**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**
**KAKINADA – 533 003, Andhra Pradesh, India**

# MASTER OF COMPUTER APPLICATIONS (MCA)
# COURSE STRUCTURE & SYLLABUS

*(Applicable for batches admitted from 2019-2020)*

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**

**MCA I SEMESTER**

| S.No | Code | Courses | L | T | P | C |
|------|------|---------|---|---|---|---|
| 1 | R19MCA1101 | Problem Solving with C | 3 | 0 | 0 | 3 |
| 2 | R19MCA1102 | Computer Organization | 3 | 0 | 0 | 3 |
| 3 | R19MCA1103 | Discrete Mathematical Structures | 3 | 0 | 0 | 3 |
| 4 | R19MCA1104 | Probability and Statistics | 3 | 0 | 0 | 3 |
| 5 | R19MCA1105 | Accounting and Financial Management | 3 | 0 | 0 | 3 |
| 6 | R19MCA1106 | English Language Communication Skills Lab | 0 | 0 | 3 | 1.5 |
| 7 | R19MCA1107 | Problem Solving with C Lab | 0 | 0 | 3 | 1.5 |
| 8 | R19MCA1108 | IT Workshop (Lab) | 0 | 0 | 3 | 1.5 |
| Total Credits | | | | | | 19.5 |

**MCA II SEMESTER**

| S.No | Code | Courses | L | T | P | C |
|------|------|---------|---|---|---|---|
| 1 | R19MCA1201 | Data Structures | 3 | 0 | 0 | 3 |
| 2 | R19MCA1202 | Operating Systems | 3 | 0 | 0 | 3 |
| 3 | R19MCA1203 | Software Engineering | 3 | 0 | 0 | 3 |
| 4 | R19MCA1204 | OOP Through Java | 3 | 0 | 0 | 3 |
| 5 | R19MCA1205 | Optimization Techniques | 3 | 0 | 0 | 3 |
| 6 | R19MCA1206 | OOP Through Java Lab | 0 | 0 | 3 | 1.5 |
| 7 | R19MCA1207 | Data Structures Lab | 0 | 0 | 3 | 1.5 |
| 8 | R19MCA1208 | Operating Systems and Linux Lab | 0 | 0 | 3 | 1.5 |
| Total Credits | | | | | | 19.5 |

## MCA III SEMESTER

| S.No | Code | Courses | L | T | P | C |
|------|------|---------|---|---|---|---|
| 1 | R19MCA2301 | Database Management Systems | 3 | 0 | 0 | 3 |
| 2 | R19MCA2302 | Computer Networks | 3 | 0 | 0 | 3 |
| 3 | R19MCA2303 | Design and Analysis of Algorithms | 3 | 0 | 0 | 3 |
| 4 | R19MCA2304 | Advanced Java Programming | 3 | 0 | 0 | 3 |
| 5 | R19MCA2305 | Object Oriented Analysis and Design | 3 | 0 | 0 | 3 |
| 6 | R19MCA2306 | Database Management Systems Lab | 0 | 0 | 3 | 1.5 |
| 7 | R19MCA2307 | Advanced Java Programming Lab | 0 | 0 | 3 | 1.5 |
| 8 | R19MCA2308 | OOAD through UML Lab | 0 | 0 | 3 | 1.5 |
| **Total Credits** | | | | | | 19.5 |

## MCA IV SEMESTER

| S.No | Code | Courses | L | T | P | C |
|------|------|---------|---|---|---|---|
| 1 | R19MCA2401 | Data Warehousing and Data Mining | 3 | 0 | 0 | 3 |
| 2 | R19MCA2402 | Full Stack Technologies | 3 | 0 | 0 | 3 |
| 3 | R19MCA2403 | Python Programming | 3 | 0 | 0 | 3 |
| 4 | R19MCA2404 | **Elective-1**<br>1. Embedded Computing<br>2. Artificial Intelligence<br>3. MOOCs-1 (NPTEL/SWAYAM)(Self learning courses) | 3 | 0 | 0 | 3 |
| 5 | R19MCA2405 | **Elective-2**<br>1. Cloud Computing<br>2. Multimedia Application Development<br>3. MOOCs-2 (NPTEL/SWAYAM) | 3 | 0 | 0 | 3 |
| 6 | R19MCA2406 | Full Stack Technologies Lab | 0 | 0 | 3 | 1.5 |
| 7 | R19MCA2407 | Data Mining with R Lab | 0 | 0 | 3 | 1.5 |
| 8 | R19MCA2408 | Python Programming Lab | 0 | 0 | 3 | 1.5 |
| **Total Credits** | | | | | | 19.5 |

**MCA V SEMESTER**

| S.No | Code | Courses | L | T | P | C |
|------|------|---------|---|---|---|---|
| 1 | R19MCA3501 | Cryptography and Network Security | 3 | 0 | 0 | 3 |
| 2 | R19MCA3502 | Big Data Analytics | 3 | 0 | 0 | 3 |
| 3 | R19MCA3503 | Machine Learning | 3 | 0 | 0 | 3 |
| 4 | R19MCA3504 | **Elective-3**<br>1. Digital Marketing<br>2. Natural Language Processing<br>3. MOOCs-3 (NPTEL/SWAYAM) | 3 | 0 | 0 | 3 |
| 5 | R19MCA3505 | **Elective-4**<br>1. Internet of Things<br>2. DevOps<br>3. MOOCs-4 (NPTEL/SWAYAM) | 3 | 0 | 0 | 3 |
| 6 | R19MCA3506 | Cryptography and Network Security Lab | 0 | 0 | 3 | 1.5 |
| 7 | R19MCA3507 | Big Data Analytics Lab | 0 | 0 | 4 | 2 |
| 8 | R19MCA3508 | Machine Learning with Python Lab | 0 | 0 | 3 | 1.5 |
| **Total Credits** | | | | | | 20 |

**MCA VI SEMESTER**

| S.No | Code | Courses | L | T | P | C |
|------|------|---------|---|---|---|---|
| 1 | R19MCA3601 | Project Seminar | 0 | 0 | 0 | 2 |
| 2 | R19MCA3602 | Dissertation/Thesis | 0 | 0 | 0 | 10 |
| **Total Credits** | | | | | | 12 |

MCA I Semester
**Problem Solving with C**

**Course Objectives:** This course is aimed at enabling the students to
- Formulate simple algorithms for
- ms (in C language).
- Test and execute the programs and correct syntax and logical errors.
- Implement conditional branching, iteration and recursion.
- Decompose a problem into functiarithmetic and logical problems.
- Translate the algorithms to prograons and synthesize a complete program using divide and conquer approach.
- Use arrays, pointers and structures to formulate algorithms and programs.
- Apply programming to solve matrix addition and multiplication problems and searching and sorting problems.
- Apply programming to solve simple numerical method problems, namely root finding of function, differentiation of function and simple integration.

**Course Outcomes (COs)**: At the end of the course, student will be able to
- Understand the basic concepts used in computer programming
- Write, compile and debug programs in C language
- Design programs involving decision structures, loops and functions.
- Understand about the application and implementation of 2-dimentional array, structures and strings
- Understand the dynamics of memory by the use of pointers.
- Develop solutions to problems using derived data types and files.

**UNIT – I:**
**Introduction to Computers:** Introduction to computer programming, Algorithm, flow chart, Program development steps. **Computer languages:** Machine level, Assembly level and High-level language. **Number System:** Conversions- decimal, binary, octal, hexadecimal. **'C' Fundamentals:** Structure of a C-program, C-character set, C Tokens- variables, constants, identifiers, data types and sizes, operators

**UNIT – II:**
**I/O Functions:** Header files, Standard I/O library functions-formatted I/O functions. **Decision making statements:** simple if, if-else, nested if-else, else-if ladder, switch-case statements and sample programs. **Iterative Statements:** for, while, do-while. Jump Statements-break, continue, goto

**UNIT – III:**
**Arrays**: declaration, initialization, storing and accessing elements of 1-D, 2-D and multidimensional arrays, **Array Applications:** addition, multiplication, transpose, symmetry of a matrix, **Strings**: declaration, initialization, reading and writing characters into strings, string operations, character and string manipulation functions

**UNIT – IV:**
**Functions-** declaration, definition, prototype, function call, return statement, types of functions, parameter passing methods, and function recursion, **Pre-processor:** #define, #include Statement, #ifdef, #endif, and storage classes.

**UNIT – V:**
**Structure and Union:** Declaration, initialization, storing and accessing elements by using structure and union, **Pointers:** Introduction to pointers, defining a pointer variable, Pointer to Pointer, Examples of pointers, using pointers in expressions, pointers and arrays. Files: Definition, Input and output operation into file.

**Text Books:**
1. Programming in C, 3rd edition, Ashok N. Kamthane, Pearson
2. Computer science, A structured programming approach using C, Third edition,  B.A. Forouzan and R. F. Gilberg, Thomson

**Reference Books:**
1. The C Programming Language, B.W. Kernighan, Dennis M. Ritchie, PHI/ Pearson.
2. C Programming with problem solving, J.A. Jones & K. Harrow, Dreamtech   Press
3. Programming in C, , 3rd Edition, Stephen G. Kochan, Pearson

MCA I Semester
**Computer Organization**

**Course Objectives:**
The objectives of this course are to
- Conceptualize the basics of organizational and architectural issues of a digital computer.
- Learn the function of each element of a memory hierarchy.
- Study various data transfer techniques in digital computer.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Understand the basic organization of computer and different instruction formats and addressing modes
- Analyze the concept of pipelining, segment registers and pin diagram of CPU.
- Understand and analyze various issues related to memory hierarchy
- Evaluate various modes of data transfer between CPU and I/O devices
- Examine various inter connection structures of multi processors

**UNIT I:**
**Basic Structure of Computers:** Functional unit, Basic Operational concepts, Bus structures, System Software, Performance- Processor clock, basic performance equation, pipelining and super scalar operations, clock rate, CISC,RISC, performance measurement,  the history of computer development- generations.

**UNIT II:**
**Instruction and Instruction Sequencing:** Register Transfer Notation, Assembly Language Notation, Basic Instruction Types, Addressing Modes, Basic Input/output Operations, role of Stacks and Queues in computer programming equation. **Component of Instructions:** Logic Instructions, shift and Rotate Instructions.

**UNIT III:**
**Type of Instructions:** Arithmetic and Logic Instructions, Branch Instructions, **The Memory Systems:** Basic memory circuits, Memory System Consideration, **Read-Only Memory:** ROM, PROM, EPROM, EEPROM, Flash Memory, **Cache Memories:** Mapping Functions, interleaving, **Secondary Storage:** Magnetic Hard Disks, Optical Disks

**UNIT IV:**
**Input/Output Organization:** Accessing I/O Devices, Interrupts: Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Direct Memory Access, **Buses:** Synchronous Bus, Asynchronous Bus, Interface Circuits, **Standard I/O Interface:** Peripheral Component Interconnect (PCI) Bus, Universal Serial Bus (USB)

**UNIT V:**
**Processing Unit: Fundamental Concepts:** Register Transfers, Performing an Arithmetic or Logic Operation, Fetching a Word from Memory, Execution of Complete Instruction, Hardwired Control, **Micro programmed Control:** Microinstructions, Micro program Sequencing, Wide Branch Addressing Microinstructions with next –Address Field

**Text Books:**
1. Computer Organization, 5th Edition, Carl Hamacher, Zvonks Vranesic, Safea Zaky, McGraw Hill
2. Computer Architecture and Organization,3rd Edition, John P. Hayes , McGraw Hill

**Reference Books:**
1. Computer Organization and Architecture, Sixth Edition, William Stallings, Pearson/PHI
2. Structured Computer Organization, 4th Edition, Andrew S. Tanenbaum, PHI/Pearson
3. Fundamentals or Computer Organization and Design, Sivaraama Dandamudi Springer Int. Edition

MCA I Semester
**Discrete Mathematical Structures**

**Course Objectives:**
The objective of this course is to provide the necessary back ground of discrete structures with particular reference to the relationships between discrete structures and their data structure counterparts including algorithm development.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Perform operations on various discrete structures such as sets, functions, relations, and sequences.
- Ability to solve problems using Counting techniques, Permutation and Combination, Recursion and generating functions
- Apply algorithms and use of graphs and trees as tools to visualize and simplify Problems
- Use of K-Maps and Truth Tables to construct and verify correctness of a Boolean expression
- Create the various properties of algebraic systems like Rings, Monoids and Groups

**UNIT I:**
**Mathematical Logic:** Statements and notations, Connectives, Well formed formulas, Truth tables, tautology, equivalence implication, Normal forms, Theory of inference for the statement calculus, Rules of inference, Consistency of premises and indirect method of proof, Automatic Theorem Proving Predicate calculus: Predicates, statement functions, variables and quantifiers, predicate formulas, free & bound variables, universe of discourse, inference theory of predicate calculus

**UNIT II:**
**Set theory & Relations:** Introduction, Relations and ordering, Properties of binary Relations, Equivalence, Compatibility Relations, Partial ordering, Hasse diagram. **Functions:** composition of functions, Inverse Function, Recursive Functions, Lattice and its Properties, Pigeon hole Principles and its application. **Algebraic structures**: Algebraic systems, Examples and general properties, Semi groups and monoids, groups, sub groups, Definitions, Examples, mom orphism, Isomorphism and related problems.

**UNIT III:**
**Elementary Combinatorics:** Basis of counting, Enumeration of Combinations & Permutations, Enumerating of Combinations & Permutations with repetitions and constrained repetitions, Binomial Coefficients, Binomial Multinomial theorems, principles of Inclusion – Exclusion

**UNIT IV:**
**Recurrence Relations:** Generating Function of Sequences, Calculating Coefficient of generating functions, Recurrence relations, Solving recurrence relation by substitution and Generating functions, the method of Characteristic roots, Solution of Inhomogeneous Recurrence Relation.

**UNIT V:**

**Graph Theory:** Representation of Graph, Spanning Trees, BFS, DFS, Kruskals Algorithm, Binary trees, Planar Graphs, Graph Theory and Applications, Basic Concepts, Isomorphism and Sub graphs, Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic Numbers

**Text Books:**

1. Discrete Mathematical Structures with Applications to computer science J.P  Tremblery, R.Manohar, TMH
2. Discrete Mathematical for computer Scientists & Mathematicians " J.L. Molt, A.Kandel, T.P.Baker, PHI

**Reference Books:**

1. Elements of Discrete Mathematics, C L Liu, D P Mohanpatra, TMH
2. Discrete Mathematics, Schaum's Outlines, Lipschutz, Lipson, TMH
3. Discrete Mathematical Structures, 6$^{th}$ edition, Kolman, Busby, Ross, PHI, 2009
4. Discrete Mathematics, 6$^{th}$ edition Johnsonbaugh,  Pearson, 2005
5. Discrete Mathematics, Malik, Sen, 6th ed., Cengage Learning, 2004
6. Discrete Mathematics for computer science, Bogart, Stein and Drysdale, Springer, 2005

MCA I Semester
**Probability and Statistics**

**Course Objectives:**

This course aims to provide an understanding of the basic concepts in probability theory and statistical analysis. Students will learn the fundamental theory of distribution of random variables, the basic theory and techniques of parameter estimation and tests of hypotheses. After taking this course, students will be able to use calculators and tables to perform simple statistical analyses for small samples and use popular statistics packages, such as SPSS, S-Plus, R or Mat Lab, to perform simple and sophisticated analyses for large samples.

**Course Outcomes(COs):** At the end of the course, student will be able to

- Demonstrate the basic knowledge on fundamental probability concepts, including random variable, probability of an event, additive rules and conditional probability.
- Derive the probability density function of transformations of random variables and use these techniques to generate data from various distributions
- Demonstrate the basic statistical concepts and measures
- Discuss several well-known distributions, including Binomial, Geometrical, Negative Binomial, Normal and Exponential Distribution
- Prove hypotheses testing

**UNIT I:**

Probability:  Sample space and events, Probability – The axioms of probability, Some Elementary theorems - Conditional probability, Baye's theorem, Random variables, Discrete and continuous distributions - Distribution function.

**UNIT II:**

Binomial, Poisson, normal distribution – related properties, Moment generating function, Moments of standard distributions – properties.

**UNIT III:**

Population and samples, Sampling distribution of mean (with known and unknown variance), proportion, variances, Sampling distribution of sums and differences, Point and interval estimators for means, variances, proportions.

**UNIT IV:**

Statistical Hypothesis – Errors of Type I and Type II errors and calculation,  One tail, two-tail tests,  Testing hypothesis concerning means, proportions and their differences using Z-test, Tests of hypothesis using Student's t-test, F-test and $\chi^2$ test.. Test of independence of attributes, ANOVA for one-way and two-way classified data.

**UNIT V:**

Statistical Quality Control methods, Methods for preparing control charts, Problems using x-bar, p,  R charts and attribute charts, Simple Correlation and Regression, Queuing Theory: Pure Birth and Death Process M/M/1 Model and Simple Problems.

**Text Books:**
1. Probability and Statistics for Engineers, Miller and John E. Freund, Prentice Hall of India
2. Probability and Statistics, D. K. Murugeson & P. Guru Swamy, Anuradha Publishers

**Reference Books:**
1. Probability, Statistics and Random processes. T. Veerrajan, Tata Mc.Graw Hill, India
2. Probability, Statistics and Queuing theory applications for Computer Sciences, 2 ed, Trivedi, John Wiley

MCA I Semester
**Accounting and Financial Management**

**Course Objectives:**
The main objectives of this course are
- To understand the basic concepts and processes used to determine product costs
- To be able to interpret cost accounting statements
- To be able to analyze and evaluate information for cost ascertainment, planning, control and decision making

**Course Outcomes (COs)**: At the end of the course, student will be able to
- Understand the balance sheet preparation and perform analysis
- Understand the budget preparation and control of a company
- Decide about the state of affairs of a particular firm / company
- Ensure the preparation of fiscal policies of the organization
- Ensure the factors to be considered in investment policies

**UNIT I:**
Accounting Generally Accepted Accounting Principles (GAAP) & accounting standards, Characteristics and limitations of single entry system, double entry system of accounting, introduction of basis books of accounts, ledgers, Preparation of trail balance, Final accounts, company final accounts, Users of Accounting Information, Role of Accountant in modern Organizations.

**UNIT II:**
Financial Management, meaning and scope, role, objectives of time value of money over vitalization, under capitalization, profit maximization, wealth maximization, EPS maximization. Ration Analysis, advantages, limitations, Fund flow analysis, meaning, importance, preparation and interpretation of Funds flow and cash flow statements, statements of changes in working capital

**UNIT III:**
**Costing:** nature and importance and basic principles, Elements of cost, Absorption costing Vs. Marginal costing, financial accounting Vs. cost Accounting Vs. management accounting. Marginal costing and Break, even Analysis: nature, scope and importance, practical applications of marginal costing, limitation and importance of cost, volume, profit analysis, short run decisions

**UNIT IV:**
**Standard costing and budgeting :** nature, scope and computation and analysis, materials variance, labour variance and sales variance, cash budget, sales, budget, flexible Budgets, master budgets.

**UNIT V:**
**Introduction to computerized accounting system:** coding logic and codes, master files, transaction files, introduction documents used for data collection, processing of different files and Outputs obtained.

**Text Books:**
1. Accounting for Managers, P. Vijaya Kumar, and Himalaya Publications
2. Accounting for Management. Vijaya Kumar.TMH
3. Financial Accounting, S.N Maheswari and S.K. Maheswari, Vikas
4. Financial Accounting, A. Mukherjee and M. Heneef, TMH

**Reference Books:**
1. Basic Financial Accounting for Management, Ambaresh Gupta, Pearson
2. Accounts And Finance for Non accounts, Chatterjee, D.K.Himalaya
3. Financial Analysis and Accounting, P. Premchand Babu and M.  Madam Mohan, Himalaya
4. Essential of Financial Accounting, Ashish, K and Ballacharya, PHI
5. Guide to Financial Management, John Tannent, Viva

MCA I Semester
**English Communication Skills Lab**
Code: R19MCA1106

MCA I YEAR I SEMESTER (R19)

## ENGLISH COMMUNICATION SKILLS LAB

### COURSE OBJECTIVES:

The English language lab focuses on computer-aided multi-media instruction and language acquisition to achieve the following objectives:

- To sensitize the students to the nuances of English speech sounds, word accent, intonation and rhythm and understand the concepts of " neutralized accent and phonetic transcription"
- To enable students comprehend the concept of communication
- Help improve speaking skills through participation in activities such as role plays, discussions and structured talks/oral presentations
- To gain experience in extemporaneous speaking and oral presentation
- To train students to use language appropriately for interviews, group discussion and public speaking

### COURSE OUT COMES:

- Ability to convert the understanding of phonetics in every day practice.
- Better understanding of the nuances of language through audio- visual experience and group Activities
- Speaking with clarity and confidence thereby enhancing employability skills of the students.
- Refines professional writing and correspondence

### Suggested Software:

- Cambridge Advanced Learners' English Dictionary with CD.
- Grammar Made Easy by Darling Kindersley
- Punctuation Made Easy by Darling Kindersley
- Clarity Pronunciation Power – Part I
- Clarity Pronunciation Power – part II
- Oxford Advanced Learner's Compass, 8th Edition
- Lingua TOEFL CBT Insider, by Dreamtech
- TOEFL & GRE (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS)
- English in Mind (Series 1-4), Herbert Puchta and Jeff Stranks with Meredith Levy, Cambridge
- English Pronunciation in Use (Elementary, Intermediate, Advanced) Cambridge University Press
- Tedx Talks

| S.No | Topics | No. of Sessions |
|---|---|---|
| 1. | Vowels, Consonants, Pronunciation, Phonetic Transcription | 01 |
| 2. | Syllabification, Stress, Intonation, Listening Comprehension, Common errors in Pronunciation | 02 |
| 3. | In-house communication (Situational Dialogue/Role Play)<br>  a.  Individual to individual / group<br>     i. Welcoming new comers to the organization,<br>     ii.  Introducing the workplace culture<br>     iii. Explaining duties and responsibilities etc.<br>     iv. Motivating ('pep talk')<br>     v. Expressing / recording appreciation, praising / rewarding a subordinate or junior<br>  b. Subordinate / Junior to Superior<br>     i. Reporting problems / difficulties.<br>     ii. Offering suggestions | 01 |
| 4 | Data Interpretation, Reading Comprehension | 01 |
| | 1.  Public Speaking: Power point Presentations Group Discussions, Debates | 04 |
| 5 | Resume Writing and Interviews | 03 |
| 6 | Notices, Minutes of the meetings and e-Correspondence, GRE/TOEFL Models | 02 |

**Reference Books:**

1. The Human Touch: personal Skills for Professional Success – by Debra Paul.
2. The Definitive Book of body Language – by Allan Pease, Barbara Pease.
3. How to Face Interviews – by Clive Fletcher's
4. The 7 Habits of Highly Effective People – by Stephen Covery.
5. Good English – by G.H Vallins
6. Better English – G.H Vallins
7. Best English – G.H. Vallins
8. How to Talk to Anyone: 92 little tricks for big success in Relationships by Leli Lowndes.
9. English Pronunciation in use- Mark Hancock, Cambridge University Press.
10. English Phonetics and Phonology-Peter Roach, Cambridge University Press.
11. English Pronunciation in use- Mark Hewings, Cambridge University Press.
12. English Pronunciation Dictionary- Daniel Jones, Cambridge University Press.

MCA I Semester
**Problem Solving with C Lab**

**Course Objectives:**
The purpose of this course is to introduce to students to the field of programming using C language. The students will be able to enhance their analyzing and problem solving skills and use the same for writing programs in C.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Obtain hands on experience in programming to solve the real time problems.
- Write diversified solutions using C language
- Implement Programs with pointers and arrays, perform pointer arithmetic, and use the pre-processor
- Write programs that perform operations using derived data types
- Identify tasks in which the numerical techniques learned are applicable and apply them to write programs, and hence use computers effectively to solve the task.

**Experiment l**
(a) Write a C Program to calculate the area of a triangle
(b) Write a C program to find the largest of three numbers using ternary operator
(c) Write a C Program to swap two numbers without using a temporary variable
**Experiment 2**
(a) Write a C program to find the 2's complement of a binary number
(b) Write a C program to find the roots of a quadratic equation
(c) Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +,-,*, /, % and use Switch Statement)
**Experiment 3**
(a) Write a C program to find the sum of individual digits of a positive integer and, also, find the reverse of the given number.
(b) Write a C program to generate the first *n* terms of the Fibonacci sequence
(c) Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
**Experiment 4**
(a) Write a C Program to print the multiplication table of a given number
(b) Write a C Program to read a decimal number and find its equivalent binary number
(c) Write a C Program to check whether the given number is Armstrong number
   or not.
**Experiment 5**
(a) Write a C program to interchange the largest and smallest numbers in the given array
(b) Write a C program to implement a liner search on a given set of values
(c) Write a C program to implement binary search on a given set of values
**Experiment 6**
(a) Write a C program to implement sorting of an array of elements
(b) Write a C program to input two m x n matrices, check the compatibility and
   perform addition and multiplication of them.

**Experiment 7**

Write a C program that uses functions to perform the following operations:

(a) To insert a sub-string into given main string at a given position

(b) To delete n characters from a given position in a given string

(c) To replace a character of string either from beginning or ending or at a
   specified location.

**Experiment 8**

Write a C program that uses functions to perform the following operations using Structure:

(i) Reading a complex number (ii) Writing a complex number  (iii) Addition of two complex
numbers  (iv) Multiplication of two complex numbers

**Experiment 9**

Write C Programs for the following string operations without using the built in functions:

(i) to concatenate two strings (ii) to append a string to another string (iii) to compare two strings

**Experiment 10**

(a) Write a C Program to find the number of characters in a given string
   including and excluding spaces

(b) Write a C Program to copy the contents of one string to another string
   without using string handling functions

(c) Write C Program to find whether a given string is palindrome or not.

(d) Write a C program to find both the largest and smallest number of an array
   of integers using call by value and call by reference.

**Experiment 11**

Write a C program using recursion for the following:

(a) To display sum of digits of given number

(b) To find the factorial of a given integer

(c) To find the GCD (greatest common divisor) of two given integers

(d) To find the Fibonacci sequence

**Experiment 12**

(a) Write C Program to reverse a string using pointers.

(b) Write a C Program to compare two 2D arrays using pointers.

(c) Write a C program consisting of Pointer based function to exchange value of
   two integers using passing by address.

**Experiment 13**

Examples which explores the use of structures, union and other user defined variables.

**Experiment 14**

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

(b) Write a C program to count the number of characters and number of lines in
   a file.

(c) Write a C Program to merge two files into a third file, where the names of the
   files must be entered using command line arguments.

MCA I Semester
**IT Workshop (Lab)**

**Course Objectives:**
- Students learn and understand why computers are essential components in business, education and society.
- To learn the fundamentals of computing devices and use of computer hardware and software, the Internet, networking.
- To learn different applications like Microsoft Office Word, Excel, Access and PowerPoint.

**Course Outcomes (COs)**: At the end of the course, student will be able to
- Apply knowledge for computer assembling and software installation.
- Solve hardware, software trouble shooting problems.
- Develop Documents using Word processors
- Connect computer using wired and wireless connections
- Develop presentations using the presentation tool and  Perform computations using spreadsheet tool

**List of Exercises:**
Note: Faculty to consolidate the workshop manuals using the textbook and references.

**Experiment 1: Identification of the peripherals of a computer -** Prepare a report containing the block diagram of the computer along with the configuration of each component and its functionality. Describe about various I/O Devices and its usage.

**Experiment 2:** Practicing disassembling and assembling components of a PC.

**Experiment 3:** Installation of Device Drivers, MS windows, Linux Operating systems and Disk Partitioning.

**Experiment 4:** Introduction to Memory and Storage Devices, I/O Port, Assemblers, Compilers, Interpreters, Linkers and Loaders.

**Experiment 5:** Demonstration of Hardware and Software Troubleshooting.

**Experiment 6**: Demonstrating Importance of Networking, Transmission Media, Networking Devices- Gateway, Routers, Hub, Bridge, NIC, Bluetooth Technology, Wireless Technology, Modem, DSL, and Dialup Connection.

**Experiment 7**: Awareness of various threats on the Internet and its solutions.

**Experiment 8**: Demonstration and Practice on Microsoft Word.

**Experiment 9:**  Demonstration and Practice on Microsoft Excel.

**Experiment 10:**  Demonstration and Practice on Microsoft Power Point.

**Experiment 11:** Demonstration and Practice on LaTeX.

**Text Books:**
1 Computer Fundamentals, Anita Goel, Pearson India Education, 2017
2 PC Hardware Trouble Shooting Made Easy, TMH

**Reference Books:**
1. Upgrading and Repairing PCs, 18th Edition, Scott Mueller, QUE, Pearson, 2008
2. Comdex Information Technology, Vikas Gupta, Dreamtech
3. Essential Computer and IT Fundamentals for Engineering and Science Students, Dr. N. B. Venkateswarlu, S. Chand Publishers
4. Information Technology Workshop, 3rd Edition, G Praveen Babu, M V Narayana, BS Publications

MCA II Semester
**Data Structures**

**Course Objectives:**
- To understand algorithms and its analysis procedure.
- To design and implement various data structure algorithms
- To introduce various techniques for representation of different types of data in the real world.
- Compute the complexities of various algorithms.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Select appropriate data structures as applied to specified problem definition.
- Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.
- Compare Linear and Non-Linear data structures.
- Apply appropriate sorting/searching technique for given problem.
- Design advance data structure using Non Linear data structure.
- Determine and analyze the complexity of given Algorithms

**UNIT-I:**
**Data structure:** Definition, types of data structures Recursion Definition, Design Methodology and Implementation of recursive algorithms, Linear and binary recursion, recursive algorithms for factorial function, GCD computation, Fibonacci sequence. Preliminaries of algorithms, analysis and complexity. **Searching Techniques:** Linear Search, Binary Search and Fibonacci Search. **Sorting Techniques:** Insertion sort, selection sort, exchange-bubble sort, quick sort and merge sort Algorithms.

**UNIT-II:**
**Stacks:** Basic Stack Operations, Representation of a Stack using Arrays, **Applications of Stack:** Reversing list, Factorial Calculation, Infix to postfix Transformation, Evaluating Arithmetic Expressions.

**UNIT-III:**
**Queues:** Basic Queues Operations, Representation of a Queue using array, Implementation of Queue Operations using Stack. **Applications of Queues**-Circular Queues, De-queue, Priority Queues.

**UNIT-IV:**
**Linked Lists:** Introduction, single linked list, representation of a linked list in memory, Operations on a single linked list, reversing a single linked list, Circular linked list and doubly linked list.

**UNIT-V:**
**Trees**: Binary Trees, terminology, representation and traversals- pre, post & in order traversals.
**Graphs-** terminology, representation tend traversals (BFS & DFS), Spanning trees, shortest path algorithms, Minimum spanning trees- Prim's, Kruskal's, Dijstra's and Warshall's algorithms.

**Text Books:**
1. Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahani and Susan Anderson Freed, 2nd Edition, Universities Press
2. Data Structures with C, 2ndEdition, Seymour Lipscutz, Schaum's Outlines, TMH
3. Data structures using C, 2ndEdition, Reema Thareja, Oxford higher education

**Reference Books:**
1. Data structures: A Pseudo code Approach with C, 2nd edition, R.F.Gilberg     and B.A.Forouzan, Cengage Learning
2. Data structures and Algorithm Analysis in C, 2nd edition, M.A.Weiss,     Pearson
3. Data Structures using C, A.M.Tanenbaum, Y. Langsam, M.J.Augenstein,     Pearson

MCA II Semester
**Operating Systems**

**Course Objectives:**
- To introduce different types of operating systems.
- To learn process management techniques.
- To learn various memory management techniques.
- To introduce the architecture of Linux operating system.
- To learn basic Unix commands and to write shell scripts.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Understand the basics of operating systems like kernel, shell, types and views of operating systems
- Understands CPU scheduling algorithms and compare the results using Gantt chart.
- Explain various memory management techniques and concept of thrashing
- Apply disk scheduling algorithms for better utilization of external memory
- Understand the architecture of Unix operating system
- Write and execute shell programs

**UNIT–I**:
**Introduction to Operating System Concept:** Types of operating systems, operating systems concepts, operating systems services, Introduction to System call, System call types. **Process Management:** Process concept, The process, Process State Diagram , Process control block, Process Scheduling- Scheduling Queues, Schedulers, Operations on Processes, Interprocess Communication, Threading Issues, Scheduling-Basic Concepts, Scheduling Criteria, Scheduling Algorithms.

**UNIT–II:**
**Memory Management:** Swapping, Contiguous Memory Allocation, Paging, structure of the Page Table, Segmentation **Virtual Memory Management:** Virtual Memory, Demand Paging, Page-Replacement Algorithms, Thrashing. **File system Interface:** the concept of a file, Access Methods, Directory structure, File system mounting, files sharing, protection. **File System implementation:** File system structure, allocation methods, free-space management, Disk scheduling

**UNIT–III:**
**Concurrency:** Process Synchronization, The Critical- Section Problem, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization examples **Principles of deadlock:** System Model, Deadlock  characterization, Deadlock Prevention, Detection and Avoidance, Recovery form Deadlock

**UNIT–IV:**
**Introduction to Unix operating system:** History of Unix, features of Unix Operating system, Architecture of Unix operating system, Unix File system, Unix environment, Basic Unix commands. Using Vi editor, Different types of shells. **Shell Programming**: Shell Variables, The Export Command, The Profile File a Script Run During Starting, The First Shell Script, The read Command, Positional parameters, The $? Variable knowing the exit Status-More about the Set Command, The Exit Command

**UNIT-V:**

**Bourne shell Programming:** Introduction, Writing and executing shell scripts, Positional parameters, Branching Control Structures, Loop Control Structures, The Continue and Break Statement, The expr Command: Performing Integer Arithmetic-Real Arithmetic in Shell Programs, The here Document(<<), The Sleep Command-Debugging Scripts, The Script Command, The eval Command

**Text Books:**
1. Operating system concepts, 7/e, Abraham Siliberschatz, Galvin, John Wiley & Sons, Inc
2. Advanced programming in the unix environment by W. Richard Stevens
3. Unix network programming by W. Richard Stevens

**Reference Books:**
1. Operating systems, 6/E, William stallings, PHI/Pearson
2. Operating systems 3/e, Dietal, Dietal, Pearson
3. Operating systems, 2/e, Dhamdhere, TMH
4. Introduction to Unix and shell programming, Venkateshmurhty
5. Unix and shell programming by B.M. Harwani, OXFORD university press

MCA II Semester
**Software Engineering**

**Course Objectives:**
- To understand the nature of software development and software life cycle models
- To understand methods of capturing, specifying, visualizing and analyzing software requirements.
- To know basics of testing and understanding concept of software quality assurance and software configuration management process.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Define various software application domains and remember different process model used in software development.
- Explain needs for software specifications also they can classify different types of software requirements and their gathering techniques.
- Convert the requirements model into the design model and demonstrate use of software and user interface design principles.
- Distinguish among SCM and SQA and can classify different testing strategies and tactics and compare them.
- Justify role of SDLC in Software Project Development.

**UNIT–I:**
**Introduction to Software Engineering:** The evolving role of software, Changing Nature of Software, Software myths. (Text Book 3), **The software problem:** Cost, schedule and quality, Scale and change.

**UNIT–II:**
**Software Process:** Process and project, component software process, **Software development process models :** Waterfall model, prototyping, iterative development, relational unified process, time boxing model, Extreme programming and agile process, using process models in a project, Project management process.

**UNIT- III:**
**Software requirement analysis and specification:** Value of good SRS, requirement process, requirement specification, functional specifications with use-cases, other approaches for analysis, validation, **Planning a software project:** Effort estimation, project schedule and staffing, quality planning, risk management planning, project monitoring plan, detailed scheduling.

**UNIT– IV:**
**Software Architecture:** Role of software architecture, architecture views, components and connector view, architecture styles for C & C view, documenting architecture design, evaluating architectures, **Design:** Design concepts, function-oriented design, object oriented design, detailed design, verification, metrics.

**UNIT-V:**

**Coding and Unit testing:** Programming principles and guidelines, incrementally developing code, managing evolving code, unit testing, code inspection, metrics, **Testing:** Testing concepts, testing process, black-box testing, white-box testing, metrics.

**Text Books:**

1. A Concise introduction to software engineering (undergraduate topics in computer science), Pankaj Jalote, Springer International Edition
2. Software Engineering, APrecise approach, Pankaj Jalote, Wiley
3. Software Engineering, 3/e, & 7e Roger S.Pressman, TMH

**Reference Books:**

1. Software Engineering, 8/e, Sommerville, Pearson
2. Software Engineering principles and practice, W S Jawadekar, TMH
3. Software Engineering concepts, R Fairley, TMH

MCA II Semester
**OOP through Java**

**Course Objectives:**
- To learn the object oriented programming concepts.
- To introduce the principles of inheritance and polymorphism and demonstrate how they are related to the design of abstract classes
- To introduce the implementation of packages and interfaces
- To introduce the concept of multithreading and exception handling
- To introduce the design of Graphical User Interface using applets and swing  controls

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Understand the use OOP concepts
- Apply OOP concepts to solve real world problems
- Understand the concepts of packages and interfaces
- Understand the concepts of exception handing, multithread applications with synchronization
- Design the GUI based applications using AWT and Swings
- Understand the concept of Collection Framework

**UNIT–I:**
**Basics of Object Oriented Programming (OOP):** Need for OO paradigm , A way of viewing world- Agents, responsibility, messages, methods, classes and instances, class hierarchies (Inheritance), method binding, overriding and exceptions, summary of OOP concepts, coping with complexity, abstraction mechanisms, **Java Basics:** Data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and costing, simple java program, classes and objects- concepts of classes, objects, constructors methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, string handling.

**UNIT–II:**
**Inheritance:** Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance- specialization, specification, construction, extension, limitation, combination, benefits of inheritance costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism, abstract classes, **Packages and Interfaces:** Defining, Creating and Accessing a package, Understanding CLASSPATH, Importing packages, differences between classes and interfaces, defining an interface, Implementing interface, applying interfaces variables in interface and extending interfaces.

**UNIT–III:**
**Exception handling and Multithreading**: Concepts of exception handling, benefits of exception handling, Termination or presumptive models, exception hierarchy, usage of try, catch, throws and finally, built in exceptions, creating own exception sub classes. Differences between multi threading and multitasking, thread life cycle, creating threads, synchronizing threads, daemon threads, thread groups.

## UNIT–IV:

**Event Handling:** Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes, inner classes. The AWT class hierarchy , user-interface components- labels, button, canvas, scrollbars, text components, check box, check box groups, choices, list panes- scroll pane, dialogs, menu bar, graphics, layout manager- layout manager types- boarder, grid, flow, card and grid bag.

## UNIT-V:

**Applets:** Concepts of Applets, differences between applets and applications, lifecycle of an applet, types of applets, creating applets, passing parameters to applets, **Swings:** Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons-The JButton class, Check boxes, Radio Buttons, Combo boxes, Tabbed panes, Scroll panes, Trees and Tables.

## Text Books:

1. Java-The complete reference,7/e, Herbert schildt, TMH
2. JAVA: How to program, 8/e, Dietal , Dietal,PHI
3. Introduction of programming with JAVA,S.Dean,TMH
4. Introduction to Java programming, 6/e, Y.Daniel Liang, Pearson

## Reference Books:

1. Core Java 2, Vol 1(Vol 2) Fundamentals(Advanced), 7/e, Cay.S.Horstmann, Gary Cornell, Pearson
2. Big Java2,3/e, Cay.S. Horstmann, Wiley
3. Object Oriented Programming through Java, P.Radha Krishna, University Press
4. JAVA& Object Orientation an Introduction, 2/e, John Hunt, Springer
5. Introduction to JAVA Programming, 7/e, Y. Daniel Liang, Pearson. , TMH

MCA II Semester
**Optimization Techniques**

**Course Objectives:**
- To understand the theory of optimization methods and algorithms developed for solving various types of optimization problems.
- To develop and promote research interest in applying optimization techniques in problems of Engineering and Technology.
- To apply the mathematical results and numerical techniques of optimization theory to concrete Engineering problems.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Describe clearly a problem, identify its parts and analyze the individual functions
- Feasibility study for solving an optimization problem
- Becoming a mathematical translation of the verbal formulation of an optimization problem
- To design algorithms, the repetitive use of which will lead reliably to finding an approximate solution
- Discovery, study and solve optimization problems
- Investigate, study, develop, organize and promote innovative solutions for various applications

**UNIT–I:**
**Development:** Definition, Characteristics and Phrases, scientific method. Types of models, general methods for solving, operations research modes, **Allocation:** introduction linear programming formulation, graphical solution, simplex methods, artificial variable technique, duality principle,

**UNIT-II:**
**Transportation problem:** Formulation, optimal solution, unbalanced transportation, **Assignment problem:** formulation, optimal solution, variations problem, degeneracy i.e. non square MXN) matrix, restrictions, **Sequencing:** Introduction, optimal solution for processing each of n jobs through three machines, travelling salesman problem (i.e.) shortest acyclic route models.

**UNIT-III:**
**Replacement:** Introduction, replacement of items that deteriorate when money value is not counted and counted, and replacement of items that fail completely (i.e.) group replacements, **Waiting lines:** Introduction, single channel, poisson arrivals, exponential service time infinite population and unrestricted queue.
**UNIT-IV:**
**Inventory:** Introduction, single item, deterministic models, production is instantaneous or at a constant rate , shortages are allowed or not allowed and with drawls from stock is continuous, purchase inventory model with one price break, shortages are not allowed, instantaneous production demand production or purchase cost is relevant, stochastic models, simple problems.

**UNIT–V:**
**Theory of Games:** Introduction, minmax (maximum), criterion and optimal strategy solution of games with saddle points, rectangular without saddle points. **Dynamic programming:** Introduction, Bellman's Principle of optimality, solutions for simple problems, **Project Management:** PERT and CPM, difference between PERT and CPM, PERT/CPM network components and precedence relations, Time Estimates for activities

**Text Books:**
1. Operations Research, S.D.Sharma, Ramnath, & Kedarnath co, Meerut
2. Operations Research, An introduction, 8/e, Taha, Pearson

**Reference Books:**
1. Operations Research, P.K.Gupta, D.S. Hira, S.Chand
2. Operations Research, R.D.Asrhedkar, R.V.Kulkarni
3. Operations Research, Problems & sollutons, 3/e, JKSharma, Macmillan
4. Operations Research, 8/e, Hillier, Liberman, TMH
5. Operations Research, 2/e, Panneerselvam

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**
**KAKINADA – 533 003, Andhra Pradesh, India**

MCA II Semester
**OOP through Java Lab**

**Course Objectives:**
- To understand how to design, implement, test, debug, and document programs that use basic data types and computation, simple I/O, conditional and control structures, string handling and functions.
- To understand the importance of Classes & objects along with constructors, Arrays and Vectors.
- Discuss the principles of inheritance, interface and packages and demonstrate though problem analysis assignments how they relate to the design of methods, abstract classes and interfaces and packages.
- To understand importance of Multi-threading & different exception handling mechanisms.
- To learn experience of designing, implementing, testing, and debugging graphical user interfaces in Java using applet and AWT that respond to different user events.
- To understand Java Swings for designing GUI applications based on MVC architecture

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Apply OOP concepts to solve real world problems
- Implement different forms of inheritance
- Create packages and to reuse them.
- Implement multi threaded programs using synchronization concepts
- Create user defined exceptions
- Design GUI applications using AWT and SWINGS.

**List of Experiments:**
1. The Fibonacci sequence is defined by the following rule. The first 2 values in the sequence are 1, 1. Every subsequent value is the sum of the 2 values preceding it. Write a Java Program that uses both recursive and non recursive functions to print the nth value of the Fibonacci sequence.
2. Write a Java Program that prompts the user for an integer and then prints out all the prime numbers up to that Integer.
3. Write a Java Program that checks whether a given string is a palindrome or not. Ex. MALAYALAM is a palindrome.
4. Write a Java Program for sorting a given list of names in ascending order.
5. Write a Java Program that illustrates how runtime polymorphism is achieved.
6. Write a Java Program to create and demonstrate packages.
7. Write a Java Program, using StringTokenizer class, which reads a line of integers and then displays each integer and the sum of all integers.
8. Write a Java Program that reads on file name form the user then displays information about whether the file exists, whether the file is readable/ writable, the type of file and the length of the file in bytes and display the content of the using FileInputStream class.
9. Write a Java Program that displays the number of characters, lines and words in a text/text file.

10. Write an Applet that displays the content of a file.
11. Write a Java Program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +-*?% operations. Add a text field to display the result.
12. Write a Java Program for handling mouse events.
13. Write a Java Program demonstrating the life cycle of a thread.
14. Write a Java Program that lets users create Pie charts. Design your own user interface (with Swings & AWT).
15. Write a Java Program to implement a Queue, using user defined Exception Handling (also make use of throw, throws).

MCA II Semester
**Data Structures Lab**

**Course Objectives:**
- To design and implement various data structures.
- To implement operations like searching, insertion and deletion, traversing    mechanism
- To develop applications using data structure algorithms.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Implement various basic data structures and its operations.
- Apply sorting and searching algorithms to given numbers
- Implement various tree operations.
- Implement various graphs algorithms.
- Develop applications using various data structures.

**Experiment 1:**
Write recursive program which computes the n$^{th}$ Fibonacci number, for appropriate values of n.

**Experiment 2:**
Write recursive program for the following
a) Write recursive and non recursive C program for calculation of Factorial of an integer.
b) Write recursive and non recursive C program for calculation of GCD (n, m)
c) Write recursive and non recursive C program for Towers of Hanoi: N disks are to be transferred from peg S to peg D with Peg I as the intermediate peg.

**Experiment 3:**
a) Write C program that use both recursive and non recursive functions to perform Linear search for a Key value in a given list.
b) Write C program that use both recursive and non recursive functions to perform Binary search for a Key value in a given list.

**Experiment 4:**
a) Write C program that implement Bubble sort, to sort a given list of integers in ascending order.
b) Write C program that implement Quick sort, to sort a given list of integers in ascending order.

**Experiment 5:**
a) Write C program that implement heap sort, to sort a given list of integers in ascending order.
b) Write C program that implement merge sort, to sort a given list of integers in ascending order.

**Experiment 6:**
a) Write  C program that implement stack (its operations) using  arrays.
b) Write  C program that implement stack (its operations) using  Linked list.

**Experiment 7:**
a) Write a C program that uses Stack operations to Convert  infix expression into postfix expression.
a) Write C program that implement Queue (its operations) using arrays.
b) Write C program that implement Queue (its operations) using linked lists.

**Experiment 8:**
Write a C program that uses functions to create a singly linked list and perform various operations on it.

**Experiment 9:**
Write a C program to store a polynomial expression in memory using linked list and perform polynomial addition.

**Experiment 10**:
a) Write a recursive C program for traversing a binary tree in preorder, inorder and postorder.
b) Write a non recursive C program for traversing a binary tree in preorder, inorder and postorder.

**Experiment 11:**
a) Write a C program to  implement Prims' algorithm.
b) Write a C program to  implement Kruskal's algorithm.

MCA II Semester
## Operating Systems and Linux Lab

**Course Objectives:**
- To implement CPU scheduling algorithms
- To implement Disk scheduling algorithms
- To execute different types of Linux commands
- To write shell scripts

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Implement various CPU scheduling algorithms and compare results
- Implement various disk scheduling algorithms and compare results
- Implement page replace algorithms
- Implement various memory management techniques.
- Execute basic Linux commands

**List of Experiments:**
### Operating Systems Lab

1. Simulate the Following CPU Scheduling Algorithms
   A) FCFS   B) SJF   C) Priority   D) Round Robin
2. Multiprogramming-Memory Management- Implementation of fork(), wait(), exec() and exit()
3. Simulate The Following
   a. Multiprogramming with A Fixed Number Of Tasks (MFT)
   b. Multiprogramming with A Variable Number Of Tasks (MVT)
4. Simulate Bankers Algorithm for Dead Lock Avoidance
5. Simulate Bankers Algorithm for Dead Lock Prevention.
6. Simulate The Following Page Replacement Algorithms.
   A) FIFO        B) LRU        C) LFU
7. Simulate the Following File Allocation Strategies
   A) Sequenced    B) Indexed    C) Linked

### Linux  Lab
1. Write a Shell program to check whether given number is prime or not.
2. Write a shell script which will display Fibonacci series up to the given range.
3. Write a shell script to check whether the given number is Armstrong or not.
4. Write a shell script to the calculate the value of $n_c^r$
5. Write a shell script to accept student number, name, marks in 5 subjects.
   Find total, average and grade using the following rules:
   Avg>=80 then grade A
   Avg<80&&Avg>=70 then grade B
   Avg<70&&Avg>=60 then grade C
   Avg<60&&Avg>=50 then grade D
   Avg<50&&Avg>=40 then grade E
6. Write a shell script to find minimum and maximum elements in the given list of elements.
7. Write a shell program to check whether the given string is palindrome or not.
8. Write an awk program to print sum, avg of students marks list

MCA III Semester
**Data Base Management Systems**

**Course Objectives:**
The objective of the course is to present an introduction to database management systems, with an emphasis on how to organize, maintain and retrieve efficiently, and effectively information from a DBMS.

**Course Outcomes (COs)**: At the end of the course, student will be able to
- Explain DBMS architecture, physical and logical database designs, database modelling, relational, hierarchical and network models.
- Identify basic database storage structures and access techniques such as file organizations, indexing methods including B-tree, and hashing
- Learn and apply Structured Query Language (SQL) for database definition and database manipulation
- Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database
- Understand various transaction processing, concurrency control mechanisms and database protection mechanisms
- Improve the database design by applying normalization techniques

**UNIT-I:**
**INTRODUCTION-**Database system, Characteristics (Database Vs File System), Database Users (Actors on Scene, Workers behind the scene), Advantages of Data base systems, Database applications, Brief introduction of different Data Models; Concepts of Schema, Instance and data independence; Three tier schema architecture for data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

**UNIT-II:**
**RELATIONAL MODEL:** Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, constraints (Domain, Key constraints, integrity constraints) and their importance **BASIC SQL:** Simple Database schema, data types, table definitions (create, alter), different DML operations (insert, delete, update), basic SQL querying (select and project) using where clause, arithmetic & logical operations, SQL functions (Date and Time, Numeric, String conversion).

**UNIT-III:**
**ENTITY RELATIONSHIP MODEL:** Introduction, Representation of entities, attributes, entity set, relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams. **SQL:** Creating tables with relationship, implementation of key and integrity constraints, nested queries, sub queries, grouping, aggregation, ordering, implementation of different types of joins, view(updatable and non-updatable), relational set operations.

**UNIT-IV:**
**SCHEMA REFINEMENT (NORMALIZATION):** Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency(1NF, 2NF and 3 NF), concept of surrogate key, Boyce-codd normal form(BCNF), Lossless join and dependency preserving decomposition, Fourth normal form(4NF).

**UNIT V:**
**TRANSACTION CONCEPT:** Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability, Failure Classification, Storage, Recovery and Atomicity, Recovery algorithm.
**INDEXING TECHNIQUES:** B+ Trees:  Search, Insert, Delete algorithms, File Organization and Indexing,  Cluster Indexes, Primary and Secondary Indexes , Index data Structures,  Hash Based Indexing: Tree base Indexing ,Comparison of File Organizations, Indexes and Performance Tuning

**Text Books:**
1. Data base Management Systems, 3rd    Edition, Raghu Ramakrishnan, Johannes Gehrke, TMH ,2003
2. Data base System Concepts, VI edition, A.Silberschatz, H.F. Korth, S.Sudarshan,    McGraw hill, 2006
3. Fundamentals of Database Systems, 5th edition, Ramez Elmasri, Shamkant
   Navathe, Pearson Education,2008

**Reference Books:**
1. Database Management System Oracle SQL and PL/SQL, P.K.Das Gupta,   PHI
2. Database System Concepts, Peter Rob & Carlos Coronel, Cengage Learning, 2008
3. Database Systems, A Practical approach to Design Implementation and Management Fourth edition, Thomas Connolly, Carolyn Begg, Pearson education
4. Database Principles, Programming, and Performance, P.O'Neil, E.O'Neil, 2nd
   ed., ELSEVIER
5. Fundamentals of Relational Database Management systems, S.Sumathi, S.Esakkirajan, Springer
6. Introduction to Database Management, Student Edition, M.L.Gillenson and others, Wiley
7. Database Development and Management, Lee Chao, Auerbach publications,
   Taylor & Francis Group
8. Introduction to Database Systems, C.J.Date, Pearson Education

MCA III Semester
**Computer Networks**

**Course Objectives:**
At the end of the course, the students will be able to:
- Understands the fundamental concepts of computer networking and OSI reference model.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the student for entry advanced courses in computer networking.
- Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Understand computer network basics, network architecture, TCP/IP and OSI reference models
- Identify and understand various techniques and modes of transmission
- Understand data link protocols, multi-channel access protocols and IEEE 802 standards for LAN
- Describe routing and congestion in network layer with routing algorithms and classify IPV4 addressing scheme
- Discuss the elements and protocols of transport layer
- Understand network security and define various protocols such as FTP, HTTP, Telnet, DNS

**UNIT-I:**
**Network Hardware reference model:** Transmission media, Narrowband ISDN, Broad band ISDN, ATM, **The data Link layer:** Design Issues, Error detection and correction, Elementary Data Link Protocols, Sliding window protocols : Data link layer in HDLC, Internet and ATM.

**UNIT-II:**
**Channel allocation methods:** TDM, FDM, ALOHA, Carrier sense Multiple access protocols, Collision Free protocols – IEEE standard 802.11 for LANS – Ethernet, Token Bus, Token ring, Bridges, **Network layer Routing Algorithms:** Shortest path, Flooding, Flow based Distance vector, Link state, Hierarchical, Broadcast routing, Congestion Control algorithms-General principles of congestion control, Congestion prevention polices, Choke packets and Load shedding.

**UNIT-III:**
**Internet Working:** Tunneling, internetworking, Fragmentation, network layer in the internet – IP protocols, IP address, Subnets, Internet control protocols, OSPF, BOP, Internet multicasting, Mobile IP, Network layer in the ATM Networks – cell formats, connection setup, routing and switching, service categories, and quality of service, ATM LANs.

**UNIT-IV:**

**The Transport Layer:** Elements of transport protocols – addressing, establishing a connection, releasing connection, flow control and buffering and crash recovery, end to end protocols: UDP, reliable Byte Stream (TCP) end to end format, segment format, connection establishment and termination, sliding window revisited, adaptive retransmission, TCP extension, Remote Procedure Call – BLAST, CHAN, SELECT, and DCE.

**UNIT-V:**

**Application Layer:** Network Security, Cryptographic Algorithms: DES, RSA. Security Mechanisms: Authentication Protocols, Firewalls. Name service (DNS) Domains Hierarchy, Name servers. Traditional Applications: SMTP, MIME, World Wide Web: HTTP, Network Management: SNMP.

**Text Books:**

1. Computer Networks, 4/e, Andrew Tanenbaum, Pearson
2. Data and computer communications, 8/e, stallings, PHI

**Reference Books:**

1. Data communications and networking, 4/e Forouzan, TMH
2. Computer Networks – A System Approach, 2/e, Peterson, Bruce Davie, and Harcourt Asia
3. Compute communications and networking technologies, Gallo, Hancock, and Cengage
4. An Engineering approach to compute networking, Kesha, Pearson
5. Communication networks, 2/e, Leon-Garcia, TMH
6. Computer networks, Anuranjan Misra, ACME Learning

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA
## KAKINADA – 533 003, Andhra Pradesh, India

MCA III Semester
**Design and Analysis of Algorithms**

**Course Objectives:**
This course aims to introduce the classic algorithms in various domains, and techniques for designing efficient algorithms

**Course Outcomes (COs)**: At the end of the course, student will be able to
- Explain the basic concepts of time and space complexity, divide-and-conquer Strategy, dynamic programming, greedy and approximate algorithms
- Describe the methodologies of how to analyze an algorithm
- Understands the data structures of graph colouring and back tracking
- Design a better algorithm to solve the problems
- To differentiate between tractable and intractable problems
- Ability to analyze the performance of algorithms.

**UNIT-I:**
**Introduction:** Algorithm, Pseudo code for expressing algorithms, performance Analysis-Space complexity, Time complexity, Asymptotic Notation- Big oh notation, Omega notation, Theta notation and Little oh notation, probabilistic analysis, Amortized analysis. Disjoint Sets- disjoint set operations, union and find algorithms, spanning trees, connected components and bi-connected components.

**UNIT-II:**
**Divide and conquer:** General method, applications-Binary search, Quick sort, Merge sort, Stassen's matrix multiplication. **Greedy method:** General method, applications-Job sequencing with deadlines, 0/1 knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

**UNIT-III:**
**Dynamic Programming:** General method, applications-Matrix chain multiplication, Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Travelling sales person problem, Reliability design.

**UNIT-IV:**
**Backtracking:** General method, applications-n-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

**UNIT-V:**
**Branch and Bound:** General method, applications - Travelling sales person problem, 0/1 knapsack problem- LC Branch and Bound solution, FIFO Branch and Bound solution. NP-Hard and NP-Complete problems: Basic concepts, non deterministic algorithms, NP - Hard and NP Complete classes, Cook's theorem.

**Text Books:**
1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharam, Universities Press
2. The Algorithm Design Manual, 2nd edition, Steven S. Skiena, Springer
3. Introduction to Algorithms, second edition, T.H.Cormen, C.E.Leiserson, R.L.Rivest and C.Stein, PHI Pvt. Ltd

**Reference Books:**
1. Introduction to the Design and Analysis of Algorithms, Anany Levitin, PEA
2. Design and Analysis of Algorithms, Pearson Education, Parag Himanshu Dave, Himansu Balachandra Dave
3. Introduction to Design and Analysis of Algorithms A strategic approach, R.C.T. Lee, S.S.Tseng, R.C.Chang and T.Tsai, Mc GrawHill.
4. Design and Analysis of algorithms, Pearson education, Aho, Ullman and Hopcroft

MCA III Semester
**Advanced Java Programming**

**Course Objectives:**
- To understand the concepts of naming and directory services.
- To learn distributed computing using RMI.
- To acquire knowledge on creation of software components using JAVA Beans.
- To learn Server-Side Programming using Servlets.
- To learn the creation of pure Dynamic Web Application using JDBC.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Apply the concepts of server side technologies for dynamic web applications
- Describe Lifecycle of a Servlet and its Package
- Explain about network and security programming using Java and know about the application of dynamic page functionality in web pages Servlets
- Invoke the remote methods in an application using Remote Method Invocation (RMI).
- Implement the multi-tier architecture of web-based enterprise applications using Enterprise JavaBeans (EJB)

**UNIT-I: Directory Services and JNDI:** Naming and Directory Services, Using JNDI, Java and LDAP, LDAP operations, Searching an LDAP Server

**UNIT-II: Distributed Computing Using RMI:** RMI Architecture, Locating Remote Objects and Developing Applications with RMI, Parameter passing in RMI, RMI firewalls and HTTP

**UNIT-III**
**Java Beans**: Introduction to Java Beans, Advantages of Java Beans, BDK Introspection, Using Bound properties, Bean Info Interface, Constrained properties Persistence, Customizes, Java Beans API, Introduction to EJB's, **EJB Architecture and Design:** EJB Container and its services, Working with EJBs, Design of the EJB Tier

**UNIT-IV:**
**Web Servers and Servlets:** Tomcat web server, Introduction to Servlets: Lifecycle of a Servlet, JSDK, The Servlet API, The javax.servlet Package, Reading Servlet parameters, Reading Initialization parameters. The javax.servlet HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues

**UNIT V:**
**Database Access:** Database Programming using JDBC, Studying Javax.sql.*package, **J2EE and Web Services:** Web Service Technologies (SOAP, WSDL, and UDDI), Developing Web services, making Services smarter.

**Text Books:**

1. Professional Java Server Programming, J2EE 1.3 edition, APRESS      publications (Units 1,2,4,8)
2. The complete Reference Java 8th Edition by Patrick Naughton and Herbert Schildt. TMH (Units 3,5,6,7)

**Reference Books:**

1. Programming world wide web-Sebesta, Pearson
2. Core SERVLETS ANDJAVASERVER PAGES VOLUME 1: CORE TECHNOLOGIES By Marty Hall and Larry Brown Pearson
3. Internet and World Wide Web – How to program by Dietel and Nieto PHI/Pearson Education Asia
4. Murach's beginning JAVA JDK 5, Murach, SPD
5. An Introduction to web Design and Programming –Wang-Thomson
6. Web Applications Technologies Concepts-Knuckles,John Wiley
7. Web Warrior Guide to Web Programmming-Bai/Ekedaw-Thomas
8. Beginning Web Programming-Jon Duckett WROX

MCA III Semester
**Object Oriented Analysis and Design**

**Course Objectives:**
- To understand the Object-based view of Systems
- To develop robust object-based models for Systems
- To inculcate necessary skills to handle complexity in software design

**Course Outcomes(COs)**: At the end of the course, student will be able to
- OOAD practices in developing software projects with management perspective
- Understand the object-oriented software development process, including object oriented methodologies
- Compare various object relationships like inheritance, association, whole-part and dependency relationships
- Compare various object relationships like inheritance, association, whole-part and dependency relationships
- Apply rational software suite for the construction of UML models and expressing the appropriate notations associated with each model
- Ability to analyze and model software specifications.

**UNIT- I:**
**Introduction to UML**: The meaning of Object-Orientation, object identity, encapsulation, information hiding, polymorphism, genericity, importance of modelling, principles of modelling, object oriented modelling, conceptual model of the UML, Architecture.

**UNIT-II:**
**Basic structural Modelling**: Classes, relationships, common mechanisms, diagrams, **Advanced structural modelling:** advanced relationships, interfaces, types & roles, packages, instances, **Class & object diagrams**: Terms, concepts, examples, modelling techniques, class & Object diagrams.

**UNIT-III:**
**Collaboration diagrams**: Terms, Concepts, depicting a message, polymorphism in collaboration diagrams, iterated messages, use of self in messages, **Sequence diagrams:** Terms, concepts, differences between collaboration and sequence diagrams, depicting synchronous messages with/without priority call back mechanism broadcast message.

**UNIT- IV:**
**Behavioral Modelling**: Interactions, use cases, use case diagrams, activity diagrams, **Advanced Behavioral Modelling**: Events and signals, state machines, processes & threads, time and space, state chart diagrams.

**UNIT–V:**
**Architectural Modelling:** Terms, concepts, examples, modelling techniques for component diagrams and deployment diagrams.

**Text Books:**
1. The Unified Modelling Language User Guide, Grady Booch, Rambaugh, Ivar Jacobson, PEA
2. Fundamentals of Object Oriented Design in UML, Meilir Page Jones, Addison Wesley

**Reference Books:**
1. Head First Object Oriented Analysis &Design, Mclaughlin, SPD O'Reilly, 2006
2. Object oriented Analysis& Design Using UML, Mahesh, PHI
3. The Unified Modeling Language Reference Manual, 2/e, Rambaugh, Grady Booch, etc., PEA
4. Object Oriented Analysis & Design, Satzinger, Jackson and Thomson
5. Object Oriented Analysis Design & Implementation, Dathan, Ramnath,   University Press
6. Object Oriented Analysis & Design, John Deacon, PEA
7. Fundamentals of Object Oriented Analysis and Design in UML, M Pages Jones, PEA
8. Object-Oriented Design with UML, Barclay, Savage, Elsevier, 2008

MCA III Semester
**Data Base Management Systems Lab**

**Course Objectives:**
- To design and implement a database schema for a given problem domain
- To create and manipulate tables using SQL queries
- To prepare reports
- To develop applications using PL/SQL

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Understand, appreciate and effectively explain the underlying concepts of database technologies
- Design and implement a database schema for a given problem-domain
- Apply normalization techniques to avoid redundancy
- Populate and query a database using SQL commands
- Declare and enforce integrity constraints on a database using a state-of-the-art RDBMS
- Write PL/SQL programs including stored procedures, stored functions, cursors, packages

**List of Experiments:**

1. Execute all DDL, DML and DCL commands on sample tables.
2. Implementation of different types of operators and  built-in functions  with suitable examples
3. Create views, partitions, Sequence, Indexes and locks for a particular DB.
4. Implement different types of constraints on relations.
5. Implementation of sub queries and nested queries.
6. Implementation of different types of joins  with suitable examples
7. Write PL/SQL block for an application using exception handling.
8. Write PL/SQL block for an application using cursors and all types of triggers.
9. Write a DBMS program to prepare Pl/SQL reports for an application using functions.
10. Write a PL/SQL block for transaction operations of a typical application using package.
11. Design and develop an application using any front end and back end tool (make use of ER diagram and DFD).

Typical Applications – Banking, Electricity Billing, Library Operation, Pay roll, Insurance, Inventory etc.

MCA III Semester
**Advanced Java Programming Lab**

**Course Objectives:**
It aims to introduce the students to some concepts of advanced programming and practice on reusing components.

- The course covers Graphical User Interface (GUI) networking, and database manipulation.
- Student will be able to use advanced technology in Java such as Internationalization, and Remote method Invocation
- Student will learn how to work with JavaBeans.
- Student will be able to develop web application using Java Servlet and Java Server Pages technology.

**Course Outcomes(COs)**: At the end of the course, student will be able to

- Implement sophisticated Java applications and well-organized, complex computer programs with both command line and graphical user interfaces
- Learn to access database through Java programs, using Java Data Base Connectivity (JDBC)
- Create dynamic web pages, using Servlets and JSP & make a reusable software component, using Java Bean
- Invoke the remote methods in an application using Remote Method Invocation (RMI)
- Understand the multi-tier architecture of web-based enterprise applications using Enterprise JavaBeans (EJB)

**List of Experiments**
1. Write a Program for Sending e-mail in Java.
2. Write a Program to implement Single Client-Server Communication.
3. Write a  Program to implement the Login_Id Form using JDBC
4. Write a  Program to implement the SQL commands using JDBC
5. Write a Java Program to implement Client – Server Communication using RMI.
6. Write a Java Servlet Program to implement and demonstrate get() and Post methods(Using HTTP Servlet Class).
7. Write a Program for Displaying Multiplication Table in Servlet for a Number Entered in Html Page
8. Write a  Program for Designing a Login Form Using Html and Displaying the Contents of the Login Form along with Date and Time in Servlet (Html to Servlet Communication)
9. Write a Java Servlet Program to Auto Web Page Refresh (Consider a webpage which is displaying Date and time or stock market status. For all such type of pages, you would need to refresh your web page regularly; Java Servlet makes this job easy by providing refresh automatically after a given interval).
10. Write a Program for Marks List Processing in Servlet with Records Taken from MS-Access (Servlet and JDBC connectivity)
11. Write a  Program Login Form Validation using JavaBeans
12. An EJB application that demonstrates Session Bean (with appropriate business logic).
13. An EJB application that demonstrates MDB (with appropriate business logic).
14. An EJB application that demonstrates persistence (with appropriate business logic).
15. Implement a program to execute selected query using JDBC.
16. Write a program to update the customer information using JDBC.

MCA III Semester
**OOAD through UML Lab**

**Course Objectives:**
- To know the practical issues of the different object oriented analysis and design concepts
- Inculcate the art of object oriented software analysis and design
- Apply forward and reverse engineering of a software system
- Carry out the analysis and design of a system in an object oriented way

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Know the syntax of different UML diagrams
- Create use case documents that capture requirements for a software system
- Create class diagrams that model both the domain model and design model of a software system.
- Create interaction diagrams that model the dynamic aspects of a software system.
- Write code that builds a software system.
- Develop simple applications

**Note:** For performing the experiments consider any case study (ATM/ Banking / Library /Hospital management systems)

**Experiment 1:**
Familiarization with Rational Rose or Umbrella environment

**Experiment 2:**
a) Identify and analyze events
b) Identify Use cases
c) Develop event table

**Experiment 3:**
a) Identify & analyze domain classes
b) Represent use cases and a domain class diagram using Rational Rose
c) Develop CRUD matrix to represent relationships between use cases and problem domain classes

**Experiment 4:**
a) Develop Use case diagrams
b) Develop elaborate Use case descriptions & scenarios
c) Develop prototypes (without functionality)

**Experiment 5:**
a) Develop system sequence diagrams and high-level sequence diagrams for each use case
b) Identify MVC classes / objects for each use case
c) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing interactions among all the three-layer objects

**Experiment 6:**
a) Develop detailed design class model (use GRASP patterns for responsibility assignment)
b) Develop three-layer package diagrams for each case study

**Experiment 7:**
a) Develop Use case Packages
b) Develop component diagrams
c) Identify relationships between use cases and represent them
d) Refine domain class model by showing all the associations among classes

**Experiment 8:**
Develop sample diagrams for other UML diagrams - state chart diagrams, activity diagrams and deployment diagrams

MCA IV Semester
**Data Warehousing and Data Mining**

**Course Objectives:**
- Be familiar with mathematical foundations of data mining tools..
- Understand and implement classical models and algorithms in data warehouses and data mining
- Characterize the kinds of patterns that can be discovered by association rule mining, classification and clustering.
- Master data mining techniques in various applications like social, scientific and environmental context.
- Develop skill in selecting the appropriate data mining algorithm for solving practical problems.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Explain the concept of Data Mining and its functionalities
- Discuss about Data Pre-processing Techniques like data cleaning, integration, transformation and reduction
- Illustrate need for data mining and in what ways it is different from traditional statistical techniques.
- Demonstrate Alternative techniques for Bayes' Theorem
- Demonstrate Association analysis techniques for generating association rules from data.

**UNIT-1:**
Introduction to Data mining, types of Data, Data Quality, Data Processing, Measures of Similarity and Dissimilarity, Exploring Data: Data Set, Summary Statistics, Visualization, OLAP and multi dimensional data analysis.

**UNIT-II:**
**Classification:** Basic Concepts, Decision Trees and model evaluation: General approach for solving a classification problem, Decision Tree induction, Model over fitting: due to presence of noise, due to lack of representation samples, Evaluating the performance of classifier. Nearest Neighborhood classifier, Bayesian Classifier, Support vector Machines: Linear SVM, Separable and Non Separable case.

**UNIT-III:**
**Association Analysis:** Problem Definition, Frequent Item-set generation, rule generation, compact representation of frequent item sets, FP-Growth Algorithms. Handling Categorical, Continuous attributes, Concept hierarchy, Sequential, Sub graph patterns

**UNIT-IV:**
**Clustering:** Over view, K-means, Agglomerative Hierarchical clustering, DBSCAN, Cluster evaluation: overview, Unsupervised Cluster Evaluation using cohesion and separation, using proximity matrix, Scalable Clustering algorithm

**UNIT-V:**

**Web data mining:** Introduction, Web terminology and characteristics, Web content mining, Web usage mining, web structure mining, Search Engines: Characteristics, Functionality, Architecture, Ranking of WebPages, Enterprise search

**Text Books:**

1. Introduction to Data Mining: Pang-Ning tan, Michael Steinbach, Vipin kumar, Addision-Wesley
2. Introduction to Data Mining with Case Studies: GK Gupta; Prentice Hall

**Reference Books:**

1. Data Mining: Introductory and Advanced Topics, Margaret H Dunham, Pearson, 2008

MCA IV Semester
**Full Stack Technologies**

**Course Objectives:**
From the course the student will learn
- Translate user requirements into the overall architecture and implementation of new systems and Manage Project and coordinate with the Client.
- Write backend code in Python/Java, PHP languages and Writing optimized front end code HTML and JavaScript.
- Understand, create and debug database related queries and Create test code to validate the applications against client requirement.
- Monitor the performance of web applications & infrastructure and Troubleshooting web application with a fast and accurate a resolution.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Identify the Basic Concepts of Web & Markup Languages
- Develop web Applications using Scripting Languages & Frameworks
- Creating & Running Applications using JSP libraries
- Creating Our First Controller Working with and Displaying in Angular Js and Nested Forms with ng-form
- Working with the Files in React JS and Constructing Elements with Data

**UNIT – I: HTML**
Web Essentials: Clients, Servers, and Communication. The Internet-Basic Internet Protocols -The World Wide Web-HTTP request message-response message-Web Clients Web Servers. Markup Languages: XHTML an Introduction to HTML, History, Versions, Basic, XHTML Syntax and Semantics Some Fundamental HTML Elements-Relative URLs-Lists-tables-Frames-Forms-HTML 5.0.

**UNIT – II: Cascading Style Sheets (CSS)**
Style Sheets: CSS-Introduction to Cascading Style Sheets-Features-Core Syntax-Style Sheets and HTML- Style Rule Cascading and Inheritance-Text Properties-Box Model Normal Flow Box Layout beyond the Normal Flow-CSS3.0, Boot strap basics, Boot strap CSS3, Introduction to Java Script, Jscript basics, JScripts objects, JSON, Don.

**UNIT – III: Jscript**
Separating Programming and Presentation: JSP Technology, Introduction to JSP and Servlets-Running JSP Applications, Basic JSP-JavaBeans Classes and JSP-Tag Libraries and Files-Support for the Model-View-Controller Paradigm- Mongo DB, JQuery, Mean stack Fundamentals

**UNIT – IV: Angular Js**
Introducing AngularJS, Starting Out with AngularJS, Basic AngularJS, Directives and Controllers, AngularJS Modules, Creating First Controller, working with and Displaying, Arrays, more Directives, working with ng-repeat, Unit Testing in AngularJS, Forms, Inputs, and Services, Working with ng-model, Working with Forms, Leverage Data-Binding and Models, Form Validation and States, Error Handling with Forms, ngModelOptions, Nested Forms with ng-form, Other Form Controls.

**UNIT – V: React JS**
Introduction to React, Obstacles and Roadblocks, keeping Up with the Changes, Working with the Files, Pure React, Page Setup, The Virtual DOM, React Elements, ReactDOM, Children, Constructing Elements with Data, React Components, DOM Rendering, Factories

**Text Books:**
1. Jeffrey C. Jackson, "Web Technologies--A Computer Science Perspective", Pearson Education, 2006
2. Robert. W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson Education, 2007
3. AngularJS: Up and Running Enhanced Productivity with Structured Web Apps By Brad Green, Shyam Seshadri Publisher: O'Reilly Media
4. Learning React Functional Web Development with React and Redux By Alex_Banks, Eve Porcello Publisher: O'Reilly Media
5. Head First Java, 2nd Edition by Bert Bates, Kathy Sierra Publisher: O'Reilly Media, Inc

MCA IV Semester
**Python Programming**

**Course Objectives:**
- To acquire programming skills in core Python.
- To acquire Object Oriented Skills in Python.
- To develop the skill of designing Graphical user Interfaces in Python.
- To develop the ability to write database applications in Python.

**Course Outcomes(COs):** At the end of the course, student will be able to
- Understand and comprehend the basics of python programming.
- Demonstrate the principles of structured programming and be able to describe, design, implement, and test structured programs using currently accepted methodology.
- Explain the use of the built-in data structures list, sets, tuples and dictionary.
- Make use of functions and its applications.
- Identify real-world applications using oops, files and exception handling provided by python.

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

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

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

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

**UNIT – V:**
**Object Oriented Programming OOP in Python**: Classes, 'self variable', Methods, Constructor Method, Inheritance, Overriding Methods, Data hiding, **Error and Exceptions:** Difference between an error and Exception, Handling Exception, try except block, Raising Exceptions, User defined Exceptions, **Brief Tour of the Standard Library** - Operating System Interface - String Pattern Matching, Mathematics, Internet Access, Dates and Times, Data Compression, Multithreading, GUI Programming, Turtle Graphics, **Testing:**  Why testing is required ?, Basic concepts of testing, Unit testing in Python, Writing Test cases, Running Tests.

**Text books**

1. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
2. Learning Python, Mark Lutz, Orielly

**Reference Books:**

1. Think Python, Allen Downey, Green Tea Press
2. Core Python Programming, W.Chun, Pearson
3. Introduction to Python, Kenneth A. Lambert, Cengage

MCA IV Semester
## Embedded Computing

**Course Objectives:**
- Develop an understanding of the technologies behind embedded computing systems.
- Introduce students to the various software components invloved in embedded system design and development.
- Expose students to the recent trends in embedded system design.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Demonstrate the role of individual components involved in a typical embedded system.
- Analyze the characteristics of different computing elements and select the most appropriate one for an embedded system.
- Model the operation of a given embedded system.
- Substantiate the role of different software modules in the development of an embedded system.
- Develop simple tasks to run on an RTOS. 6. Examine the latest trends prevalent in embedded system design.

### UNIT-I:
Introduction to Embedded System: Understanding the Basic Concepts, The Typical Embedded System – Characteristics and Quality attributes

### UNIT-II:
Hardware Software Co-Design and Program Modelling – Fundamental Issues, Computational Models- Data Flow Graph, Control Data Flow Graph, State Machine, Sequential Model, Concurrent Model, Object oriented model, UML

### UNIT-III:
Design and Development of Embedded Product – Firmware Design and Development – Design Approaches, Firmware Development Languages

### UNIT-IV:
Integration and Testing of Embedded Hardware and Firmware- Integration of Hardware and Firmware. Embedded System Development Environment – IDEs, Cross Compilers, Disassemblers, Decompilers, Simulators, Emulators and Debuggers.

### UNIT-V:
RTOS based Design – Basic operating system services. Interrupt handling in RTOS environment. Design Principles. Task scheduling models. How to Choose an RTOS. Embedded Product Development Life Cycle – Description – Objectives -Phases – Approaches. Recent Trends in Embedded Computing.

**Text Books:**
1. Shibu K.V., Introduction to Embedded Systems, McGraw Hill Education (India), 2009.
2. Raj Kamal, Embedded Systems: Architecture, Programming and Design, Third Edition, McGraw Hill Education (India).

**Reference Book:**
1. Jean J. Labrose, MicroC OS II: The Real Time Kernel, Second Edition, CRC Press.
2. Steave Heath, Embedded System Design, Second Edition, Elsevier.
3. J Staunstrup and Wayne Wolf, Hardware / Software Co-Design: Principles and Practice, Prentice Hall.

MCA IV Semester
**Artificial Intelligence**

**Course Objectives:**
- Gain a historical perspective of AI and its foundations.
- Become familiar with basic principles of AI toward problem solving, inference, perception, knowledge representation, and learning.
- Investigate applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
- Experience AI development tools such as an 'AI language', expert system shell, and/or data mining tool. Experiment with a machine learning model for simulation and analysis.
- Explore the current scope, potential, limitations, and implications of intelligent systems.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents
- Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game based techniques to solve them.
- Develop intelligent algorithms for constraint satisfaction problems and also design intelligent systems for Game Playing
- Attain the capability to represent various real life problem domains using logic based techniques and use this to perform inference or planning.
- Solve problems with uncertain information using Bayesian approaches.

**UNIT-I:**
**Introduction to artificial intelligence:** Introduction , history, intelligent systems, foundations of AI, applications, tic-tac-tie game playing, development of AI languages, current trends in AI, **Problem solving: state-space search and control strategies:** Introduction, general problem solving, characteristics of problem, exhaustive searches, heuristic search techniques, iterative-deepening a*, constraint satisfaction

**UNIT-II:**
**Problem reduction and game playing:** Introduction, problem reduction, game playing, alpha-beta pruning, two-player perfect information games, **Logic concepts:** Introduction, propositional calculus, proportional logic, natural deduction system, axiomatic system, semantic tableau system in proportional logic, resolution refutation in proportional logic, predicate logic

**UNIT-III:**
**Knowledge representation:** Introduction, approaches to knowledge representation, knowledge representation using semantic network, extended semantic networks for KR, knowledge representation using frames, **advanced knowledge representation techniques:** Introduction, conceptual dependency theory, script structure, cyc theory, case grammars, semantic web, **Expert system and applications:** Introduction phases in building expert systems, expert system versus traditional systems, rule-based expert systems blackboard systems truth maintenance systems, application of expert systems, list of shells and tools

## UNIT-IV:

**Uncertainty measure: probability theory:** Introduction, probability theory, Bayesian belief networks, certainty factor theory, dempster-shafer theory , **Fuzzy sets and fuzzy logic:** Introduction, fuzzy sets, fuzzy set operations, types of membership functions, multi valued logic, fuzzy logic, linguistic variables and hedges, fuzzy propositions, inference rules for fuzzy propositions, fuzzy systems.

## UNIT-V:

**Machine learning paradigms:** Introduction, machine learning systems, supervised and unsupervised learnings, inductive learning, deductive learning, clustering, support vector machines, case based reasoning and learning, **Artificial neural networks:** Introduction, artificial networks, single layer feed forward networks, multi layered forward networks, design issues of artificial neural networks

**Text Books:**
1. Artificial Intelligence- Saroj Kaushik, CENGAGE Learning,
2. Artificial intelligence, A modern Approach, 2nd ed, Stuart Russel, Peter Norvig, PEA
3. Artificial Intelligence- 3rd ed, Rich, Kevin Knight, Shiv Shankar B Nair, TMH
4. Introduction to Artificial Intelligence, Patterson, PHI

**Reference Books:**
1. Artificial intelligence, structures and Strategies for Complex problem solving,
   5th ed, George F Lugar, PEA
2. Introduction to Artificial Intelligence, Ertel, Wolf Gang, Springer
3. Artificial Intelligence, A new Synthesis, Nils J Nilsson, Elsevier

MCA IV Semester
**Cloud Computing**

**Course Objectives:**
- To implement Virtualization
- To implement Task Scheduling algorithms.
- Apply Map-Reduce concept to applications.
- To build Private Cloud.
- Broadly educate to know the impact of engineering on legal and societal issues involved.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Interpret the key dimensions of the challenge of Cloud Computing
- Examine the economics, financial, and technological implications for selecting cloud computing for own organization.
- Assessing the financial, technological, and organizational capacity of employer's for actively initiating and installing cloud-based applications
- Evaluate own organizations' needs for capacity building and training in cloud computing-related IT areas.
- To Illustrate Virtualization for Data-Center Automation.

**UNIT I:**
**Introduction:** Network centric computing, Network centric content, peer-to –peer systems, cloud computing delivery models and services, Ethical issues, Vulnerabilities, Major challenges for cloud computing. **Parallel and Distributed Systems:** introduction, architecture, distributed systems, communication protocols, logical clocks, message delivery rules, concurrency, and model concurrency with Petri Nets.

**UNIT II:**
**Cloud Infrastructure:** At Amazon, The Google Perspective, Microsoft Windows Azure, Open Source Software Platforms, Cloud storage diversity, Inter cloud, energy use and ecological impact, responsibility sharing, user experience, Software licensing, **Cloud Computing :** Applications and Paradigms: Challenges for cloud, existing cloud applications and new opportunities, architectural styles, workflows, The Zookeeper, The Map Reduce Program model, HPC on cloud, biological research

**UNIT III:**
**Cloud Resource virtualization:** Virtualization, layering and virtualization, virtual machine monitors, virtual machines, virtualization- full and para, performance and security isolation, hardware support for virtualization, Case Study: Xen, vBlades, **Cloud Resource Management and Scheduling**: Policies and Mechanisms, Applications of control theory to task scheduling, Stability of  a two-level resource allocation architecture, feedback control based on dynamic thresholds, coordination, resource bundling, scheduling algorithms, fair queuing, start time fair queuing, cloud scheduling subject to deadlines, Scheduling Map Reduce applications, Resource management and dynamic application scaling

**UNIT IV:**
**Storage Systems:** Evolution of storage technology, storage models, file systems and database, distributed file systems, general parallel file systems. Google file system. Apache Hadoop, Big Table, Megastore (text book 1), Amazon Simple Storage Service(S3) (Text book 2), **Cloud Security:** Cloud security risks, security – a top concern for cloud users, privacy and privacy impact assessment, trust, OS security, Virtual machine security, Security risks

**UNIT V:**
**Cloud Application Development:** Amazon Web Services : EC2 – instances, connecting clients, security rules, launching, usage of S3 in Java, Installing Simple Notification Service on Ubuntu 10.04, Installing Hadoop on Eclipse, Cloud based simulation of a Distributed trust algorithm, Cloud service for adaptive data streaming ( Text Book 1), **Google:** Google App Engine, Google Web Toolkit (Text Book 2), **Microsoft:**  Azure Services Platform, Windows live, Exchange Online, Share Point Services, Microsoft Dynamics CRM (Text Book 2)

**Text Books:**
1. Cloud Computing, Theory and Practice, Dan C Marinescu, MK Elsevier
2. Cloud Computing, A Practical Approach, Anthony T Velte, Toby J Velte, Robert Elsenpeter, TMH

**Reference book:**
1. Mastering Cloud Computing, Foundations and Application Programming, Raj Kumar Buyya, Christen vecctiola, S Tammarai selvi, TMH

MCA IV Semester
**Multimedia Application Development**

**Course Objective:**
The objective is to
- To analyze and explain various technologies involved to support multimedia application development.
- To understand multimedia authoring and Understanding the constraints on multimedia systems and the range of technologies available to multimedia systems designers and integrators.
- To understand how the quality of multimedia systems is perceived and how this Relate to the design of multimedia input, output and editing systems.
- To understand different compression principles and different compression technique and to know the mathematics involved in digital and analog conversion of components of multimedia.
- To design and develop multimedia systems according to the requirements of multimedia application and understand the particular issues of virtual reality.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Design and implementation of hypermedia & multimedia systems.
- Achieve an in-depth understanding of the impact of multimedia on personal and distributed computer systems, the range of media types and tools to support their digital conversion and manipulation of images, audio and video and their compression.
- Develop an interactive multimedia application to display their ability to use multimedia tools including multimedia authoring.
- Evaluate and Discuss Excellent and ineffective interactive multimedia design.
- Design and implement a number of multimedia network applications and Evaluate and discuss virtual reality systems.

**UNIT- I:**
**Fundamental concepts in Text and Image:** Multimedia and hypermedia, World Wide Web, overview of multimedia software tools. Graphics and image data representation graphics/image data types, file formats, **Color in image and video:** color science, color models in images, color models in video.

**UNIT- II:**
**Fundamental Concepts in Video and Digital Audio:** Types of video signals, analog video, and digital video, digitization of sound, MIDI, quantization and transmission of audio.

**UNIT-III:**
**Action Script I:** Action Script Features, Object-Oriented Action Script, Data types and Type Checking, Classes, Authoring an Action Script Class, **Action Script II:** Inheritance, Authoring an Action Script 2.0 Subclass, Interfaces, Packages, Exceptions, **Application Development:** An OOP Application Frame work, Using Components with Action Script Movie Clip Subclasses.

## UNIT- IV

**Multimedia Data Compression:** Lossless compression algorithm: Run-Length Coding, Variable Length Coding, Dictionary Based Coding, Arithmetic Coding, Lossless Image Compression, Lossy compression algorithm: Quantization, Transform Coding, Wavelet-Based Coding, Embedded Zero tree of Wavelet Coefficients Set Partitioning in Hierarchical Trees (SPIHT), **Basic Video Compression Techniques:** Introduction to video compression, video compression based on motion compensation, search for motion vectors, MPEG, Basic Audio Compression Techniques.

## UNIT-V

**Multimedia Networks:** Basics of Multimedia Networks, Multimedia Network Communications and Applications: Quality of Multimedia Data Transmission, Multimedia over IP, Multimedia over ATM networks, Transport of MPEG- 4, Media-on- Demand (MOD).

**Text Books:**
1. Fundamentals of Multimedia, Ze-Nian Li , Mark S. Drew, PHI/ PEA
2. Multimedia Systems, Parag Havaldar, Gerard Medioni, cengage, 2009
3. Essentials Action Script 3.0, Colin Moock, SPD O, Reilly, 2007

**Reference Books:**
1. Multimedia Applications, Steinmetz, Nahrstedt, Springer
2. Digital Multimedia, Nigel Chapman, Jenny Chapman, Wiley- Dreamtech
3. Multimedia & Communications Technology, Steve Heath, Elsevier
4. Multimedia Technology & Applications, David Hilman , Galgotia
5. Multimedia Technologies, Banerji, Mohan Ghosh,MGH

MCA IV Semester
**Full Stack Technologies Lab**

**Course Objectives:**
From the course the student will
- Learn the core concepts of both the frontend and backend programming course.
- Get familiar with the latest web development technologies.
- Learn all about SQL and Mongo databases.
- Learn complete web development process.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Identify the Basic Concepts of Web & Markup Languages
- Develop web Applications using Scripting Languages & Frameworks
- Creating & Running Applications using JSP libraries
- Creating Our First Controller Working with and Displaying in Angular Js and Nested Forms with ng-form
- Working with the Files in React JS and Constructing Elements with Data

**List of Experiments:**

1. Implementation of 'get' and 'post' methods.
2. CSS implementation in colors, boarder padding.
3. CSS3 implementation button frames tables, navigation bars.
4. Create registration and login forms with validations using Jscript query.
5. Jscript to retrieve student information from student database using database connectivity.
6. Angular Js data binding
7. Angular JS directives and Events
8. Using angular Js fetching data from MySQL.
9. Using React Js creating constructs data elements.
10. Using React Js implementations DoM
11. Invoking data using Jscript from Mongo DB.
12. Create an Online fee payment form using JSCript and MangoDB

**Reference/ Preferred Text Books:**
1. Jeffrey C. Jackson, "Web Technologies--A Computer Science Perspective", Pearson Education, 2006
2. Robert. W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson Education, 2007
3. Angular JS: Up and Running Enhanced Productivity with Structured Web Apps By Brad Green, Shyam Seshadri Publisher: O'Reilly Media
4. Learning React Functional Web Development with React and Redux By Alex Banks, Eve Porcello Publisher: O'Reilly Media
5. Head First Java, 2nd Edition by Bert Bates, Kathy Sierra Publisher: O'Reilly Media, Inc

MCA IV Semester
**Data Mining with R Lab**

**Course Objectives:**
- To understand the mathematical basics quickly and covers each and every condition of data mining in order to prepare for real-world problems.
- The various classes of algorithms will be covered to give a foundation to further apply knowledge to dive deeper into the different flavors of algorithms.
- Students should aware of packages and libraries of R and also familiar with functions used in R for visualization.
- To enable students to use R to conduct analytics on large real life datasets.
- To familiarize students with how various statistics like mean median etc. can be collected for data exploration in R.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Extend the functionality of R by using add-on packages
- Extract data from files and other sources and perform various data manipulation tasks on them.
- Code statistical functions in R
- Use R Graphics and Tables to visualize results of various statistical operations on data
- Apply the knowledge of R gained to data Analytics for real life applications

**List of Experiments:**
1. Implement all basic R commands.
2. Interact data through .csv files (Import from and export to .csv files).
3. Get and Clean data using swirl excercises.(Use 'swirl' package, library and install that topic from swirl).
4. Visualize all Statistical measures (Mean,Mode, Median, Range, Inter Quartile Range etc., using Histograms, Boxplots and Scatter Plots).
5. Create a data frame with the following structure.

| EMP ID | EMP NAME | SALARY | START DATE |
|--------|----------|--------|------------|
| 1 | Satish | 5000 | 01-11-2013 |
| 2 | Vani | 7500 | 05-06-2011 |
| 3 | Ramesh | 10000 | 21-09-1999 |
| 4 | Praveen | 9500 | 13-09-2005 |
| 5 | Pallavi | 4500 | 23-10-2000 |

   a. Extract two column names using column name.
   b. Extract the first two rows and then all columns.
   c. Extract $3^{rd}$ and $5^{th}$ row with $2^{nd}$ and $4^{th}$ column.
6. Write R Program using 'apply' group of functions to create and apply normalization function on each of the numeric variables/columns of iris dataset to transform them into
   i. 0 to 1 range with min-max normalization.
   ii. a value around 0 with z-score normalization.
7. Create a data frame with 10 observations and 3 variables and add new rows and columns to it using 'rbind' and 'cbind' function.

8. Write R program to implement linear and multiple regression on 'mtcars' dataset to estimate the value of 'mpg' variable, with best $R^2$ and plot the original values in 'green' and predicted values in 'red'.
9. Write R Programs to implement k-means clustering.
10. Write R Programs to implement k-medoids clustering.
11. Write R Programs to implement density based clustering on iris dataset.
12. Write a R Program to implement decision trees using 'readingSkills' dataset.
13. Implement decision trees using 'iris' dataset using package party and 'rpart'.
14. Use a Corpus() function to create a data corpus then Build a term Matrix and Reveal word frequencies.

**Reference Books:**
1. www.tutorialspoint.com/r
2. www.r-tutor.com
3. R and Data Mining: Examples and Case Studies Yanchang Zhao.
4. https://towardsdatascience.com/

MCA IV Semester
**Python Programming Lab**

**Course Objectives:**
- To acquire programming skills in core Python.
- To acquire Object Oriented Skills in Python.
- To develop the skill of designing Graphical user Interfaces in Python.
- To develop the ability to write database applications in Python.

**Course Outcomes(COs):** At the end of the course, student will be able to
- Understand and comprehend the basics of python programming.
- Demonstrate the principles of structured programming and be able to describe, design, implement, and test structured programs using currently accepted methodology.
- Explain the use of the built-in data structures list, sets, tuples and dictionary.
- Make use of functions and its applications.
- Identify real-world applications using oops, files and exception handling provided by python.

**List of Experiments:**
**Experiment 1 - Basics**
a) Running instructions in Interactive interpreter and a Python Script
b) Write a program to purposefully raise Indentation Error and Correct it

**Experiment 2 - Operations**
a) Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)
b) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.

**Experiment - 3 Control Flow**
a) Write a Program for checking whether the given number is a even number or     not.
b) Write a program using a for loop that loops over a sequence.
c) Find the sum of all the primes below two million. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:
   1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

**Experiment 4 -  DS**
a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure
b) Write a program to use split and join methods  in the string and trace a birthday with a dictionary data structure.

**Experiment - 5 DS -   Continued**
a) Write a program combine_lists that combines these lists into a dictionary.
b) Write a program to count frequency of characters in a given file. Can you use
   character frequency to tell whether the given file is a Python program file, C     program file or a text file?

**Experiment - 6 Files**

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

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

**Experiment -  7 Functions**

a) Write a function ball_collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding. Hint:  Represent a ball on a plane as a tuple of (x, y, r), r being the radius. If (distance between two balls centers) <= (sum of their radii) then (they are colliding)

b) Find mean, median, mode for the given set of numbers in a list.

**Experiment - 8  Functions - Continued**

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

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

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

**Experiment - 9 - Functions - Problem Solving**

a) Write a function cumulative_product to compute cumulative product of a list
 of numbers.

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

c) Write function to compute gcd, lcm of two numbers. Each function shouldn't
   exceed one line.

**Experiment 10 - Multi-D Lists**

a)  Write a program that defines a matrix and prints

b)  Write a program to perform addition of two square matrices

c)  Write a program to perform  multiplication of two square matrices

**Experiment 11 -  Modules**

a)  Install packages requests, flask and explore them using (pip)

b)  Write a script that imports requests and fetch content from the page.

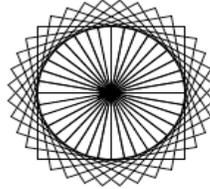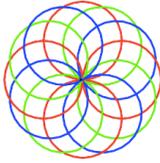c)  Write a simple script that serves a simple HTTPResponse and a simple HTML Page

**Experiment 12- OOP**

a)  Class variables and instance variable and illustration of the self variable
   i)  Robot                ii) ATM Machine

**Experiment 13 - GUI, Graphics**

a) Write a GUI for an Expression Calculator using tk

b) Write a program to implement the following figures using turtle



**Experiment 14 - Testing**

a) Write a test-case to check the function even _numbers which return True on passing a list of all even numbers

b) Write a test-case to check the function reverse_string which returns the     reversed string

**Experiment 15 - Advanced**

a) Build any one classical data structure.

b) Write a program to solve knapsack problem.

MCA V Semester
## Cryptography and Network Security

**Course Objectives:**
The main objectives of this course are toexplore the working principles and utilities of various cryptographic algorithms including secret key cryptography, hashes and message digests, public key algorithms, design issues and working principles of various authentication protocols and various secure communication standards including Kerberos, IPsec, and SSL/TLS.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Explain different security threats and countermeasures and foundation course of cryptography mathematics.
- Classify the basic principles of symmetric key algorithms and operations of some symmetric key algorithms and asymmetric key cryptography
- Revise the basic principles of Public key algorithms and Working operations of some Asymmetric key algorithms such as RSA, ECC and some more
- Design applications of hash algorithms, digital signatures and key management techniques
- Determine the knowledge of Application layer, Transport layer and Network layer security Protocols such as PGP, S/MIME, SSL,TSL, and IPsec .

**UNIT I:**
**Basic Principles :** Security Goals, Cryptographic Attacks, Services and Mechanisms, Mathematics of Cryptography.

**UNIT II:**
**Symmetric Encryption:** Mathematics of Symmetric Key Cryptography, Introduction to Modern Symmetric Key Ciphers, Data Encryption Standard, Advanced Encryption Standard.

**UNIT III:**
**Asymmetric Encryption:** Mathematics of Asymmetric Key Cryptography, Asymmetric Key Cryptography

**UNIT IV:**
**Data Integrity, Digital Signature Schemes & Key Management :** Message Integrity and Message Authentication, Cryptographic Hash Functions, Digital Signature, Key Management.

**UNIT V:**
**Network Security-I:** Security at application layer: PGP and S/MIME, Security at the Transport Layer: SSL and TLS, **Network Security-II :** Security at the Network Layer: IPSec, System Security

**Text Books:**
1. Cryptography and Network Security, 3$^{rd}$ Edition Behrouz A Forouzan, Deb deep Mukhopadhyay, McGraw Hill
2. Cryptography and Network Security, William Stallings, (6e) Pearson.
3. Everyday Cryptography, Keith M.Martin, Oxford

**Reference Books:**
1. Network Security and Cryptography, Bernard Meneges, Cengage Learning

MCA V Semester
**Big Data Analytics**

**Course Objectives:**
The main objective of this course is to
- Provide an overview of an exciting growing field of Big Data analytics
- Introduce the tools required to manage and analyze big data like Hadoop, MapReduce etc.,

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Understand the programming requirements viz., generic types and methods to perform data analysis
- Understand the existing technologies and the need of distributed files systems to analyze the big data
- To understand and analyze Map-Reduce programming model for better optimization
- Collect, manage, store, query, and analyze big data; and identify the need of interfaces to perform I/O operations in Hadoop
- Identify the need based tools, viz., Pig and Hive and to handle
- Formulate an effective strategy to implement a successful Data analytics project

**UNIT-I:**
**Data structures in Java:** Linked List, Stacks, Queues, Sets, Maps; Generics: Generic classes and Type parameters, Implementing Generic Types, Generic Methods, Wrapper Classes, Concept of Serialization

**UNIT-II:**
**Working with Big Data:**  Google File System, Hadoop Distributed File System (HDFS), Building blocks of Hadoop (Name node, Data node, Secondary Name node, Job Tracker, Task Tracker), Introducing and Configuring Hadoop cluster (Local, Pseudo-distributed mode, Fully Distributed mode), Configuring XML files.

**UNIT-III:**
**Writing Map Reduce Programs:** A Weather Dataset, Understanding Hadoop API for Map Reduce Framework (Old and New), **Basic programs of Hadoop Map Reduce:** Driver code, Mapper code, Reducer code, Record Reader, Combiner, Partitioner

**UNIT-IV**
**Hadoop I/O:** The Writable Interface, Writable Comparable and comparators, **Writable Classes**: Writable wrappers for Java primitives, Text, Bytes Writable, Null Writable, Object Writable and Generic Writable, Writable collections, Implementing a Custom Writable: Implementing a Raw Comparator for speed, Custom comparators

**UNIT-V**
**Pig:** Hadoop Programming Made Easier, Admiring the Pig Architecture, Going with the Pig Latin Application Flow, Working through the ABCs of Pig Latin, Evaluating Local and Distributed Modes of Running Pig Scripts, Checking out the Pig Script Interfaces, Scripting with Pig Latin, **Applying Structure to Hadoop Data with Hive:** Saying Hello to Hive, Seeing How the Hive is Put Together, Getting Started with Apache Hive, Examining the Hive Clients, Working with Hive Data Types, Creating and Managing Databases and Tables, Seeing How the Hive Data Manipulation Language Works, Querying and Analyzing Data

**Text Books:**
1. Big Java 4th Edition, Cay Horstmann, Wiley John Wiley & Sons, INC
2. Hadoop: The Definitive Guide by Tom White, 3$^{rd}$ Edition, O'Reilly
3. Hadoop in Action by Chuck Lam, MANNING Publ
4. Hadoop for Dummies by Dirk deRoos, Paul C.Zikopoulos, Roman B.Melnyk,Bruce Brown and Rafael Coss

**Reference Books:**
1. Hadoop in Practice by Alex Holmes, MANNING Publ
2. Hadoop MapReduce Cookbook, Srinath Perera, Thilina Gunarathne

**Web Resources:**
1. Hadoop:https://hadoop.apache.org/
2. Hive: https://cwiki.apache.org/confluence/display/Hive/Home/
3. Piglatin: https://pig.apache.org/docs/r0.7.0/tutorial.html

MCA V Semester
**Machine Learning**

**Course Objectives:**
- Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
- Analyze a given problem in the language/framework of different AI methods (e.g., as a search problem, as a constraint satisfaction problem, as a planning problem, as a Markov decision process, etc).
- Implement basic AI algorithms (e.g., standard search algorithms or dynamic programming).
- Design and carry out an empirical evaluation of different algorithms on problem formalization, and state the conclusions that the evaluation supports.

**Course Outcomes:**
After the completion of the course, student will be able to
- Explain the definition and usage of the term 'the internet of things' in different contexts.
- Demonstrate on various network protocols used in IoT.
- Analyze on various key wireless technologies used in IoT systems, such as WiFi, 6LoWPAN, Bluetooth and ZigBee.
- Illustrate on the role of big data, cloud computing and data analytics in IoT system.
- Design a simple IoT system made up of sensors, wireless network connection, data analytics and display/actuators, and write the necessary control software.

**Unit-I: Introduction-**Towards Intelligent Machines, Well posed Problems, Example of Applications in diverse fields, Data Representation, Domain Knowledge for Productive use of Machine Learning, Diversity of Data: Structured / Unstructured, Forms of Learning, Machine Learning and Data Mining, Basic Linear Algebra in Machine Learning Techniques.

**Unit-II: Supervised Learning-** Rationale and Basics: Learning from Observations, Bias and Why Learning Works: Computational Learning Theory, Occam's Razor Principle and Over fitting Avoidance Heuristic Search in inductive Learning, Estimating Generalization Errors, Metrics for assessing regression, Metris for assessing classification.

**Unit-III: Statistical Learning-** Machine Learning and Inferential Statistical Analysis, Descriptive Statistics in learning techniques, Bayesian Reasoning: A probabilistic approach to inference, K-Nearest Neighbor Classifier. Discriminant functions and regression functions, Linear Regression with Least Square Error Criterion, Logistic Regression for Classification Tasks, Fisher's Linear Discriminant and Thresholding for Classification, Minimum Description Length Principle.

**Unit-IV: Support Vector Machines (SVM)-** Introduction, Linear Discriminant Functions for Binary Classification, Perceptron Algorithm, Large Margin Classifier for linearly seperable data, Linear Soft Margin Classifier for Overlapping Classes, Kernel Induced Feature Spaces, Nonlinear Classifier, Regression by Support vector Machines.

**Learning with Neural Networks:** Towards Cognitive Machine, Neuron Models, Network Architectures, Perceptrons, Linear neuron and the Widrow-Hoff Learning Rule, The error correction delta rule.

**Unit -V:** Multilayer Perceptron Networks and error back propagation algorithm, Radial Basis Functions Networks. **Decision Tree Learning**: Introduction, Example of classification decision tree, measures of impurity for evaluating splits in decision trees, ID3, C4.5, and CART decision trees, pruning the tree, strengths and weakness of decision tree approach.

**Textbooks:**
1. Applied Machine Learning, M. Gopal, McGraw Hill Education
2. Machine Learning: A Probabilistic Perspective, Kevin Murphy, MIT Press,2012
3. The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer 2009 (freely available online)

**Reference Books:**
1. Pattern Recognition and Machine Learning, Christopher Bishop, Springer,2007
2. Programming Collective Intelligence: Building Smart Web 2.0 Applications - Toby Segaran
3. Building Machine Learning Systems with Python - WilliRichert, Luis Pedro Coelho

MCA V Semester
**Digital Marketing**

**Course Objectives:**
Digital marketing aims at being SMART (Specific, Measurable, Achievable, Relevant and Time Related) so that people can withstand against competitors.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Explain about web pages with basic HTML5, DHTML tags using CSS and XML, the overview of W3C DOM.
- Discuss the key elements of a digital Java Scripts.
- Apply search engine optimization techniques to a website.
- Illustrate how the effectiveness of a digital marketing campaign can be measured
- Demonstrate advanced practical skills in common digital marketing tools such as SEO, SEM, Social media and Blogs

**UNIT I:**
**HTML:** Introduction, HTML5, Audio Elements, Video Elements, Organizing Elements.
**Scripting Documents:** Dynamic Document content, Document properties, Legacy DOM, Document Collections, Overview of the W3C DOM, Traversing a Document, Finding Elements in a Document, Modifying a Document, Adding Content to a Document Example
**UNIT II:**
**Cascading Style Sheets and Dynamic HTML:** Overview of CSS, CSS for DHTML Scripting inline Styles, Scripting computed styles, Scripting CSS Classes, Scripting Style Sheets, **Java Script and XML:** Obtaining XML Documents, Manipulating XML with the DOM API, Transforming XML with XSLT querying XML with X path, Serializing XML, Example, XML and Web services.
**UNIT III:**
**Search Engine Optimization (SEO):** Searching Engine Marketing, Search Engine Optimization, Measuring SEO Success, Mapping with SEO Journey, **Search Advertising:** Online Advertising Payment Models, Search Advertising (Desktop & Mobile Devices), Planning & Executing a search Advertising Camping, Strategic Implications of Advertising on the search Network.
**UNIT IV:**
**Search Media Marketing:** What is Social Media? Social Media Marketing, Social Media Marketing Strategy, Adopting Social Media in Organizations: Internal Learning, Paid-Owned-Earned Media, Social CRM, **Mobile Marketing:**
Mobile Internet in India, What is Mobile Marketing? Email Marketing Strategy, Forms of Mobile Marketing, Mobile Advertising, M-Commerce.
**UNIT V:**
**E-Mail Marketing:** E-Mail Marketing in India, What is E-Mail Marketing? E-Mail Marketing Strategy, Executing E-Mail Marketing, **Internet Marketing:**
Internet Marketing Strategy, Content Marketing, Content Marketing in India.

**Text Books:**
1. Programming the World Wide Web, Robet W Sebesta, Pearson, 8th edition, 2015
2. Java Script: The Definite Guide David Flanagan, O' Reilly Publisher
3. Internet Marketing- A Practical approach in the India Context by Moutusy Maity, Oxford

MCA V Semester
**Natural Language Processing**

**Course Objectives:**
This course introduces the fundamental concepts and techniques of natural language processing (NLP).

- Students will gain an in-depth understanding of the computational properties of natural languages and the commonly used algorithms for processing linguistic information.
- The course examines NLP models and algorithms using both the traditional symbolic and the more recent statistical approaches.
- Enable students to be capable to describe the application based on natural language processing and to show the points of syntactic, semantic and pragmatic processing.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Demonstrate a given text with basic Language features.
- Explain a rule based system to tackle morphology/syntax of a language.
- To design an innovative application using NLP components.
- To design a tag set to be used for statistical processing for real-time applications.
- To compare and contrast the use of different statistical approaches for different types of NLP applications.

**UNIT I:**
**Introduction** :Origins and challenges of NLP – Language Modelling: Grammar-based LM, Statistical LM – Regular Expressions, Finite-State Automata – English Morphology, Transducers for lexicon and rules, Tokenization, Detecting and Correcting Spelling Errors, Minimum Edit Distance.

**UNIT II:**
**Word Level Analysis**: Unsmoothed N-grams, Evaluating N-grams, Smoothing, Interpolation and Backoff – Word Classes, Part-of-Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models.

**UNIT III:**
**Syntactic Analysis:** Context-Free Grammars, Grammar rules for English, Treebank's, Normal Forms for grammar – Dependency Grammar – Syntactic Parsing, Ambiguity, Dynamic Programming parsing – Shallow parsing – Probabilistic CFG, Probabilistic CYK, Probabilistic Lexicalized CFGs – Feature structures, Unification of feature structures

**UNIT IV:**
**Semantics And Pragmatics:** Requirements for representation, First-Order Logic, Description Logics – Syntax-Driven Semantic analysis, Semantic attachments – Word Senses, Relations between Senses, Thematic Roles, selectional restrictions – Word Sense Disambiguation, WSD using Supervised, Dictionary & Thesaurus, Bootstrapping methods – Word Similarity using Thesaurus and Distributional methods

**UNIT V:**

**Discourse Analysis And Lexical Resources**: Discourse segmentation, Coherence – Reference Phenomena, Anaphora Resolution using Hobbs and Centering Algorithm – Co reference Resolution – Resources: Porter Stemmer, Lemmatizer, Penn Treebank, Brill's Tagger, WorldNet, PropBank, FrameNet, Brown Corpus, British National Corpus (BNC).

**Text Books:**
1. Daniel Jurafsky, James H. Martin―Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, Pearson Publication, 2014
2. Steven Bird, Ewan Klein and Edward Loper, ―Natural Language Processing with Python, First Edition, O'Reilly Media, 2009

**Reference Books:**
1. Breck Baldwin, ―Language processing with Java and Ling Pipe Cookbook, Atlantic Publisher, 2015
2. Richard M Reese, ―Natural Language Processing with Java, O'Reilly Media, 2015
3. Nitin Indurkhya and Fred J. Damerau, ―Handbook of Natural Language Processing, Second, Chapman and Hall/CRC Press, 2010
4. Tanveer Siddiqui, U.S. Tiwary, ―Natural Language Processing and Information Retrieval, Oxford University Press, 2008

MCA V Semester
**Internet of Things**

**Course Objectives:**
Students will be explored to the interconnection and integration of the physical world and the cyber space. They are also able to design & develop IOT Devices.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Familiarise the concepts of IoT and apply IoT to different applications
- Analyze and evaluate protocols used in IoT.
- Analyze and evaluate the data received through sensors in IoT
- understand and be able to explain the role of big data, cloud computing and data analytics in a typical IoT system
- Identify how IoT differs from traditional data collection systems
- Compare and Contrast the use of Devices, Gateways and Data Management in IoT.

**UNIT I:**
**The Internet of Things:** An Overview of Internet of things, Internet of Things Technology, behind Io Ts Sources of the Io Ts, M2M Communication, Examples of IoTs, Design Principles For Connected Devices Internet Connectivity Principles, Internet connectivity, Application Layer Protocols: HTTP, HTTPS, FTP, Telnet.

**UNIT II:**
Business Models for Business Processes in the Internet of Things ,IoT/M2M systems LAYERS AND designs standardizations ,Modified OSI Stack for the IoT/M2M Systems ,ETSI M2M domains and High-level capabilities ,Communication Technologies, Data Enrichment and Consolidation and Device Management Gateway Ease of designing and affordability

**UNIT III:**
Design Principles for the Web Connectivity for connected-Devices, Web Communication protocols for Connected Devices, Message Communication protocols for Connected Devices, Web Connectivity for connected-Devices.

**UNIT IV:**
Data Acquiring, Organizing and Analytics in IoT/M2M, Applications /Services /Business Processes, IOT/M2M Data Acquiring and Storage, Business Models for Business Processes in the Internet Of Things, Organizing Data, Transactions, Business Processes, Integration and Enterprise Systems.

**UNIT V:**
Data Collection, Storage and Computing Using a Cloud Platform for IoT/M2M Applications/Services, Data Collection, Storage and Computing Using cloud platform Everything as a service and Cloud Service Models, IOT cloud-based services using the Xively (Pachube/COSM), Nimbits and other platforms Sensor, Participatory Sensing, Actuator, Radio Frequency Identification, and Wireless, Sensor Network Technology, Sensors Technology ,Sensing the World.

**Text Books:**

1. Internet of Things: Architecture, Design Principles And Applications, Rajkamal,  McGraw Hill Higher Education
2. Internet of Things, A.Bahgya and V.Madisetti, Univesity Press, 2015

**Reference Books:**

1. Designing the Internet of Things, Adrian McEwen and Hakim Cassimally, Wiley
2. Getting Started with the Internet of Things CunoPfister , Oreilly

MCA V Semester
**DevOps**

**Course Objectives:**
DevOps improves collaboration and productivity by automating infrastructure and workflows and continuously measuring applications performance

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Understand the principles of continuous development and deployment, automation of configuration management, inter-team collaboration, and IT service agility
- Describe DevOps & DevSecOps methodologies and their key concepts
- Explain the types of version control systems, continuous integration tools, continuous monitoring tools, and cloud models
- Set up complete private infrastructure using version control systems and CI/CD tools

**UNIT I:**
Phases of Software Development life cycle. Values and principles of agile software development.

**UNIT II:**
Fundamentals of DevOps: Architecture, Deployments, Orchestration, Need, Instance of applications, DevOps delivery pipeline, DevOps eco system.

**UNIT III:**
DevOps adoption in projects: Technology aspects, Agiling capabilities, Tool stack implementation, People aspect, processes

**UNIT IV:**
CI/CD: Introduction to Continuous Integration, Continuous Delivery and Deployment , Benefits of CI/CD, Metrics to track CICD practices

**UNIT V:**
Devops Maturity Model: Key factors of DevOps maturity model, stages of Devops maturity model, DevOps maturity Assessment

**Text Books:**
1. The DevOPS Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations  by Gene Kim  , John Willis , Patrick Debois  , Jez Humb,O'Reilly publications
2. What is Devops?  Infrastructure as code By in Mike Loukides ,O'Reilly publications.
3. Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation, by Jez Humble and David Farley
4. Achieving DevOps: A Novel About Delivering the Best of Agile, DevOps, and Microservices by Dave Harrison, Knox Lively

**Reference Books:**
1. Building a DevOps Culture by Mandi Walls, O'Reilly publications
2. The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline With Containerized Microservices by Viktor Farcic

MCA V Semester
**Cryptography and Network Security Lab**

**Course Objectives:**
- To learn basic understanding of cryptography, how it has evolved, and some key encryption techniques used today.
- To understand and implement encryption and decryption using Ceaser Cipher, Substitution Cipher, Hill Cipher.

**Course Outcomes**: At the end of the course, student will be able to
- Apply the knowledge of symmetric cryptography to implement encryption and decryption using Ceaser Cipher, Substitution Cipher, Hill Cipher
- Demonstrate the different algorithms like DES, BlowFish, and Rijndael, encrypt the text "Hello world" using Blowfish Algorithm.
- Analyze and implement public key algorithms like RSA, Diffie-Hellman Key Exchange mechanism, the message digest of a text using the SHA-1 algorithm

**List of Experiments:**
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 and XOR each character in this string with 127 and display the result
3. Write a Java program to perform encryption and decryption using the following algorithms:
    a) Ceaser Cipher
    b) Substitution Cipher
    c) Hill Cipher
4. Write a Java program to implement the DES algorithm logic
5. Write a C/JAVA program to implement the BlowFish algorithm logic
6. Write a C/JAVA program to implement the Rijndael algorithm logic.
7. Using Java Cryptography, encrypt the text "Hello world" using BlowFish. Create your own key using Java key tool.
8. Write a Java program to implement RSA Algoithm
9. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).
10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.

MCA V Semester
**Big Data Analytics Lab**

**Course Objectives:**
The objectives of this course are to
- Introduce the tools required to manage and analyze big data like Hadoop.
- Impart knowledge of Map reduce paradigm to solve complex problems Map-Reduce

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Demonstrate the setup of Big Data in distributed environment
- Efficiently Work with Big Data Platform.
- Have Hands-on experience on developing applications using MapReduce
- Identify the need of interfaces to perform I/O operations in Hadoop and analyze big data
- Demonstrate the understanding of storing and managing Big Data using HDFS, Pig and Hive tools
- Design and implement algorithms for Big Data Mining for various real time applications

**Experiment 1:**
Implement the following Data structures in Java
a)Linked Lists b) Stacks c) Queues d) Set e) Map

**Experiment 2:**
(i)Perform setting  up and Installing Hadoop in its three operating modes:
   Standalone, Pseudo distributed and Fully distributed
(ii)Use web based tools to monitor your Hadoop setup.

**Experiment 3:**
Implement the following file management tasks in Hadoop:
- Adding files and directories
- Retrieving files
- Deleting files

 Hint: A typical Hadoop workflow creates data files (such as log files) elsewhere and copies them
 into HDFS using one of the above command line utilities.

**Experiment 4:**
Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.

**Experiment 5:**
Write a Map Reduce program that mines weather data.
Weather sensors collecting data every hour at many locations across the
globe gather a large volume of log data, which is a good candidate for analysis with MapReduce,
since  it is semi structured and record oriented.

**Experiment 6:**
Implement Matrix Multiplication with Hadoop Map Reduce

**Experiment 7:**
Install and Run Pig then write Pig Latin scripts to sort, group, join, project, and filter your data.

**Experiment 8:**
Install and Run Hive then  use Hive to create, alter, and drop databases, tables, views, functions, and indexes

**Text Books:**
1. Big Java 4th Edition, Cay Horstmann, Wiley John Wiley & Sons, INC
2. Hadoop: The Definitive Guide by Tom White, 3$^{rd}$ Edition, O'reilly
3. Hadoop in Action by Chuck Lam, MANNING Publ
4. Hadoop for Dummies by Dirk deRoos, Paul C.Zikopoulos, Roman B.Melnyk, Bruce Brown, Rafael Coss

**Reference Books:**
1. Hadoop in Practice by Alex Holmes, MANNING Publ
2. Hadoop MapReduce Cookbook, Srinath Perera, Thilina Gunarathne

MCA V Semester
**Machine Learning with Python Lab**

**Course Objectives:**
 This course will enable students to
- To learn and understand different Data sets in implementing the machine learning algorithms
- Implement the machine learning concepts and algorithms in any suitable language of choice.

**Course Outcomes(COs)**: At the end of the course, student will be able to
- Implement procedures for the machine learning algorithms
- Design Python programs for various Learning algorithms
- Apply appropriate data sets to the Machine Learning algorithms
- Identify and apply Machine Learning algorithms to solve real world  problems

**List of Experiments:**
1. Hypothesis based on thyroid dataset.
   Implement and demonstrate the FIND-S algorithm for finding the most specific
2. For a given diabetic dataset, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.
3. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
4. Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.
5. Write a program to implement the naïve Bayesian classifier for Iris data set. Compute the accuracy of the classifier, considering few test data sets.
6. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.
7. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library classes/API.
8. Apply EM algorithm to cluster a Heart Disease Data Set. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.
9. Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions.
10. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.