alphanumeric symbols such as * source: http://www.vbtutor.net/vb_sample/CalMax.htm
12. Write a program in Visual Basic to evaluate Simultaneous Equations Solver
    Description: Simultaneous equations are equations that involves two or more unknown variables. There must be as many equations as the number of unknown variables in order for us to solve the problem.
    Source: http://www.vbtutor.net/vb_sample/simuleq.htm
13. Write a program in Visual Basic to calculate Prime Number
    Description : This program can test whether a number entered by the user is a prime number or not. A prime number is a number that cannot be divided by other numbers other than by itself. Examples are 2, 3, 5, 7, 11, 13,17,19, 23,29, 31, 37 and more.
    Source: http://www.vbtutor.net/vb_sample/Prime.htm
14. Write a program in Visual Basic to Simple Harmonic Motion
    Description: Simple harmonic motion is the motion of a simple harmonic oscillator. The motion is periodic, as it repeats itself at standard intervals in a specific manner with constant amplitude. It is characterized by its amplitude, its period(the time for a single oscillation), its frequency(the number of cycles per unit time), and its phase(which determines the starting point on the sine wave). The period, and its inverse the frequency, are constants determined by the overall system, while the amplitude and phase are determined by the initial conditions (position and velocity)of that system.
    Source: http://www.vbtutor.net/vb_sample/shm.htm
15. Write a program in Visual Basic to Simple Harmonic Motion Boogle
    Description: Boogle is a type of words puzzle game where the players can form as many words as possible from the characters displayed on anxn square. Words can be formed in many ways, from left to right, from right to left, top to bottom, bottom to top, diagonal, zigzag manner and etc.
    Source: http://www.vbtutor.net/vb_sample/boggle.htm
16. Write a program in Visual Basic to Drawing pad
    Description: the user needs to fill in all the coordinates and selects a color before proceeding to draw the required shape. If the user forgets to fill in the coordinates or selects a color, he or she will be prompted to do so.
    http://www.vbtutor.net/vb_sample/drawing.htm
17. Write a program in Visual Basic to Calculator
    Source: http://www.vbtutor.net/vb_sample/bmi_calculator.htm
18. Write a program in Visual Basic to using ADO control
    Source: http://www.vbtutor.net/vb_sample/dbase3.html

19. Write a program in Visual Basic to create digital clock
http://www.vbtutor.net/vb_sample/clock.htm

20. Write a program in Visual Basic to create Web Browser
http://www.vbtutor.net/vb_sample/webbrowser.htm

Course Outcomes:

1. Understand, analyze and implement simultaneous equation solver
2. Understand, design and implement simple harmonic motion using standard technologies
3. Understand and design drawing pad and create calculator applications using standard technologies
4. Understand the usage and applications of ADO control
5. Understand and design Digital clock and web browser applications using standard technologies

MCA II Year – II Semester

16MCA404 MULTIMEDIA SYSTEMS

L	T	P	C
4	0	0	4

Course objectives:

1. Learn and understand the fundamental components of multimedia systems such as audio, video and images.
2. Understand the need of Animations, data compression and various compression techniques.
3. Acquire knowledge on fundamentals of storage mechanism using optical storage media like Video disks, CD-ROM and Multimedia workstations.
4. Understand the services provided by multimedia operating system, the structure and organization of the file system.
5. Get the awareness on Multimedia communication system and multimedia database management system.

UNIT-I

Introduction to Multimedia: media and Data Streams: Medium Main Properties of Multimedia System-Multimedia-Traditional data streams Characteristics-Data streams Characteristics for continuous Media – Information Units-Sound/Audio: Basic Concepts-Computer Image Processing.

UNIT-II

Video and Animation: Basic Concepts-Television-Computer based Animation-Data Compression: Storage Space-Coding Requirements-Source, Entropy and Hybrid coding-some Basic Compression Techniques-JPEGH.261-MPEG_DVI.

UNIT-III

Optical storage media: Basic Technology-Video Disks and other WORMs Compact Disk Read Only Memory-CD-ROM Extended Architecture-Further CD-ROM Technologies-Computer Technology: Communication Architecture-Multimedia Workstation.

UNIT-IV

Multimedia Operating Systems: Real Time-Resource management-Process Management-File Systems-Additional Operating System issues-system Architecture.

UNIT-V

Multimedia Communication Systems: Application Subsystem – Transport Subsystem-Quality subsystem Quality of service and Resource Management-Database Systems: Multimedia Database Management System-Characteristics of MDBMS-Data Analysis – Data Structure*Operations on Data Integration in a Database Model.

Course Outcomes:

Upon completion of the course students will be able to:

1. Describe and differentiate multimedia and non-multimedia systems, text, image, video and audio.
2. Design animations and apply various data compression techniques for data compression.
3. To design, capture, store and integrate sound, images and video to deliver multimodal information.
4. To acquire fundamentals principles of optical storage media, including digitization and data compression for non-textual information.
5. Describe the mechanisms for providing QoS guarantees in the network and to propose experiments to analyse their performance.

Text Book:

Ralf Steinmetz and KlaraNahrstedt, Multimedia: Computing, Communications and Applications, pearson Education Asia.

Reference Books:

1. Tay Vaughan, Multimedia Making it work, Tata McGraw-Hill, Edition, 2001
2. …. Jeffcoate, Multimedia in practice Technology and Application, Prentice Hall, 1995
3. John F. Koeel Buford, Multimedia systems, Addison Wesley, 1994.
4. Fred Halsall, Multimedia communications, Pearson Edition 2001.
5. Prabhat K Andleigh and KiranThatkar, Multimedia systems Design, PHI 2005.

MCA II Year−II Semester

16MCA217 MULTIMEDIA SYSTEMS PRACTICAL

L	T	P	C
0	0	3	2

Course Objectives:

Student will be able to

1. Develop an applications like Photo slideshow and to show boat sailing in water using Macromedia Flash.
2. Design an application for creating an advertisement using Layers, Motion Tween & Shape, tween and also importing an objects from the library and apply zoom in /zoom out effects using Macromedia Flash.
3. Create Publishing Banner Ads and Quiz on C-Language or Java.
4. Develop applications like E-Harathi and create Logo with audio effects.
5. Design applications like effective interactive greeting cards and development of website.

List of Experiments:

1. Design and develop a software for Photo slideshow using Macromedia Flash.
2. Design and develop a software to show boat sailing in water using any Multimedia S/W
3. Design and develop a software for creating an advertisement using Layers, Motion Tween &shape, tween in Macromedia flash.
4. Design and develop a software for importing an object from the library apply the zoom in effect and zoom out effect using Macromedia flash.
5. Design and develop a software for creating a Publishing Banner ads using Macromedia Flash.
6. Design and develop a software for creating a Quiz on C or Java objective questions.
7. Design and develop a software E- Harathi using Macromedia Flash.
8. Design and develop a software to design an Logo with audio effects using macromedia Flash.
9. Design and develop a software an Interactive greeting card using Macromedia Flash.
10. Design and develop a software for a Website using Macromedia Flash.(Minimum of 6 pages)

Course Outcomes:

Upon completion of the course students will be able to:

1. Design and Develop applications like Slideshows and boat animations using Macromedia Flash.
2. Understand and create an advertisements and how to apply zoom in/zoom out effects using Macromedia Flash.
3. Design colorful and Publishing Banner Ads attractively and create quiz questions on C-Language or Java using Macromedia Flash.
4. Design and Development of animation effects for E-Harathi and Logo Creation with audio effects using Macromedia Flash.
5. Develop Greeting cards and effective website using Macromedia Flash.

MCA II Year−II Semester

16MCA405 ADVANCED DATA STRUCTURES

L	T	P	C
4	0	0	4

Course Description:
This course helps students to understand the concepts and applicability of data structures. It gives an in depth working mechanism of various data structure algorithms.

Course Objectives:
Students are expected to
1. Understand the basic and advanced concepts of OOPs and applicability of OOPs in ADTs.
2. Understand the ADT implementation using OOPs.
3. Understand and analyseSorting and traversal algorithms including hashing techniques.
4. Understand the working mechanism of memory allocation, Storage and Garbage allocation related issues.
5. Understand the concepts of notations and their completeness's in practical applications.

UNIT- I

Basic Concepts of OOPs – Templates Function and class templates – Algorithms: performance analysis: time complexity and space complexity– ADT – List (Singly– Doubly and Circular) Implementation – Array – Pointer – Cursor Implementation

UNIT –II

Stacks and Queues – ADT– Implementation and Applications – Trees – General– Binary – Binary Search – Expression Search – AVL – Introduction to Red Black trees and Splay tree – B Trees – Implementations – Tree Traversals

UNIT- III

Set – Implementation – Basic Operations on Set – Priority Queue – Implementation – Graphs – Directed Graphs – Shortest Path Problem – Undirected Graph – Spanning Trees – Graph Traversals: hash table representation: hash functions: collision resolution :separate chaining: open addressing: linear probing: quadratic probing: double hashing: rehashing

UNIT- IV

Issues – Managing Equal Sized Blocks – Garbage Collection Algorithms for Equal Sized Blocks – Storage Allocation for Objects with Mixed Sizes – Buddy Systems – Storage Compaction

UNIT- V

Searching Techniques – Sorting – Internal Sorting – Bubble Sort – Insertion Sort – Quick Sort – Heap Sort – Bin Sort – Radix Sort – External Sorting – Merge Sort – Multiway Merge Sort – Polyphase Sorting – Design Techniques – Divide and Conquer – Dynamic Programming – Greedy Algorithm – Backtracking – Local Search Algorithms.

O-notation, θ-notation, Ω-notation; how to analyze algorithms, Solving recurrence equations, Master theorems, Generating function techniques, Constructive induction , important algorithmic design paradigms like Greedy, dynamic, Divide and Conquer

Polynomial time problem, Non-deterministically Polynomial (NP) Problems, NP-Completeness (Hard Problem) and reducibility :TSP, KNAPSACK, Vertex Cover Problem

Course Outcomes:

Students will able to

1. Apply the basic and advanced concepts of OOPs and applicability of OOPs in ADTs.
2. Implement the ADT implementation using OOPs.
3. Implement and analyze Sorting and traversal algorithms including hashing techniques.
4. Analyze the working mechanism of memory allocation, Storage and Garbage allocation related issues.
5. Understand and apply the concepts of notations and their completeness's in practical applications.

References:

1. Mark Allen Weiss, ―Data Structures and Algorithm Analysis in C++‖, Pearson Education, 2002.
2. AhoHopcroft Ullman, ―Data Structures and Algorithms‖, Pearson Education, 2002.
3. Horowitz Sahni, Rajasekaran, ―Computer Algorithms‖, Galgotia, 2000.
4. Tanenbaum A.S, Langram Y, Augestien M.J., ‖Data Structures using C & C++‖, Prentice Hall of India, 2002.
5. Data structures, Algorithms and Applications in C++, S.Sahni, University Press (India) Pvt.Ltd, 2nd edition, Universities Press Orient Longman Pvt. Ltd.
6. Data structures and Algorithms in C++, Michael T.Goodrich, R.Tamassia and Mount, Wiley student edition, John Wiley and Sons.
7. Data structures using C and C++, Langsam, Augenstein and Tanenbaum, PHI.

MCA II Year–II Semester

16MCA218 ADVANCED DATA STRUCTURES PRACTICAL

L	T	P	C
0	0	3	2

Course Description:

This course helps the students understand the concepts and applicability of available programming paradigm to solve the advance data structure related problems.

Course Objectives:

Students are expected to

1. Understand and find the programming feasibility to solve multi stacks, advanced versions of quees.
2. Understand the usage and applicability of Heaps.
3. Understand the usage and applicability of various non-linear data structures
4. Understand the concepts of sorting procedures
5. Understand and implement greedy algorithms, knapsack & dynamic programming, graph colouring.

List of Experiments:

1. Multistacks
2. Double Ended Queue (Deques) & Circular Queues
3. Min Heap
4. Deaps
5. Leftist Heap
6. AVL Tree
7. B:Tree
8. Quick Sort
9. Greedy algorithm
10. Knapsack using Dynamic Programming
11. Graph coloring using backtracking

Course Outcomes:

Students will able to

1. Identify and apply programming feasibility to solve multi stacks, advanced versions of quees.
2. Find out the solution of Heaps and their applicability.
3. Understand the Identify and applicability of various non-linear data structures
4. Apply the concepts of sorting procedures
5. Implement greedy algorithms, knapsack & dynamic programming, graph colouring.

DISCIPLINE ELECTIVE-II

MCA II Year – II Semester

16MCA406 SOFTWARE TESTING

L	T	P	C
4	0	0	4

Course Prerequisite: Students must have knowledge on Software Engineering concepts and OOAD Concepts.

Course Description:
This course is provides students to have a clear understanding and sound knowledge in foundations, techniques & tools and their usage in the area of software testing in industries. It makes students be competent in software testing domain, whether as a developer or a tester. It also helps students to gain in-depth knowledge in strengths and weakness in software testing techniques.

Course Objectives:

Student will be able to understand:

1. The need of software testing, practices and their applicability in various types of software.
2. The relevance of transaction flow and dataflow in software testing.
3. Classify the domains; apply the testing strategies in domain and interfaces.
4. Identify various paths for data and instruction flow, regular expressions.
5. Get an overview about decision tables, charts, graphs and testing tips.

UNIT- I

BASICS OF SOFTWARE TESTING
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 data flow testing, strategies in data flow testing, application of dataflow testing.

UNIT-III

DOMAINTESTING
Domains and paths, Nice &ugly domains, domain testing, domain and interface testing, domains and testability.

UNIT-IV

PATH AND REGULAR EXPRESSIONS
Paths, Path products and Regular expressions: path products & path expression, reduction procedure, applications, regular expressions &flow anomaly detection.

UNIT –V

LOGICBASED TESTING
Overview, decision tables, path expressions, kvcharts, specifications. State, StateGraphs and Transition testing: state graphs, good &bad state graphs, state testing, Testability tips.

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. Identify and apply suitable testing practices in various software.
2. Draw attention towards transaction flow and data flow in various software.
3. Split the entire software into different domains; apply the testing strategies in domain and their interfaces.
4. Increase the efficiency of software by concentrating on data and instruction flow, regular expressions.
5. Identify the usage of decision tables, charts, graphs and testing tips.

Text Book:
Software testingtechniques-BarisBeizer, Dreamtech, second edition

Reference Books:

1. Software Testing Tools–Dr.K.V.K.K.Prasad, Dreamtech.
2. The craft of software testing- Brian Marick, Pearson Education.
3. SoftwareTesting, 3rdedition, P.C. Jorgensen, AurbachPublications(Dist.bySPD).
4. Software Testing in the Real World–EdwardKit,Pearson.
5. Effective methods of Software Testing, Perry, John Wiley,2nd Edition,1999.
6. Art of Software Testing–Meyers,JohnWiley.
7. Software Testing, N. Chauhan, Oxford University Press.
8. Software Testing, M.G. Limaye,TMH.
9. Software Testing, S. Desikan,G.Ramesh, Pearson.
10. Foundations of Software Testing, D.Graham&Others,Cengage Learning.

Mode of Evaluation: Assignment, Seminar, Written Examination.

MCA II Year – II Semester

16MCA219 SOFTWARE TESTING PRACTICAL

L	T	P	C
0	0	3	2

Course Objectives:

To learn to use the following (or similar) automated testing tools to automatetesting:

1. Win Runner/QTP for functional testing.
2. LoadRunner for Load/Stress testing.
3. Test Director for test management.
4. JUnit, HTML Unit, CPP Unit.

Sample problems on testing:

1. Write programs in 'C' Language to demonstrate the working of the following constructs:
   a. do...while
   b. while….do
   c. if…else
   d. switch
   e. for

2. "A program written in 'C' language for Matrix Multiplication fails" Introspect the causes for its failure and write down the possible reasons for its failure.
3. Take any system (e.g. ATM system) and study its system specifications and report the variousbugs.
4. Write the test cases for any known application (e.g. Banking application)
5. Create a test plan document for any application (e.g. Library Management System)
6. Study of any testing tool (e.g. Win runner)
7. Study of any web testing tool (e.g. Selenium)
8. Study of any bug tracking tool (e.g. Bugzilla, bugbit)
9. Study of any test management tool (e.g. Test Director)
10. Study of any open source-testing tool (e.g. Test Link)
11. Take a mini project (e.g. University admission, Placement Portal) and execute it. During the Lifecycle of the mini project create the various testing documents* and final test report document.

Additional problems on testing:

1. Test the following using JUnit and CPPUnit:
   a. Sorting problems
   b. Searching problems
   c. Finding gcd of two integers
   d. Finding factorial of anumber.
2. Test web based forms using HTML Unit.
3. Test database stored procedures using SQLUnit.

(Use sufficient number of test cases in solving above Problems)

*Note: To create the various testing related documents refer to the text "Effective Software TestingMethodologies by William E. Perry"

Course Outcomes:

Students are expected to solve the following using automated testing tools:

1. Able to understand and apply the working mechanism of Win Runner/QTP for functional testing.
2. Able to understand and apply the working mechanism of LoadRunner for Load/Stress testing.
3. To design and solve the Test Director for test management.
4. To apply the automated testing procedures for JUnit, HTML Unit, CPP Unit

MCA II Year - II Semester

16MCA407 NETWORK SECURITY ESSENTIALS AND STANDARDS

L	T	C
4	0	4

Course Prerequisite: Knowledge on Networking Concepts.

Course Description:

It introduces techniques for protecting information and network components against attacks. Highlights the core cryptographic mechanisms and their implementation procedure to provide security for the data. Investigates various networking security standards and methods for enforcing and enhancing those standards. It also covers the electronic mailing system and supporting protocols.

Course Objectives:

1. To Study Security: Ciphering of plain text , Attacks, Services & Mechanisms
2. Study of various Cryptography security algorithms and encryption methods for security and protection
3. Study key management issues and implementation of hash function to messages
4. To Understand Digital signature and its importance in transaction processing
5. Study on Web security, firewalls and supporting protocols

UNIT-I

INTRODUCTION: Terminology—notation-primeronnetworking-typesofattacks-Layerandcryptograp-Authorization-KeyEscrow-Viruses,wormsandTrojanHorses-MultiLevelmodeofsecurity-legalissues.

UNIT-II

CRYPTOGRAPHY: Introduction-SecretKeycryptography-PublicKeyCryptography-Hash algorithm-DES-IDEA-AES-ModesofOperations-HashesandMessageDigests-MD2-MD4-MD5andSHA-1-RSA-Diffie-Hellamn-DigitalSignatureStandard(DSS)-EllipticCurveCryptography.

UNIT-III

AUTHENTICATION: Password based authentication-address based authentication-Cryptographic authentication Protocols-Passwordsascryptographickeys-trusted Intermediaries-certificaterevocation-MultipletrustedIntermediaries-SessionKey Establishment-Delegation.

UNIT-IV

STANDARDS: KerberosV4-KerberosV5-PublicKeyInfrastructure-RealTime communication Security-IPsec: AHandESP-IPsec:IKE–SSL/TLS

UNIT-V

ELECTRONICMAIL:E-MailSecurity-PEM&S/MIMEandPGP

Course Outcomes:

Student will able to

1. Understand and identify various security fundamentals and standards
2. Understand and analyse the applicability of various keys and their working mechanism
3. Explain the need of authentication and types of authentications.
4. Explain the standards of public key infrastructure and security in real time communications
5. Explain the need and working mechanism of electronic mails.

Textbook:

Network Security Private Communication in a public world, CharlieKaufman, RadiaPerlman&MikeSpeciner,PearsonEducation/PrenticeHallofIndiaPrivate Ltd.,NewDelhi.(Chapters: 1to6, 9,13to22)

Reference Books:

1. NetworkSecurityEssentialsApplicationsandStandards,WilliamStallings,Pearson Education,NewDelhi

2. CryptographyandNetworksecurity,AtulKahate,TataMcGraw-HillPubcompanyLtd.,NewDelhi

MCA II Year - II Semester

16MCA220 NETWORK SECURITY ESENTIAL PRACTICAL

L	T	P	C
0	0	3	2

Course Prerequisite: Knowledge on any high-level programming language.

Course Description:

This course provides technical grounding in network security concepts and algorithms. It deals with the security cryptographic algorithms like AES, DES and Message Digest etc. In essence it helps to the user to secure the confidentiality of the data.

Course Objectives:

Student will be able to understand:

1. Get knowledge on basic constructs of security programs
2. Acquire knowledge on AES, DES and Message Digest algorithms
3. Experiment cryptographic based encryption and decryption algorithms

List of Programs:

1. Find the frequency of occurrence of characters
2. Remove strings and special characters from a given string
3. A study on DES and Triple-DES algorithms
4. Simple program to implement the concept of Encryption and Decryption
5. Compare to stings without using built in function
6. Read a string and split it into two string arrays
7. Write a program to implement RSA algorithm
8. Split a string into two strings and implement encryption and decryption on each
9. Write a program to implement Diffie-Hellman key exchange algorithm
10. Write a program for odd parity and even parity checking process
11. Read a string and print the ASCII value of each character
12. A study on AES algorithm
13. Write a program to implement the concept of message digest
14. Write a program to create a simple login form
15. Pointer based encryption and decryption implementation
16. Write a program for implementing RC4 algorithm.

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. They will be able to write program on basic construct of network security programs
2. Familiar with important cryptographic algorithms like AES, DES etc.,
3. Capable of implementing encryption and decryption in simple login forms

Mode of Evaluation: Observation, Record, Program Execution and Viva.

MCA II Year - II Semester

16MCA408 COMPUTER GRAPHICS

L	T	P	C
4	0	0	4

Course Description:
The course deals withthe fundamentals of visualization and visual communication with techniques in a broad sense and with suitable theoretical examples. It includes survey of graphical representation system, understanding the viewing attribute of 2D and 3D graphical system along with geometrical transformations.

Course Objectives:
Students are expected to
1. Have a survey and understand various graphical representation systems
2. Understand the attributes and viewing system of 2D images
3. Understand 3D and 3D representation graphical system
4. Understand 3D geometrical transformations
5. Analyze the luminous models and their applications

UNIT- I

A survey of computer graphics, overview of graphics systems, output primitives

UNIT- II

Attributes of output primitives, 2-d geometric transformations, 2-d viewing.

UNIT- III

Structures and hierarchical modeling, graphical user interfaces and interactive input methods, 3-d concepts, 3-d object representations.

UNIT –IV

3-D Geometric and modeling transformations, 3d viewing, visible-surface detection methods.

UNIT –V

Illumination models and surface – rendering methods, color models and color applications, computer animation.

Course Outcomes:

Students will able to:

1. Work with the concepts of computer graphics such as geometrical transformations, illumination models, removal of hidden surfaces and rendering
2. Work with the ideas in some fundamental algorithms for computer graphics, compare and evaluate.
3. Apply fundamental principles within interaction programming
4. Understand the fundamental concepts within information and scientific visualization

Text Book:

Donald Hearn and M.Pauline Baker, Computer Graphics C Version, Second Edition, Pearson Educations.2005.

Reference Books:

1. Steven Harrington (1987), Computer Graphics – A Programming Approach, Second Edition, McGraw – Hill International Editions.

2. William M. Newman and Robert F. Sprowli (1979), Principles of Interactive Computer Graphics, second Edition, McGraw – Hill International Editions.

3. FS Hill Jr. Cmputer Graphics using Open Gl, second Editions, 2005.

4. J.D.Foley Wesley,199, second Edition in C.

5. R.C.S Asthana and N.K.Sinha "Computer Graphics for Scientists and Engineers" New Age International Limited, Second Revised Edition.

MCA II Year II Semester

16MCA221 COMPUTER GRAPHICS PRACTICAL

L	T	P	C
0	0	3	2

Course Objectives:

1. Understand and use the basics of graphics algorithms
2. Understand and implement graphical transformations
3. Understand and analyze 2D and 3D graphical applications / transformations
4. Understand and apply the animation using / for standard input devices
5. Work on 2D / 3D image processing.

List of Experiments:

1. Design and develop software in Java to draw Ellipse using different algorithms
2. Design and develop software in Java to show 3 D Translations on Applet.
3. Design and develop software in Java to implement Cohen Sutherland Clipping Algorithm
4. Design and develop software in Java to show image in various Shading or blur using filters.
5. Design and develop software in Java to show Bezier curves.
6. Design and develop software in Java to display bar chart for a given student data in Array fill bars in chart using various filling algorithms.
7. Design and develop software in Java to Rotate the text by a given angle from a keyboard.
8. Design and develop software in Java to display a Wall Clock
9. Design and develop software in Java to display the text for given Font and Size from keyboard
10. Design and develop software in Java to graphically view the Solution for travelling sales man problem for a given data using Java 2D API

Course Outcomes:

Students will able to

1. Write program to demonstrate the JAVA programming skills implement graphics primitives.
2. Write programs that demonstrate JAVA programming skills to show geometrical transformations.
3. Demonstrate an understanding of the implementation of 2D and 3D in graphics applications.
4. Write JAVA program functions to implement visibility detection.
5. Write programs that demonstrate computer graphics animation for standard input devices.
6. Write programs that demonstrate 2D image processing using JAVA

MCA II Year II Semester

16MCA409 PARALLEL COMPUTING

L	T	P	C
4	0	0	4

Course Objectives:

1. Understand the fundamental of CPU and GPU technologies and their working mechanism
2. Understand the working procedure of Serial and Parallel algorithms
3. Understand the techniques and concepts of GPU optimization techniques
4. Understand and analyze the Parallel computing and optimization patterns
5. Understand the working mechanism of Libraries and dynamic parallelism

UNIT – I

Introduction and the GPU ProgrammingTechnology Trend: CPU programming, Latency, Bandwidth GPU Programming: Design Goals, Kernel, Map CPU vs GPU: Squaring a number
GPU Hardware and Parallel Communication PatternsCommunication Patterns: Map, Gather, Scatter, Stencil, TransposeGPU Hardware: Streaming Multiprocessors, Kernel, Thread Blocks, Threads GPU Memory Model: Synchronization, Barrier, Memory access, Coalesce, AtomicsStrategies for efficient CUDA programming

UNIT – II

Fundamental Parallel Algorithms Step complexity, Work complexityReduce: Serial vs Parallel Implementation, Global and Shared Memory BandwidthScan: Serial vs Parallel Implementation, Inclusive vs Exclusive Scan, Hillis Steele vsBlelloch Scan Histogram: Serial vs Parallel Implementation, Atomics, Local Memory, Reduction Fundamental Parallel Algorithms Compact: Core Algorithm, ProcedureAllocate: StrategySegmented Scan: SpMV, CSRSort: Brick Sort, Merge Sort, Sorting Networks, Radix Sort, Quicksort

UNIT – III

Optimizing GPU Programs Levels of Optimization: APODAnalyze: Hotspots, Amdahl's Law Parallelize: Matrix Transpose, Bandwidth, Tiling, NVVP, Little's Law, OccupancyThread Divergence: Warp, SIMD, SIMT, SwitchCPU-GPU Interaction: Streams

UNIT- IV

Parallel Computing Patterns: Dense N-Body: Tiling, SpMV, Thread per Row/Thread per Element, Traversal of Graph: Depth-first vs Breadth-first searchGraph Data StructureList Ranking: Merrills Linear ComplexityHashTableThe Frontiers and Future of GPU Computing Parallel Optimization Patterns: Data Layout transformation, Scatter, Tiling, Privatization, Binning, Compaction, Regularization

UNIT- V

Libraries: cuBLASCUDA C++ Programming Power Tools: Thrust, CUB, cudaDMA
Other Languages: PyCUDA, MATLAB
Dynamic Parallelism: Bulk, Nested, Task, Recursive, Quicksort

Course Outcomes:

1. Explain the CPU and GPU technologies and their working mechanism
2. Analyze and distinguish the working procedure of Serial and Parallel algorithms
3. Understand and solve the GPU optimization techniques for efficiency
4. Understand and analyze the Parallel computing and optimization patterns
5. Understand the working mechanism of Libraries and dynamic parallelism in real world scenarios.

MCA II Year - II Semester

16MCA410 PATTERN RECOGNITION

L	T	P	C
4	0	0	4

Course Objectives:

1. To learn about the concept of pattern recognition through various approaches
2. To study about the clustering concepts and the usage in pattern recognition
3. To know about the feature extraction concepts and structural pattern recognition
4. To understand the concepts of about Artificial Neural Networks and their usage in pattern recognition
5. To get an idea about the recent advances in pattern recognition and their usage in major application areas

UNIT – I

PATTERN CLASSIFIER

Overview of Pattern Recognition – Discriminant Functions – Supervised Learning –Parametric estimation – Maximum likelihood estimation. – Bayesian parameter estimation – Problems with Bayesian Approach – Pattern classification by distance functions – Minimum distance pattern classifier

UNIT – II

CLUSTERING

Clustering for supervised learning – Clustering for classification – Clustering Concept – C Means Algorithm – Hierarchical Clustering – Graph Theoretic approach to pattern clustering – Cluster Validity

UNIT – III

FEATURE EXTRACTION AND STRUCTURAL PATTERN RECOGNITION

Feature Selection through functional approximation – Binary Selection - Elements of Formal Grammars – Syntactic Description – Stochastic Grammars and applications – Graph Based Structural Representation

UNIT – IV

NEURAL PATTERN RECOGNITION

Introduction to Neural Networks – Feedforward Networks and training by Back Propagation – Neural Network Structure for Pattern Recognition – Neural Network based pattern association – Unsupervised Learning in neural pattern recognition

UNIT- V

RECENT ADVANCES AND APPLICATIONS
Fuzzy Pattern Classifiers – Pattern Classification Using Genetic Algorithms – Medical Applications – Image Processing – Financial applications

Course Outcomes:

Upon completion of the course the student can able to

1. Apply the learning strategies in pattern recognition
2. Apply clustering techniques in pattern recognition
3. Identify and analyze the usage of grammars in pattern recognition
4. Analyze the Artificial Neural Networks suitable for pattern analysis
5. Make use of the fuzzy techniques for proper patterns and analyze the relevant application areas.

References:

1. Robert Schalkoff, "Pattern Recognition statistical, structural and neural approaches", JohnWiley& Sons Inc. 2007
2. M.Narasimha Murthy and V.Susheela Devi, "Pattern Recognition", Springer 2011
3. R.O.Duda.P.E Hart & D.G Stork, "Pattern Classification", 2nd Edition J.WileyInc 2001
4. Christoper M Bishop, "Neural Network for pattern Recognition", Oxford University press 2008
5. C.M.Bishop, "Pattern Recognition and Machine Learning", Springer 2006
6. Geoff Dougherty, "Pattern Recognition and classification: An Introduction", springer 2013

DISCIPLINE ELECTIVE-III

MCA III Year – I Semester

16MCA411 BIG DATA AND ANALYTICS

L	T	P	C
4	0	0	4

Course Prerequisite: Student should have at least basic knowledge of Java and Linux.

Course Description:

With the advance of IT storage, processing, computation, and sensing technologies, Big Data has become a novel norm of life. Almost all industries are bracing into the challenge of Big Data and want to dig out valuable information to get insight to solve their challenges.

This course shall provide the fundamental knowledge to equip students being able to handle those challenges. This Big Data Analytics course shall first introduce the Big Data Platform, Intelligent data analysis, Analytic Processes and Tools. Then will guide through the Hadoop platform and its environment and applications on Big Data Using Pig and Hive, Data processing operators in Pig and Hive services. Afterwards, the course will zoom in to introduce Visualizations, Visual data analysis techniques and interaction techniques.

Course Objectives:

1. To explore the fundamental concepts of big data analytics
2. To learn to analyze the big data using intelligent techniques.
3. To understand Hadoop, MongoDB, NoSql, Map Reduce Concepts.
4. To learn to use various techniques for mining data stream.
5. To understand Pig &HiveQL.

UNIT-I

Introduction to Big Data: Characteristics of Data, Evolution of Big Data, Definition of Big Data, Challenges with Big Data, What is Big Data?, Why Big Data? , Traditional Business Intelligence (BI) versus Big Data , A Typical Data Warehouse Environment, A Typical Hadoop Environment, What is New Today?, What is changing in the Realms of Big Data?.

Big Data Analytics: What is Big Data Analytics?, What Big Data Analytics Isn't?, Why this Sudden Hype Around Big Data Analytics? , Classification of Analytics, Greatest Challenges that Prevent Businesses from Capitalizing on Big Data, Top Challenges Facing Big Data, Why is Big Data Analytics Important?.

UNIT-II

Introduction to Hadoop: Why Hadoop?, Why not RDBMS? , RDBMS versus Hadoop, History of Hadoop, Hadoop Overview, HDFS (Hadoop Distributed File System), Processing Data with Hadoop, Interacting with Hadoop Ecosystem.

Introduction to MongoDB: What is MongoDB?, Why MongoDB?, Terms Used in RDBMS and MongoDB, Data Types in MongoDB, MongoDB Query Language.

UNIT-III

What is NoSQL?: NoSQL Business Drivers**,** NoSQL Data Architectural Patterns**,** Variations of NoSQL Architectural Patterns, Using NoSQL to Manage Big Data.

MapReduce: MapReduce and The New Software Stack, Mapper, Reducer, Combiner, Partitioner, Searching, Sorting, Compression.

UNIT-IV

Mining Data Streams: Data Stream Management Systems, Data Stream Mining, Examples of Data Stream Applications, Stream Queries, Issues in Data Stream Query Processing, Sampling in Data Streams, Filtering Streams, Counting Distinct Elements in a Stream, Querying on Windows − Counting Ones in a Window, Decaying Windows.**Introduction to Machine Learning:** Introduction to Machine Learning, Machine Learning Algorithms.

UNIT-V

Introduction to Hive: What is Hive? , Hive Architecture, Hive Data Types, Hive File Format, Hive Query Language (HQL), User-Defined Function (UDF).
Introduction to Pig: What is Pig? , The Anatomy of Pig, Pig on Hadoop, Data Types in Pig, HDFS Commands, Relational Operators, Complex Data Types, Piggy Bank, User-Defined Functions (UDF), Pig versus Hive.

Course Outcomes:
The students will be able to:
1. Work with big data platform
2. Analyze the big data analytic techniques for useful business applications.
3. Design efficient algorithms for mining the data from large volumes.
4. Analyze the HADOOP and Map Reduce technologies associated with big data analytics
5. Explore on Big Data applications Using Pig and Hive

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.Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos,

References:

1. Seema Acharya, SubhashiniChallappan, “Big Data Analytics” by Wiley Publishing.
2. Radha S, M. Vijayalakshmi, “Big Data Analytics” by Wiley Publishing.
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
5. PeteWarden, “Big Data Glossary”, O’Reilly, 2011.
6. Jiawei Han, MichelineKamber “Data Mining Concepts and Techniques”, Second Edition, Elsevier, Reprinted 2008.
7. “Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data”,
    a. McGrawHill Publishing, 2012
8. AnandRajaraman and Jeffrey David Ullman, “Mining of Massive Datasets”, Cambridge
    a. University Press, 2012.

Mode of Evaluation: Assignment, Seminar, Written Examination.

MCA III Year – I SEM

16MCA224 BIG DATA AND ANALYTICS PRACTICAL

	L	T	P	C
Course Prerequisites:	**0**	**0**	**3**	**2**

Student should have at least basic knowledge of Java and Linux.

Course Description:

Big Data Analytics LAB course shall first introduce the Big Data Platform, help student to explore intelligent data analysis with available tools. Then the course guide through the Hadoop platform and its environment and applications on Big Data Using Pig and Hive, Data processing operators in Pig and Hive services. The course also gives an understanding on NoSQL databases, and will zoom in to introduce Visualizations, Visual data analysis techniques and interaction techniques.

Course Objectives:

1. To explore the Big Data Platform and its environment
2. To learn the basics of Hadoop Environment, HDFS and related topics
3. To understand the various search methods and visualization techniques.
4. To learn to use NoSQL, MongoDB.
5. To understand the basics concepts of MapReduce and do hands-on

List of Experiments

T**erm work:**

Assign a case study for group of 2/3 students and each group to perform the following experiments on their case-study; each group should perform the exercises on a large dataset created by them.
Suggested Practical List:
Students will perform at least 8 programming exercises and implement one mini-project. The students can work in groups of 2/3.

1. Study of Hadoop ecosystem
2. programming exercises on Hadoop
3. programming exercises in MongoDB
4. programming exercises in NoSQL
5. Implementing simple algorithms in Map- Reduce (3) - Matrix multiplication, Aggregates, joins, sorting, searching etc.
6. Implementing any one Frequent Itemset algorithm using Map-Reduce
7. Implementing any one Clustering algorithm using Map-Reduce
8. Implementing any one data streaming algorithm using Map-Reduce
9. Pig
   Pig installation
   PIG SCRIPTS:- Study of the language Pig Latin &Pig's Grunt shell in hadoop (mapreduce) mode, Data Types –Functions (Eval, Load/Store, String, DateTime)
10. Hive Hive Installation – Study of Hive QL – Manipulation with Hive tables.
11. Mini Project: One real life large data application to be implemented (Use standard Datasets available on the web)
    i. Twitter data analysis
    ii. Fraud Detection
    iii. Text Mining etc.

Course Outcomes:

The course will help students in understanding a variety of skills, tools and techniques to understand data, examine business problems and bring about key business solutions in a structured manner. Some of these include:

1. Concepts of Hadoop Distributed File System and MapReduce framework
2. Setting up a Hadoop Cluster
3. Program in MapReduce,
4. Writing Complex Queries NoSql, MongoDB, HiveQL.
5. Performing Data Analytics using Pig and Hive

MCA III Year – I Semester

16MCA412 WIRELESS SENSOR NETWORKS

L	T	P	C
4	0	0	4

Course Description:
This course provides the fundamentals of sensor and wireless sensor networks. Various styles of architectures and their working mechanism is included. Concepts and working mechanism of protocol issues are also included. Infrastructure, Platforms and Tools are also discussed.

Course Objectives:
1. Able to learn the fundamentals of sensor and wireless sensor networks
2. Explore the applicability sensors in different architectures.
3. Work on mechanism and scheme of protocols.
4. Able to understand the various network establishment procedures
5. Work with different tools, suitable for various environments.

UNIT- I

OVERVIEW OF WIRELESS SENSOR NETWORKS
Challenges for Wireless Sensor Networks, Enabling Technologies For Wireless Sensor Networks.

UNIT –II

ARCHITECTURES
Single-Node Architecture - Hardware Components, Energy Consumption of Sensor Nodes , Operating Systems and Execution Environments, Network Architecture - Sensor Network Scenarios, Optimization Goals and Figures of Merit, Gateway Concepts.

UNIT-III

NETWORKING SENSORS
Physical Layer and Transceiver Design Considerations, MAC Protocols for Wireless Sensor Networks, Low Duty Cycle Protocols And Wakeup Concepts - S-MAC , The Mediation Device Protocol, Wakeup Radio Concepts, Address and Name Management, Assignment of MAC Addresses, Routing Protocols- Energy-Efficient Routing, Geographic Routing.

UNIT –IV

INFRASTRUCTURE ESTABLISHMENT
Topology Control, Clustering, Time Synchronization, Localization and Positioning, Sensor Tasking and Control.

UNIT-V

SENSOR NETWORK PLATFORMS AND TOOLS
Sensor Node Hardware – Berkeley Motes, Programming Challenges, Node-level software platforms, Node-level Simulators, State-centric programming.

Course Outcomes:

1. Able to understand and explain fundamentals of sensor and wireless sensor networks
2. Apply wireless sensors in different architectures.
3. Design and apply the different mechanism and scheme of protocols.
4. Able to design and apply procedures suitable for wireless sensor networks
5. To work with different tools, suitable for various sensor networks .

Text Books:

1. Holger Karl & Andreas Willig, “Protocolsand Architectures for Wireless Sensor Networks" , John Wiley, 2005.

2. Feng Zhao & Leonidas J. Guibas, “Wireless Sensor Networks- An Information Processing Approach", Elsevier, 2007.

References:

1. KazemSohraby, Daniel Minoli, &TaiebZnati, “Wireless Sensor Networks- Technology, Protocols, And Applications”, John Wiley, 2007.
2. Anna Hac, “Wireless Sensor Network Designs”, John Wiley, 2003.

MCA III Year-I Semester

16MCA413 MOBILE APPLICATION DEVELOPMENT USING ANDROID

L	T	P	C
4	0	0	4

Course Prerequisite: Java Programming

Course Description:
This course is concerned with the development of applications for android platform. Android will be used as a basis for teaching programming techniques related to the development of standalone applications. This course starts with reviewing java, history of android and architecture. It introduces the major building blocks of android in detail with the help of example project. It also covers the development of applications using widgets, events, networking concepts and sensors.

Course Objectives:

Student will be able to understand:

1. The history, fundaments of android platform and java overview.
2. Android user interfaces and major building blocks.
3. About Fragments, Intents and ActionBar in Android Platform.
4. Usage of Lists, Adapters and Broadcast receivers
5. How to use widgets, networking concepts and sensors in android based applications

UNIT- I

JAVA REVIEW & INTRODUCTION AND INSTALLATION OF ANDROID TOOLS
Java Review: OOPs Concepts, Inheritance in detail, Exception Handling, Interfaces and, packages, JVM & .jar file extension, Multi-Threading (Thread class & Runnable Interface)
Android Overview: What is Android and Android Overview, History, Android Versions, Android Flavors.
The Stack: Stack Overview, Linux, Native Layer, Dalvik, Application Framework, Applications.
Installing and Beginning Use of Android Tools: Installing JDK, Installing the Android SDK, Hello World Example, Anatomy of an Android Project, Drawable Resources, Building the Project, Android Emulator.

UNIT- II

BUILDING BLOCK OF ANDROID AND ANDROID USER INTERFACE
Main Building Blocks: A Real-World Example, Activities, Intents, Services, Content Providers, Application Context, and Example by Yamba Project overview.
Android User Interface: Two ways to create a User Interface, Views and Layout, Starting the Yamba Project, The Status Activity Layout, The Status Activity Java Class, Logging Messages in Android, Threading in Android.

UNIT- III

FRAGMENTS, INTENTS, ACTION BAR, SERVICES AND CONTENT PROVIDERS
Fragments: Fragment Example, Fragment Life cycle, Dynamically Adding Fragments.
Intents, Action Bar: Preferences the Action Bar, Shared Preferences and Updating Status Fragment, The File system explained.
Services: Refresh Service, Pulling Data from Yamba, Testing the Service.
Content Providers: Databases on Android, Status Contract Class, Update Refresh Service, Content Provider.

UNIT- IV

LIST AND ADAPTERS & BROADCAST RECEIVERS
Lists and Adapters: Main Activity, Basic Main Activity, Timeline Fragment, About Adapters, Loading the Data, Custom Logic via View Binder, Details View.
Broadcast Receivers: About Broadcast Receivers, Boot Receiver, Alarms and system services, Broadcasting Intents.

UNIT-V

APPLICATIONS WIDGETS, NETWORKING, WEB OVERVIEW, INTERACTION AND SENSORS
App Widgets: Using Content Providers through Widgets
Networking and Web Overview: Quick Example, Networking Basics, HTTP API, Apache HTTP Client, Http URL Connection, Networking in Background using AsyncTask and Async Task Loader.
Interaction and Animation: Live Wallpaper and Handler
Sensors: Sensor API, Motion Sensor, Position Sensor, Environmental Sensor, Sensor Values, Sensor Manager Class, Sensor Class, Sensor Event class, Sensor Event Listener interface, Compass Accelerometer and orientation Sensors, Sensor Examples.

Course Outcomes

Upon successful completion of this course, students will be able to:

1. Recollect the essential concepts of java; get familiar with android basics and installation.
2. Create User Interfaces with various Layouts and views.
3. Know how fragments, intents and Actionbars are used in designing applications.
4. Understand the applicability of lists, adapters and broadcast receivers.
5. Use widgets, networking concepts and sensors in developing android applications.

Text Books:
1. Learning Android , By Marko Gargenta& Masumi Nakamura, O'Reilly, II Edition
2. Programming Android,ByZigurdMednieks,LairdDornin,G.BlakeMeike& Masumi Nakamura, O'Reilly.

Mode of Evaluation: Assignment, Seminar, Written Examination

MCA III Year-I Semester

16MCA226 MOBILE APPLICATION DEVELOPMENT PRACTICAL USING ANDROID

L	T	P	C
0	0	3	2

Course Prerequisites:
Programming knowledge on Java & OOPs Concepts and Basics of Android and its concepts.

Course Description:
This course introduces the android application development. Emphasis is placed on the processes, tools and frameworks required to develop applications for current and emerging mobile computing devices. It covers the core components, android interface library, content providers, broadcast intents and various sesnsors.

Course Objectives:

Student will be able to understand

1. Basic development tools for android application development.
2. Get the knowledge on core components of user interface, taking input methods, date and time pickers and android interface library.
3. Gain knowledge on how to use content providers, service & broadcast intents and usage of various sensors

LIST OF EXPERIMENTS

1. Installation of Android Software Development Kit (SDK) and related Software needed to run Android Applications.
2. Create a Simple Android Application that displays "**Hello World"** message.
3. Develop an android Application that uses Core Components of Android User Interface Buttons, Labels, and Checkboxes.
4. Develop an android application that uses the Android Input Method Framework getting input from attached and/or internal keyboards
5. Develop an android application that performs arithmetic operations like addition, subtraction, multiplication and division.
6. Develop an android application to add Icons, SelectionLists, Sliders, Checkboxes, Date and Time Pickers.
7. Develop an Android Application that demonstrates how to use and extend the Android User interface library like java Swing Library.
8. Develop an Android Application that demonstrate how to use Android user interface Library for creating Custom Views , Options and Context Menus, Adapters and Adapter Views
9. Develop an Android Application that suggests users to download other Applications by posting notifications in the Notifications Bar by using Content Providers, Services and broadcast Intents.
10. Demonstrate usage of Android Sensors
11. Motion Sensor,

12. Environmental Sensor
13. Position sensor.

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. Use the development tools and supported plug-ins that involved in application development process.
2. Develop android applications using core components, layouts and widgets.
3. Use content providers, service & broadcast intents and sensors in developing applications.

Mode of Evaluation: Observation, Record, Program Execution and Viva.

MCA III Year-I Semester

16MCA414 KNOWLEDGE BASED SYSTEMS &DESIGN

L	T	P	C
4	0	0	4

Course Description:
This course provides an overview of artificial intelligence and information systems. Dealing with uncertainty models, system and user interfaces are detailed along with testing, validation and verification techniques. Decision making and mining related techniques are included. Concepts and working mechanism related to an expert system building is detailed.

Course Objectives:
1. To understand the artificial intelligence and information related to concepts.
2. To get clarity about various uncertainty models, interfaces, verification and validation models.
3. To understand working mechanism of decision making trees and various miming applications in artificial intelligence.
4. To explore the various methods for developing the knowledge based systems.
5. To understand and deal with expert systems.

UNIT –I

Introduction: Artificial Intelligence and information systems Fourth & fifth generation languages, Nonprocedural Paradigms, Data and Knowledge AI, knowledge based systems, Expert Systems Basic architecture of knowledge based systems Knowledge representation and the knowledge base First-Order Logic, Production Rules, Horn Clauses, Frames, Semantic Networks, Objects Meta knowledge, Conceptual modelling Inference and reasoning State space representations and search Chaining methods, resolution, inference, matching, conflict resolution

UNIT – II

Dealing with uncertainty Bayesian models, Dempster-Shafer, Certainty Factors, Fuzzy sets and systems Interfaces User interface: explanation facilities, unknown values Systems interface: language and database hooks Developer interface: knowledge acquisition, testing & debugging Verification and validation Redundancies, Conflicts, Contradictions, Incompleteness, Invalidity

UNIT – III

Induction of decision trees and data mining Learning from examples, entropy, ID3 Artificial neural networks Basic components, learning modes and laws, network types, applications;

UNIT – IV

Methodologies for building knowledge based systems Development lifecycle, structured development and prototyping Knowledge acquisition techniques, protocol analysis, repertory grid Integration with databases, data processing and information systems methodologies;

UNIT – V

Expert system building tools AI-Languages, Knowledge representation languages, E.S.-shells, products and environments Knowledge base management systems Organizational and managerial issues Knowledge as a strategic asset in the organization, knowledge problems and management Applications, pitfalls and successes.

Course Outcomes:

1. To explain the artificial intelligence and information related to concepts.
2. To explain various uncertainty models, interfaces, verification and validation models.
3. To work with decision making trees and various miming applications in artificial intelligence.
4. To explore the applicability if various methods for developing the knowledge based systems.
5. To build and apply the expert systems in various applications.

MCA III Year-I Semester

16MCA415 SEMANTIC WEB ARCHITECTURE

L	T	P	C
4	0	0	4

Course Description:
This course gives fundamental knowledge about various types web services, web architectures. In depth knowledge on how the data vectors are designed and working procedure of spatial data bases is included along with creating the web services on demand.

Course Objectives :
1. To understand the working procedure of web services.
2. To get the knowledge on semantic web architectures.
3. To understand the data types, vectors and formats.
4. To understand the need and working mechanism of spatial databases.
5. To create sematic web services with full functionalities.

UNIT –I

Introduction to web services: Fundamentals of XML, XML Syntax, XML Document Structure, Schemalanguages-DTD,XML Schema, Presentation technologies – XSL – XFORMS – XHTML – Transformation –XSLT – XLINK – XPATH – Xquery. Developing Web services-Objectives, Web service standards ,SOAP-TheProcessing model, Faults, Data representation and RPC, Protocol binding, WSDL-Interface Descriptions,Binding description, service description, UDDI-Descriptions ,Discovery.

UNIT-II

Semantic Web-Architecture of Semantic Web-Components of Semantic Web stack – ResourceDescription Framework-RDF Schema-Web Ontology language-Rule Interchange Format-Semantic webRule Language-SPARQL-Knowledge representation and reasoning –Ontology-Ontology components.

UNIT-III

Vectors-Raw materials-Raster Data-Vector Data-Types of Vector Data-know your file formats-Anatomyof a shape file-Downloading a viewer-Styling your layers-saving your map in ArcExplorer-Projections-TheRound Earth-Cartesian Planes-Coordinate Reference systems-Reprojection utilities-Rasters-Mosaics andTessellation-Temporal analysis-Panchromatic Vs multispectral-Scale and Resolution-Orthorectification-Downloading Free Rasters.

UNIT-IV

Spatial Databases-Why bother with a spatial Database-Installing Postgre SQL and PostGIS-Adding spatialFields-Inserting spatial Data-Querying spatial Data-Introspection of spatial Data-Importing spatial Data-Manipulating Data-Exporting Data-Indexing Data-Spatial Queries-Visualising Data

UNIT-V

Creating OGC Webservices-Sharing the wealth-OGC SOA for GIS-Installing Geoserver-Adding shapefilesUsing the GUI-Adding Shapefiles manually-Adding PostGIS Layers-Styling with SLD-Using OGC Webservices-Understanding WMS-WMS Get Capabilities-WMS Get MAP-Understanding WFS-WFSGetCapabilities-WFS DescribeFeatureType-Filtering WFS GetFeature Requests.

Course Outcomes:

1. To understand and explain the working procedure of web services.
2. To exhibit the knowledge on semantic web architectures.
3. To explain the various data types, vectors, formats and their applicability.
4. To explain the need and working mechanism of spatial databases.
5. To demonstrate the sematic web services with full functionalities as per demand.

Text Books:

1. 1.Xml and WebservicesUnleashed,RonSchmelzer,TravisVandersypen,Madhusiddalingaiah,Dianekennedy,Pearson Edition,2011,ISBN 978-81-317-1869-8.
2. GIS For WebDevelopers by Scott Davis , ISBN-13: 978-0-9745140-9-3

References:

1. Foundations of the Semantic Web: XML, RDF & Ontology by RajendraAkerkar, ISBN-13-9788173199851.
2. An Introduction to XML and Web Technologies by Anders Moller,MichaelSchwartzbach,PearsonEdition, ISBN 978-81-317-2607-5.

DISCIPLINE ELECTIVE-IV

MCA III Year-I Semester

16MCA416 DISTRIBUTED OPERATING SYSTEM

L	T	P	C
4	0	0	4

Course Description:
This course provides an in-depth knowledge of distributed operating systems. Working mechanism of synchronization with process and process controlling system is explained. Distributed file systems in distributed operating systems is included with a complete case study on the course.

Course Objectives:
1. To understand the difference and fundamentals of distributed operating systems.
2. To gain the knowledge on the concepts of synchronization.
3. To understand the working mechanism of process and process synchronization.
4. To understand the files and their sharing in distributed manner.
5. To work with a case study.

UNIT-I

INTRODUCTION
Introduction to Distributed System, Goals of Distributed system, Hardware and Software Concepts, Design Issues, Communication in distributed system: Layered protocols, ATM networks, Client – Server model, Remote Procedure Calls and Group Communication. Middleware and Distributed Operating Systems.

UNIT-II

SYNCHRONIZATION IN DISTRIBUTED SYSTEM
Clock synchronization, Mutual Exclusion, Election algorithm, the Bully algorithm, a Ring algorithm, Atomic Transactions, Deadlock in Distributed Systems, Distributed Deadlock Prevention, and Distributed Deadlock Detection.

UNIT-III

PROCESSES AND PROCESSORS IN DISTRIBUTED SYSTEMS
Threads, System models, Processors Allocation, Scheduling in Distributed System, Real Time Distributed Systems.

UNIT-IV

DISTRIBUTED FILE SYSTEMS
Distributed file system Design, Distributed file system Implementation, Trends in Distributed file systems. Distributed Shared Memory: What is shared memory, Consistency models, Page based distributed shared memory, shared variables distributed shared memory.

UNIT-V

CASE STUDY MACH

Introduction to MACH, process management in MACH, communication in MACH, UNIX emulation in MACH.

Course Objectives:

1. To differentiate the working procedure of distributed operating systems.
2. To explain and solve the issues related to synchronization.
3. To work with processes and process synchronization in virtual manner.
4. To understand and explain concepts of files and their sharing in distributed manner.
5. To exhibit the knowledge to work on a case study.

Text Book:

Distributed Operating System – Andrew S. Tanenbaum, PHI.

MCA III Year-I Semester

16MCA417 SERVICE ORIENTED ARCHITECTURE

L	T	P	C
4	0	0	4

Course Description:
Principles and roots of service oriented architecture are detailed in first unit of the course. Services and layer of SOA is included along with the analysis & design related issues. Platform basis and issues related services are included in last units of the course.

Course Objectives:
1. To understand the roots and principles, characteristics of SOA.
2. To get knowledge on web services, descriptions and layers.
3. To identify the issues related to service oriented analysis and their design patterns.
4. To get knowledge on platform basics.
5. To understand the services related to complications in SOA.

UNIT- I

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.

UNIT- II

Web services – Service descriptions – Messaging with SOAP –Message exchange Patterns – Coordination –Atomic Transactions – Business activities – Orchestration – Choreography - Service layer abstraction – Application Service Layer – Business Service Layer – Orchestration Service Layer

UNIT- III

Service oriented analysis – Business-centric SOA – Deriving business services- service modelling - Service Oriented Design – WSDL basics – SOAP basics – SOA composition guidelines – Entity-centric business service design – Application service design – Task- centric business service design

UNIT- IV

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)- Web Services Interoperability Technologies (WSIT) - SOA support in .NET – Common Language Runtime - ASP.NET web forms – ASP.NET web services – Web Services Enhancements (WSE)

UNIT- V

WS-BPEL basics – WS-Coordination overview - WS-Choreography, WS-Policy, WS- Security

Course Outcomes:

1. To understand and get exposure on the roots and principles, characteristics of SOA.
2. To get knowledge and apply the same on web services, descriptions and layers.
3. To identify and solve the issues related to service oriented analysis and their design patterns.
4. To get knowledge on platform basics, working with various platforms.
5. To understand and exhibit the knowledge on the services related to complications in SOA.

Textbook:

Thomas Erl, "Service-Oriented Architecture: Concepts, Technology, and Design", Pearson Education, 2005.

MCA III Year – I Semester

16MCA418 CLOUD INFRASTRUCTURE & SERVICES

L	T	P	C
4	0	0	4

Course Description:

This course first describe what is Cloud and the various service delivery models of a Cloud computing architecture, followed by the Cloud management, Performance issues, Hard ware concepts, Storage, Security and Privacy issues .

Course Objectives:

Student will be able to understand

1. Learn the basic terms of Cloud Computing, Cloud Computing architecture, characteristics of Cloud Computing, Hypervisors etc.
2. Understand the technical concepts of Cloud Computing and Cloud management issues.
3. Study performance metrics across compute, network and storage resources and various Cloud delivery and hosting models.
4. Understand the physical hardware components that make up a Cloud.
5. Focuses on how to create virtualized storage, manage security & access and provisioning models.

UNIT-I

INTROUDCTION TO CLOUD & BENEFITS

Understanding Cloud Characteristics - Basic Terms and Characteristics, Object Storage Concepts .To Grasp the Cloud—Fundamental Concepts - The True Nature of the Cloud, Virtualization and Scalability, The Cloud Hypervisor, Key Benefits of Implementing Hypervisors, Foundations of Cloud Computing

UNIT-II

TECHINICAL AND MANAGEMENT ISSUES

Technical concepts of cloud computing (within the cloud)- Technical basics of cloud and scalable computing, The Cloud Infrastructure. Cloud Management - Understanding Cloud Management Platforms, Service-Level Agreements, Managing Cloud Workloads, Securing Data in the Cloud, Managing Devices,

UNIT-III

PERFORMANCE AND MODELS IN CLOUD

Diagnosis and Performance Monitoring - Performance Concepts, Disk Performance, Impact of Configuration Changes. Cloud Delivery and Hosting Models- Private, public, hybrid, community, On-Premises vs. Off-Premises Hosting Accountability and Responsibility Based on Delivery Models, Security Differences between Models

UNIT-IV

HARDWARE CONCEPTS
Hardware Management- Cloud Hardware Resources, Management Differences between Public, Private, and Hybrid Clouds, Tiring

UNIT- V

STORAGE AND SECURITY
Storage Provisioning and Networking - Cloud Storage Concepts, Cloud vs. SAN Storage, Cloud Provisioning, Cloud Storage Technology, Cloud Security and Privacy

DISCUSS REAL LIFE CASE STUDIES FOR CLOUD COMPUTING

1. IRCTC Online Tatkaal Ticket Booking from 10 Am To 12 Pm in India.
2. Providing aadhar card to 800+ billion peoples is a major project.
3. Data security Mechanism in Google

Course Outcomes:

Upon completion of the course, students will be able to

1. To compare the operation, implementation, merits of Cloud Computing systems and suitability for each complex data intensive applications.
2. Implement controlling and managing access to the Cloud.
3. Apply best practices to achieve optimal performance and Cloud delivery and hosting models.
4. Compare the pros and cons of hardware design choices include compute, storage and networking.
5. Select the right networking protocols and network monitoring and alert mechanisms.

Text Book:
Deploying and Managing a cloud Infrastructure, SYlBEX, A Wiley Brand, Abdul Salam, Zafar Gilani, Salman UlHaq. Copyright © 2015 by John Wiley & Sons, Inc., Indianapolis, Indiana ISBN: 978-1-118-87510-0 ISBN: 978-1-118-87558-2 (ebk.)

Reference Book:

Cloud Computing, A Practical ApproachbyTobyVelte, Anthony Velte, Robert Elsenpeter, McGraw Hill Education (India) Private Limited (18 November 2009) **ISBN-10:** 0070683514 **ISBN-13:** 978-0070683518.

Mode of Evaluation: Assignments, I Mid Exam, II Mid Exam and End Exam.

MCA III Year – I Semester

16MCA419 INTERNET OF THINGS (IOT)

L	T	P	C
4	0	0	4

Course Description:
Introduction and evolution of TOI is explain with concepts of M2M. Market perspective of IOT is detailed with technology fundamentals. IOT state of art architecture and reference architectures are detailed.

Course Objectives:
1. To understand the need and global view of IOT.
2. To get knowledge on the market perspective with architectural overview.
3. To get exposure towards the technology fundamentals.
4. To get idea bout state of art architectures with reference models.
5. To understand the concepts of real world designs, industrial automation and commercial setups.

UNIT –I

M2M to IoT-The Vision-Introduction, From M2M to IoT, M2M towards IoT-the global context, A use case example, Differing Characteristics

UNIT –II

M2M to IoT – A Market Perspective– Introduction, Some Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT, The international driven global value chain and global information monopolies. **M2M to IoT-An Architectural Overview**– Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.

UNIT –III

M2M and IoT Technology Fundamentals- Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a Service(XaaS), M2M and IoT Analytics, Knowledge Management

UNIT –IV

IoT Architecture-State of the Art – Introduction, State of the art, **Architecture Reference Model-** Introduction, Reference Model and architecture, IoT reference Model

UNIT -V

IoT Reference Architecture- Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views. **Real-World Design Constraints-** Introduction, Technical Design constraints-hardware is popular again, Data representation and visualization, Interaction and remote control. **Industrial Automation-**Service-oriented architecture-based device integration, SOCRADES: realizing the enterprise integrated Web of Things, IMC-AESOP: from the Web of Things to the Cloud of Things, **Commercial Building Automation-**

Introduction, Case study: phase one-commercial building automation today, Case study: phase two-commercial building automation in the future

Course Outcomes:

1. To understand and explain the need and global view of IOT.
2. To exhibit the knowledge on the market perspective with architectural overview.
3. To explain the fundaments of IOT technologies.
4. To understand and classify state of art architectures along with reference models.
5. To explain the concepts of real world designs, industrial automation and commercial setups.

Text Book:

Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, StamatisKarnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", 1st Edition, Academic Press, 2014.

Reference Books:

1. Vijay Madisetti and ArshdeepBahga, "Internet of Things (A Hands-on-Approach)", 1st Edition, VPT, 2014.

2. Francis daCosta, "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything", 1st Edition, Apress Publications, 2013