CURRICULUM STRUCTURE

and

DETAILED SYLLABI

for

Two Year PG Programme

MCA
(Master of Computer Applications)

(Applicable for batches admitted from 2020)

UNIVERSITY COLLEGE OF ENGINEERING VIZIANAGARAM
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
VIZIANAGARAM– 533003, ANDHRA PRADESH, INDIA

Department of Information Technology
R20- Master of Computer Applications Course Structure & Syllabi

MCA Course Structure
I Semester

S.No	Course Code	Course Name	Category	L	T	P	Credits
1	MCA1101	Business Communication	BS&H	2	0	0	2
2	MCA1102	Statistics with R programming	PC	1	0	4	3
3	MCA1103	Computer Organization	PC	3	0	0	3
4	MCA1104	Operating Systems	PC	3	0	0	3
5	MCA1105	Data Structures	PC	3	0	0	3
6	MCA1106	Software Engineering	PC	0	0	3	3
7	MCA1107	Data Structures Lab	PC	0	0	3	1.5
8	MCA1108	Software Engineering Lab	PC	0	0	3	1.5
9	MCA1109	Socially Relevant Project using Design Thinking	MC	0	0	1	0.5
			Total	12	0	14	20.5

MCA Course Structure
II Semester

S.No	Course Code	Course Name	Category	L	T	P	Credits
1	MCA2101	Database Management Systems	PC	3	0	0	3
2	MCA2102	Computer Networks	PC	3	0	0	3
3	MCA2103	Object Oriented Programming with JAVA	PC	3	0	0	3
4	MCA2104	Design and Analysis of Algorithms	PC	3	0	0	3
5	MCA2105	**Elective-I** • NoSQL Databases • Mobile Application Development • Artificial Intelligence • Image Processing • Accounting for Managers	PE	3	0	0	3
6	MCA2106	DBMS Lab	PC	0	0	3	1.5
7	MCA2107	Computer Networks Lab	PC	0	0	3	1.5
8	MCA2108	JAVA Programming Lab	PC	0	0	3	1.5
9	MCA2109	Employability Skills	MC	0	0	1	0.5
10	MCA2110	Bridge Course*(Python Programming)	MC	0	0	0	
			Total	15	0	10	20

MCA Course Structure
III Semester

S.No	Course Code	Course Name	Category	L	T	P	Credits
1	MCA3101	Machine Learning with Python	PC	3	0	0	3
2	MCA3102	Internet of Things	PC	3	0	0	3
3	MCA3103	Web Technologies	PC	2	0	2	3
4	MCA3104	Data Warehousing and Mining	PC	3	0	0	3
5	MCA3105	Cryptography and Network Security					3
6	MCA3106	**Elective-II** • Soft Computing • Software Project Management • Cloud Computing • Optimization Techniques • Ad-hoc and Sensor Networks	PE	3	0	0	3
7	MCA3107	Machine Learning with Python Lab	PC	0	0	4	2
8	MCA3108	IoT Lab	PC	0	0	3	1.5
9	MCA3109	Internship / Industry Oriented Mini Project/ Skill Development Course * (Minimum 6-weeks)	PR	0	0	1	2
			Total	14	0	10	23.5

MCA Course Structure
IV Semester

S.No	Course Code	Course Name	Category	L	T	P	Credits
1	MCA4101	Elective-III * • MOOCs-1 (NPTEL/SWAYAM) Any recommended courses like • Digital Marketing • Human Resource Management • Deep Learning • Cyber Security& Forensic - Full Stack Technologies -	PE	2	0	0	2
2	MCA4102	• Elective-IV * MOOCs-2 (NPTEL/SWAYAM) -Any recommended course • Network Programming • Block Chain technologies • Software Testing Methodologies • Big Data Analytics -Data Science	PE	2	0	0	2
3	MCA4103	Project Work/ Dissertation	PR	0	0	0	12
			Total	4	0	0	16

***Students going for Industrial Project/Thesis will complete these courses through MOOCs (even in earlier semester)**

L	T	P
2	0	0

MCA I Semester

Business Communication

Code:MCA1101

Course Objectives:
To acquaint the students with fundamentals of communication, help them honing oral, written and non-verbal communication skills and to transform them as effective communicators.

UNIT I:
Purpose and process of communication: Objectives of Communication-Process of Communication-Types of communication; noise, listening skills, Types of listening, essentials of good listening and tips.

UNIT II:
Managing Organizational Communication: Formal and Informal Communication- Interpersonal and Intrapersonal communication- Role of Emotion in Interpersonal Communication- Barriers to Interpersonal Communication- Exchange Theory-Gateways for Effective Interpersonal Communication.

UNIT III:
Non-verbal communication and Body Language: Kinesics, Proxemics, Paralanguage, Haptics, handshakes, appropriate body language and mannerisms for interviews: business etiquettes-across different cultures.

UNIT IV:
Written communication: mechanics of writing, report writing- business correspondence-business letter format- Meetings and managing meetings- Resume writing-Formats and Skills.

UNIT V:
Presentation skills: prerequisites of effective presentation, format of presentation; Assertiveness – strategies of assertive behavior; Communication skills for group discussion and interviews, Interview Techniques.

Note: Relevant cases have to be discussed in each unit and in examination case is compulsory from any unit.

Outcomes:

By the end of the course students should be able to:

1. Apply business communication theory to solve workplace communication issues.
2. Demonstrate the communication skills required in the workplace.
3. Understand complex ideas in written and spoken formats.
4. Express complex ideas accurately in written and spoken formats.
5. Manage resources effectively and efficiently in an academic context.
6. Obtain information from a variety of sources and use it ethically.

Text Books:

1) Mallika Nawal: "Business Communication", Cengage Learning, New Delhi, 2012.
2) Edwin A. Gerloff, Jerry C. Wofford, Robert Cummins Organisational Communication: The key stone tomanagerialeffectiveness.
3) Meenakshi Rama: "*Business Communication*", Oxford University Press, NewDelhi
4) C.S.G. Krishnamacharyulu and Dr. Lalitha Ramakrishnan, Business Communication, Himalaya Publishing House,Mumbai

Reference Books:

1) Paul Turner: "*Organisational Communication*", JAICO Publishing House, New Delhi.
2) Sathya Swaroop Debasish, Bhagaban Das" "*Business Communication*", PHI Private Limited, New Delhi, 2009.
3) R.K.Madhukar:"BusinessCommunication",VikasPublishingHouse,New Delhi,2012.
4) Kelly M Quintanilla, Shawn T.Wahl:"Business and Professional Communication", SAGE, New Delhi,2012.
5) Sangita Mehta, NeetyKaushish: "Business Communication", University Science Press, New Delhi,2010.
6) Anjali Ghanekar: "Business Communication Skills", Everest Publishing House, NewDelhi,2011

MCA I Semester

L	T	P
1	0	4

STATISTICS WITH R PROGRAMMING

Code: MCA1102

Learning objectives:

Should be able to

1. This course includes the installation process of R Language
2. This course is intended to teach the basics of R , control statements, Create loops for iteration recursion
3. This includes math functions, linear algebra operations
4. This includes graphs and the plot()function ,
5. This includes probability distributions different linear and non linear models

UNIT-I:

Introduction, How to run R, R Sessions and Functions, Basic Math, Variables, Data Types, Vectors, Conclusion, Advanced Data Structures, Data Frames, Lists, Matrices, Arrays, Classes.

UNIT-II:

R Programming Structures, Control Statements, Loops, - Looping Over NonvectorSets,- If-Else, Arithmetic and Boolean Operators and values, Default Values for Argument, Return Values, Deciding Whether to explicitly call return- Returning Complex Objects, Functions are Objective, No Pointers in R, Recursion, A Quick sort Implementation- Extended Extended Example: A Binary Search Tree.

UNIT-III:

Doing Math and Simulation in R, Math Function, Extended Example Calculating Probability- Cumulative Sums and Products-Minima and Maxima- Calculus, Functions Fir Statistical Distribution, Sorting, Linear Algebra Operation on Vectors and Matrices, Extended Example: Vector cross Product- Extended Example: Finding Stationary Distribution of Markov Chains, Set Operation, Input /output, Accessing the Keyboard and Monitor, Reading and writer Files,

UNIT-IV:

Graphics, Creating Graphs, The Workhorse of R Base Graphics, the plot () Function – Customizing Graphs, Saving Graphs to Files.

UNIT-V:

Probability Distributions, Normal Distribution- Binomial Distribution- Poisson Distributions Other Distribution, Basic Statistics, Correlation and Covariance, T-Tests,- ANOVA. Linear Models, Simple Linear Regression, -Multiple

Regression Generalized Linear Models, Logistic Regression, - Poisson Regression- other Generalized Linear Models-Survival Analysis, Nonlinear Models, Spines- Decision- Random Forests,

Outcomes:

At the end of this course, each student will be ableto:

- Understand basic concepts such as data type and index and use them in their work
- Demonstrate use of basic functions Conceptualize and create loops to solve different types of problems
- Create their own customized functions
- Construct tables, plots and figures for descriptive statistics
- Learn to understand different probability distributions

TEXT BOOKS:

1. The Art of R Programming, Norman Matloff, Cengage Learning,Oct-2011
2. R for Everyone, Lander, Pearson,2/e

REFERENCE BOOKS:

1. R Cookbook, PaulTeetor, Oreilly,2/e
2. R in Action,RobKabacoff, Manning,2/e

L	T	P
3	0	0

MCA I Semester

Computer Organization

Code: MCA1103

Objective:

1. This course is intended to teach the Basic Structure Of Computers
2. This includes the concepts Addressing Modes and Component of Instructions
3. This course will also expose students to Arithmetic and Logic Instructions
4. This includes the concepts Input/output Organization
5. This includes the concepts Memory Systems

UNIT -I:

Basic Structure Of Computers: Functional unit, Basic Operational concepts, Bus structures, System Software, Performance, The history of computer development.

UNIT -II:

Machine Instruction and Programs: Instruction and Instruction Sequencing: Register Transfer Notation, Assembly Language Notation, Basic Instruction Types, Basic Input/output Operations, The 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: Increment,decrement,add,subtract,multiply,divide,add with carry, subtract withborrow,negate(2"scompliment),AND,OR,XOR,**Branch Instructions Addressing Modes:**Implied,Immediate,Register,Register Indirect,Auto increment and Auto decrement,Direct address,Indirect Address, RelativeAddress,Indexed ,Base Register addressingmodes.

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:

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, **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

Learning Outcomes:

The student will be able to:

1. Understand the basic operational concepts the history of computer development
2. Understand and identify the role of stacks and queues in computer programming equation.
3. Understand the different addressing modes.
4. Understand the interrupts and direct memory access and different buses
5. Understand the concept of micro programmed control

TEXT BOOKS:

1. Computer Organization, Carl Hamacher, ZvonksVranesic, SafeaZaky, 5th Edition, McGraw Hill.
2. Computer Architecture and Organization, John P. Hayes ,3rd Edition, McGraw Hill.
3. Computer System Architecture, M. Morris Mano,Rajib Mall,3/e

REFERENCE BOOKS:

1. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI
2. Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition PHI/Pearson
3. Fundamentals or Computer Organization and Design, - SivaraamaDandamudi Springer Int. Edition.

L	T	P
3	0	0

MCA I Semester

OPERATING SYSTEMS

Code: MCA1104

Objective:

Students will try to learn:
To understand the main components of an OS & their Students will able to:

- Describe the important computer system resources
- To study the process management and scheduling.
- To understand various issues in Inter Process Communication (IPC) and the role of OS in IPC.
- To understand the concepts and implementation Memory management policies and virtual memory. 5. To understand the working of an OS as a resource manager, file system manager, process manager, memory manager and I/O manager and methods used to implement the different parts of OS

UNIT-I:

Introduction: Computer –system organization, Computer- system Architecture, Operating-system Structure, Operating-system Operations, Process Management, Memory Management, Storage Management, Protection and Security, Distributed Systems, Special-purpose systems,Computing Environments , Operating-system Structure:, Operating-system Services, User , Operating-system Interface, System calls, System programs, Operating-system Design and Implementation, , Operating-system structure, Virtual Machine

UNIT-II:

Process Management: Processes: Process Concept, Process Scheduling, Operations on Processes, InterprocessCommunication, Examples of IPC Systems, Communication in Client-Server systems

Threads: Overview, Multithreading Models, Thread Libraries, Java Threads, Threading Issues, OS Examples
CPU Scheduling: Basic concepts, Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Thread Scheduling, Operating system Examples
Process Synchronization: Background, The Critical- section problem, Petersons solution, Synchronization Hardware, Semaphores, Classic problems of Synchronization, Monitors, AtomicTransactions.

UNIT-III:

Memory management: Main memory: Swapping, Contiguous memory Allocation, Paging, Structure of the Page table, Segementation
Virtual memory: Background, Demand paging, copy- on-Write, Page Replacement, Allocation of frames, Thrashing, Memory-Mapped Files.

UNIT-IV:

File-system Interface: Concept, Access Methods, Directory structure, Filesystem Mounting, File sharing, Protection

File-system Implementation: File-system Structure, Implementation, Directory Implementation, Allocation Methods, Free- Space Management, Efficiency and Performance, Recovery, Log-Structured File systems, NFS Mass –storage Structure: Overview, Disk Structure, Disk Attachment, Disk Scheduling, Disk and swap-space Management, RAID Structure, Stable- Storage Implementation, Tertiary-Storage Structure I/O systems: Overview, I/O Hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O requests to Hardware Operations, STREAMS, Performance.

UNIT-V:

Deadlocks: System model, Deadlock Characterization, Methods for handling Deadlocks, Deadlock Prevention, Deadlock avoidance Deadlock Detection and Recovery form Deadlock.
Protection: Goals of Protection, Principles of protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability – Based systems,Language-Based Protection
Security: The Security Problem, Program Threads, System and Network Threats, Cryptography as a security tool, User Authentication, Implementing security Defenses, Firewalling to protect systems and Networks.

Course outcomes

- Describe the important computer system resources and the role of operating system in their management policies and algorithms.
- Understand the process management policies and scheduling ofprocesses by CPU
- Evaluate the requirement for process synchronization and coordination handled by operating system
- Describe and analyze the memory management and its allocation policies.
- Identify use and evaluate the storage management policies with respect to different storage management technologies.

TEXTBOOKS:

1. Operating system concepts, 7/e, Abraham Siliberschatz, Galvin, John Wiley &sons , Inc.
2. Operating systems, 6/E, William stallings, PHI/Pearson.
3. Operating systems, 2/e, Dhamdhere, TMH.

REFERENCES:

1. Operating systems 3/e, Dietal, Dietal, Pearson.
2. An introduction to Operating systems, Concepts and practice, Pramod Chandra P. Bhat, PHI,5/e
3. Operating systems, Elmasri, Carrick, Levine, TMH,3/e
4. Operating systems, 3/e ,Nutt, Chaki, Neogy Pearson.
5. Operating systems, Brian L. Stuart, Cengage.2009
6. Operating systems, Haldar, Aravind, Pearson,2/e.
7. Operating systems, PAL Choudhury, PHI.3/e
8. Operating systems: design and Implementation, 3/e, Tanenbaum, Woodhull.

L	T	P
3	0	0

MCA I Semester

DATA STRUCTURES

Code: MCA1105

Course Objectives:

The objective of this course is to explore basic data structures such as stacks and queues, introduce a variety of data structures such as hash tables, search trees, tries, heaps, graphs, sorting and pattern matching algorithms

UNITI:

Introduction to C: Constants and variables, Operators and Expressions, Managing Input and Output operators, Decision making-branching and looping, Arrays.

UNIT II:

Functions, Structures and Unions, Pointers, File handling in C.

UNIT III:

Data structure: Definition, types of data structures Recursion Definition, Design Methodology and Implementation of recursive algorithms, Linear and binary recursion. Preliminaries of algorithms, analysis and complexity .

Linear list – singly linked list, Double linked list and circular linked list - implementation, insertion, deletion and searching operations on linear list.

UNIT IV:

Stacks-Operations, array and linked representations of stacks, stack applications, **Queues**-operations, array and linkedrepresentations.

Hash Table Representation: hash functions, collision resolution-separate chaining, open addressing-linear probing, quadratic probing, double hashing and rehashing, extendible hashing.

UNIT V:

Sorting Techniques: Insertion sort, selection sort, exchange-bubble sort, quick sort and merge sort Algorithms. **Trees:** Binary Trees, terminology, representation and traversals- pre, post & in order traversals. **Search Trees:** Binary Search Trees, Definition, Implementation, Operations-Searching, Insertion and Deletion

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

- Implement basic programs by using C concepts.
- Select the data structures that efficiently model the information in a problem
- Assess efficiency trade-offs among different data structure implementations or combinations
- Implement and know the application of algorithms for sorting and pattern matching.

Text Books:

1) Let Us C: Authentic Guide to C Programming Language, 17th ed., YashavantKanetkar, BPB Publications.
2) Data Structures Using C. 2nd Edition, Reema Thareja, Oxford
3) Data Structures and Algorithm Analysis in C, 2nded, Mark Allen Weiss

Reference Books:

1) Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilbergand B.A. Forouzan, Cengage Learning.
2) Programming in ANSI C, 5thed, E. Balaguruswamy,TMH

L	T	P
3	0	0

MCA I Semester

Software Engineering

Code: MCA2106

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.

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.

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.

TEXTBOOKS:

1. A Concise introduction to software engineering (undergraduate topics incomputer science), Pankaj Jalote, Springer InternationalEdition.
2. Software Engineering, APrecise approach, Pankaj Jalote,Wiley,2010
3. Software Engineering, 3/e, & 7e Roger S.Pressman,TMH
4. Fundamentals of software Engineering,RajibMall,PHI,2/e

REFERENCEBOOKS:

1. Software Engineering, 8/e, Sommerville, Pearson.
2. Software Engineering principles and practice, W S Jawadekar,TMH,7/e
3. Software Engineering concepts, RFairley,TMH,1997

L	T	P
0	0	3

MCA I Semester

Data Structures Lab

Code: MCA1107

Course Objectives: This Course will enable students to

- Design and implement various data structures.
- Implement operations like searching, insertion, and deletion, traversing
- mechanism
- Develop applications using data structure algorithms.

Experiment 1:

a) Write a program in C to display the n terms of even natural numberand their sum.
b) Write a program in C to display the n terms of harmonic series and their sum. 1 + 1/2 + 1/3 + 1/4 + 1/5 ... 1/n terms.
c) Write a C program to check whether a given number is an Armstrong number or not.
d) Write a C program to calculate the factorial of a given number.

Experiment 2:

a) Write a program in C for multiplication of two square Matrices.
b) Write a program in C to find transpose of a given matrix.

Experiment 3:

a) Write a program in C to check whether a number is a prime number or not using the function.
b) Write recursive program which computes the n^{th} Fibonacci number, for appropriate values of n.
c) Write a program in C to add numbers using call by reference.

Experiment 4:

a) Write a program in C to append multiple lines at the end of a text file.
b) Write a program in C to copy a file in another name.

Experiment 5:

Write recursive program for the following

a) Write recursive and non recursiveC program for calculation of Factorial of an integer.
b) Write recursive and non recursiveC program for calculation of GCD (n, m)
c) Write recursive and non recursiveC 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 6:

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 7:

a) Write C program that implement stack (its operations) using arrays.

b) Write C program that implement stack (its operations) using Linked list.

Experiment 8:

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 9:

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

Experiment 10:

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

Experiment 11:

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 12:

a) Write a C program to implement Prims" algorithm.

b) Write a C program to implement Kruskal"salgorithm.

Experiment 13:

Implementation of Hash table using double hashing as collision resolution function.

Experiment 14:

Implementation of Binary Search trees- Insertion and deletion.

Experiment 15:

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.

c) Write C program that implement merge sort, to sort a given list of integers in ascending order

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

- Implement various basic data structures and itsoperations.
- Apply sorting and searching algorithms to given numbers
- Implement various tree operations.
- Implement various graphs algorithms.
- Develop applications using various data structures.

Text Books:

1. Let Us C: Authentic Guide to C Programming Language, 17th ed., YashavantKanetkar, BPB Publications.
2. Data Structures Using C. 2nd Edition, Reema Thareja, Oxford
3. Data Structures and Algorithm Analysis in C, 2nded, Mark Allen Weiss

Reference Books:

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilbergand B.A. Forouzan, Cengage Learning.
2. Programming in ANSI C, 5thed, E. Balaguruswamy,TMH

L	T	P
0	0	3

MCA I Semester

Software Engineering Lab

Code: MCA1108

Course Objectives:

- To understand SRS document, design document, test cases and software configuration management and risk
- management related document To understand methods of capturing, specifying, visualizing and analyzing software requirements.

1) Take any real time problem and do the following experiments
   a. Write down the problem statement for a suggested case studies.
   b. Do requirement analysis and develop Software Requirement Specification Sheet (SRS) for suggested system.
   c. Using COCOMO model estimate effort.
   d. Perform Estimation of effort using FP Estimation for chosen system
   e. Analyze the Risk related to the project and prepare RMMM plan.
   f. Develop Time-line chart and project table using PERT or CPM project scheduling methods.
   g. Draw E-R diagrams, DFD, CFD and structured charts for the project.
   h. Design of Test cases based on requirements and design.
   i. Prepare FTR
   j. Prepare Version control and change control for software configuration items.

The suggested case studies are:

- iLearn: A digital learning environment
- Mentcare: A mental health support system
- A personal insulin pump
- A wilderness weather station
- Airbus 340 flight control system

Outcomes:

- Able to prepare SRS document, design document, test cases and software configuration management and risk management related document.
- Able to draw the E-R diagrams, DFD, CFD and structured charts for the project

TEXTBOOKS:

1. A Concise introduction to software engineering (undergraduate topics incomputer science), Pankaj Jalote, SpringerInternationalEdition.
2. Software Engineering, APrecise approach, Pankaj Jalote,Wiley,2010
3. Software Engineering, 3/e, & 7e Roger S.Pressman,TMH
4. Fundamentals of software Engineering,Rajib Mall,PHI,2/e

REFERENCEBOOKS:

1. Software Engineering, 8/e, Sommerville, Pearson.
2. Software Engineering principles and practice, W S Jawadekar,TMH,7/e
3. Software Engineering concepts, R Fairley,TMH,1997

MCA I Semester

L	T	P
0	0	1

Socially Relevant Project using Design Thinking

Code: MCA1109

Course Objectives:

- Build mindsets & foundations essential for designers
- Learn about the Human-Centered Design methodology and understand their real-world applications
- Use Design Thinking for problem solving methodology for investigating illdefined problems.
- Undergo several design challenges and work towards the final design challenge

Apply Design Thinking on the following Streams to

- Project Stream 1: Electronics, Robotics, IOT and Sensors
- Project Stream 2: Computer Science and IT Applications
- Project Stream 3: Mechanical and Electrical tools
- Project Stream4: Eco-friendly solutions for waste management, infrastructure, safety, alternative energy sources, Agriculture, Environmental science and other fields of engineering.

How to Pursue The Project Work?

- The first part will be learning-based-masking students to embrace the methodology by exploring all the phases of design thinking through the wallet/ bag challenge and podcasts.
- The second part will be more discussion-based and will focus on building some necessary skills as designers and learning about complementary material for human- centered design.
- The class will then divide into teams and they will be working with one another for about 2 – 3 weeks. These teams and design challenges will be the basis for the final project and final presentation to be presented.
- The teams start with **Design Challenge** and go through all the phases more in depth from coming up with the right question to empathizing to ideating to prototyping and to testing.
- Outside of class, students will also be gathering the requirements, identifying the challenges, usability, importance etc
- At the end, Students are required to submit the final reports, and will be evaluated by the faculty.

Tasks to be done:

Task 1: Everyone is a Designer

- Understand class objectives & harness the designer mindset Task 2: The

Wallet/Bag Challenge and Podcast

- Gain a quick introduction to the design thinking methodology

- Go through all stages of the methodology through a simple design challenge
- Podcast: Observe, Listen and Engage with the surrounding environment and identify a design challenge.

Task 3: Teams & Problems

- Start Design Challenge and learn about teams & problems through this
- Foster team collaboration, find inspiration from the environment and learn how to identify problems

Task 4: Empathizing

- Continue Design Challenge and learn empathy
- Learn techniques on how to empathize with users
- Go to the field and interview people in their environments
- Submit Activity Card

Task 5: Ideating

- Continue Design Challenge and learn how to brainstorm effectively
- Encourage exploration and foster spaces for brainstorming
- Submit Activity Card

Task 6: Prototyping

- Continue Design Challenge and learn how to create effective prototypes
- Build tangible models and use them as communication tools
- Start giving constructive feedback to classmates and teammates
- Submit Activity Card

Task 7: Testing

- Finish Design Challenge and iterate prototypes and ideas through user feedback
- Evolve ideas and prototypes through user feedback and constructive criticism
- Get peer feedback on individual and group performance
- Submit Activity Card

Task 8:

- Final Report Submission and Presentation

Note: The colleges may arrange for Guest Speakers from Various Design Fields: Graphic Design, Industrial Design, Architecture, Product Design, Organizational Design, etc to enrich the students with Design Thinking Concept.

Textbooks:

1. Tom Kelly, *The Art of Innovation: Lessons in Creativity From IDEO, America's Leading Design Firm* (Profile Books, 2002)

References:

1. Tim Brown, Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation (HarperBusiness,2009)
2. Jeanne Liedtka, Randy Salzman, and Daisy Azer, Design Thinking for the Greater Good: Innovation in the Social Sector (Columbia Business School Publishing, 2017)

Other Useful Design Thinking Frameworks and Methodologies:

- Human-Centered Design Toolkit (IDEO); https://www.ideo.com/post/design-kit
- Design Thinking Boot Camp Bootleg (Stanford D-School); https://dschool.stanford.edu/resources/the-bootcamp-bootleg
- Collective Action Toolkit (frogdesign); https://www.frogdesign.com/wpcontent/uploads/2016/03/CAT_2.0_Eng lish.pdf
- Design Thinking for Educators (IDEO); https://designthinkingforeducators.com/

MCA II Semester

L	T	P
3	0	0

Database Management Systems

Code: MCA2101

Course Objectives:

This Course will enable students to

- Explain the concept of databases, database management systems, database structures and how they work.
- Make use of Entity-Relationship Modeling and Relational Modeling for creating simple databases from the real world scenarios.
- Write relational algebra and structured query language (SQL) statements.
- Normalize a database using Normalization Rules.
- Discuss the issues associated with Transaction Management and Recovery, Tree Structured and Hash-Based Indexing

UNITI:

Introduction to Databases: Introduction, An Example, Characteristics of the Database Approach, Actors on Scene, Workers behind the scene, Advantages of Using the DBMS Approach, A Brief History of Database Applications, When Not to Use a DBMS **[Text book-3]**

Overview of Database Languages and Architectures: Data Models, Schemas and Instances, Three-Schema Architecture and Data Independence, Database Languages and Interfaces, The Database System Environment, Centralized and Client/Server Architecture for DBMSs, Classification of Database Management Systems **[Text book-3]**

UNIT II:

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design with the ER Model, Conceptual Design for Large Enterprises

Relational Model: Introduction to the Relational Model, Integrity Constraints over Relations, Enforcing Integrity Constraints, Querying Relational Data, Logical Database Design: ER to Relational, Introduction to Views, Destroying/Altering Tables and Views

UNIT III:

Relational Algebra: Selection and Projection, Set Operations, Renaming, Joins, Division, More Examples of Algebra Queries.

SQL: Queries, Constraints, Triggers: The Form of a Basic SQL Query, UNION, INTERSECT and EXCEPT, Nested Queries, Aggregate Operators, Null Values, Complex Integrity Constraints in SQL, Triggers and Active Databases, Designing Active Databases.

UNIT IV:

Introduction to Normalization Using Functional and Multivalued Dependencies: Informal Design Guidelines for Relation Schema, Functional Dependencies, Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.

UNIT V:

Transaction Management and Concurrency Control: Transaction Concept, A Simple Transaction Model, Storage Structure, ACID Properties, Serializability, Transaction Isolation Levels, Concurrency Control, Lock-Based Protocols, Validation-Based Protocols **[Text Book-2]**

Note: For Practical Examples Please Go Through Reference 1

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

- Illustrate the concept of databases, database management systems, database languages, database structures and their work
- Apply ER modeling and Relational modeling for designing simple databases.
- Summarize the concepts related to relational model and SQL and Write database queries using relational algebra and structured query language.
- Design and develop databases from the real world by applying the concepts of Normalization.
- Outline the issues associated with Transaction Management and Recovery, Tree Structured and Hash-Based Indexing

Text Books:

1) Database Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke, Mc Graw-Hill
2) Database System Concepts, 6/e, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Mc Graw-Hill
3) Database Systems, 6/e RamezElmasri, Shamkant B. Navathe, Pearson

Reference Books:

1) Database Systems, 9/e, Carlos Coronel, Steven Morris, Peter Rob, Cengage
2) Introduction to Database Systems, 8/e, C J Date, Pearson

MCA II Semester

L	T	P
3	0	0

Computer Networks

Code: MCA2102

Course Objectives:

At the end of the course, the students will be ableto:

- Understands the fundamental concepts of computer networking and OSIReference model.
- Familiarize the student with the basic taxonomy and terminology of thecomputer networking area.
- Learn and understand the advanced networking concepts, preparing thestudent for entry advanced courses in computer networking.
- Develop and gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

UNIT I:

Introduction: Network Topologies WAN, LAN, MAN. Reference models- The OSI Reference Model- the TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models.**Physical Layer -**Introduction to physical layer-Data and Signals, Periodic analog signals, digital signals, transmission impairment,

,Data rate limits, performance -Introduction to Guided Media- Twisted-pair cable, Coaxial cable and Fiber optic cable and Unguided media: Wireless-Radio waves, microwaves, infrared.

UNIT II:

The Data Link Layer - Services Provided to the Network Layer – Framing – ErrorControl–Flow Control, Error Detection and Correction – Error-Correcting Codes – Error Detecting Codes. **Elementary Data Link Protocols-** A Utopian Simplex Protocol-A Simplex Stop and Wait Protocol for an Error free channel-A Simplex Stop and Wait Protocol for a Noisy Channel, Sliding Window Protocols- A One Bit Sliding Window Protocol-A Protocol Using Go-Back-N- A Protocol Using Selective Repeat.

UNIT III:

The Medium Access Control Sub layer-The Channel Allocation Problem-Static Channel Allocation-Assumptions for Dynamic Channel Allocation, Multiple Access Protocols-Aloha-Pure aloha- slotted aloha-Carrier Sense Multiple Access Protocols- Collision-Free Protocols-Limited Contention Protocols. **Wireless LAN Protocols-** Ethernet-Classic Ethernet Physical Layer-Classic Ethernet MAC Sub-layer Protocol-Ethernet Performance-Fast Ethernet- Wireless LANs-The 802.11 Architecture and Protocol Stack-The 802.11 Physical Layer-The802.11 MAC Sub-layer Protocol- The 805.11 Frame Structure-Services.

UNIT IV:

The Network Layer Design Issues - Store and Forward Packet Switching- Services Provided to the Transport layer- Implementation of Connectionless Service-Implementation of Connection Oriented Service- Comparison of Virtual Circuit and Datagram Networks, Routing Algorithms-The Optimality principle- Shortest path, Flooding, Distance vector, Link state, Hierarchical. **Congestion Control algorithms-**General principles of congestion control, Congestion prevention polices, Approaches to Congestion Control-Traffic Aware Routing- Admission Control-Traffic Throttling-Load Shedding. **Internet Working:** How networks differ- How networks can be connected- Tunneling, internetwork routing-, Fragmentation, network layer in the internet – IP protocols-IP Version

4 protocol-, IP addresses-, Subnets-IP Version 6-The main IPV6 header- Internet control protocols- ICMP-ARP-DHCP.

UNITV:

The Transport Layer: Transport layer protocols: Introduction-services- port number-User data gram protocol-User datagram-UDP services-UDP applications-Transmission control protocol: TCP services- TCP features- Segment- A TCP connection- windows in TCP- flow control-Error control. **Application Layer --** World Wide Web: HTTP , FTP-Two connections-control connection-Data connection-security of FTP-Electronic mail-Architecture- web based mail- email security- TELENET-local versus remote Logging. **Domain Name System:** Name Space, DNS in Internet, - Resolution-Caching- Resource Records- DNS messages- Registrars-security of DNS Name Servers.

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

- Explain the network architecture, TCP/IP and OSI reference models
- Identify and understand various techniques and modes of transmission
- Demonstrate the 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
- Develop network security and define various protocols such as FTP, HTTP, Telnet, DNS

Text Books:

1) Computer Networks: Andrew S Tanenbaum David J. Wetherall, 5/e, Pearson
2) Data communications and networking: Behrouz Forouzan, 5/e, McGraw Hill
3) Data and computer Communications, William Stallings, PEARSON, 10/e

Reference Books:

1) Computer Networks – A System Approach, Peterson, Bruce Davie, 2/e , Harcourt Asia
2) Compute communications and networking technologies, Gallo, Hancock, Cengage
3) An Engineering approach to compute networking, Kesha, Pearson

MCA II Semester

L	T	P
3	0	0

Object Oriented Programming using JAVA

Code: MCA1103

Course Objectives:

- To understand the basic concepts of 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 understand the implementation of packages and interfaces
- To introduce the concept of multithreading and exception handling
- To learn and understand the design of Graphical User Interface using applets and swing controls

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.

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

- Describe the uses OOP concepts
- Apply OOP concepts to solve real world problems
- Distinguish the concept of packages and interfaces
- Demonstrate the exception handing, multithread applications with synchronization
- Design the GUI based applications using AWT and Swings Discuss the

Collection Framework

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,1/e
4) JAVA& Object Orientation an Introduction, 2/e, John Hunt, Springer
5) Introduction to JAVA Programming, 7/e, Y. Daniel Liang, Pearson. , TMH

L	T	P
3	0	0

MCA II Semester

Design and Analysis of Algorithms

Code: MCA2104

Course Objectives:

- To provide an introduction to formalisms to understand, analyze and denote time complexities of algorithms
- To introduce the different algorithmic approaches for problem solving through numerous example problems
- To provide some theoretical grounding in terms of finding the lower bounds of algorithms and the NP-completeness

UNIT I:

Introduction: Algorithm, Pseudocode 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 costs panning trees, Single sources hortest 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.

Course Outcomes:

- Describeasymptoticnotationusedfordenotingperformanceof algorithms
- Analyzethe performance of a given algorithm and denote its time complexity using the asymptotic notation for recursive and non-recursive algorithms
- List and describe various algorithmic approaches
- Solve problems using divide and conquer, greedy, dynamic programming, backtracking and branch and bound algorithmic approaches
- Apply graph search algorithms to real worldproblems
- Demonstrate an understanding of NP- Completeness theory and lower bound theory

Text Books:

1) Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharam, UniversitiesPress,2/e
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.Rivestand C.Stein, PHI Pvt.Ltd

Reference Books:

1) Introduction to the Design and Analysis of Algorithms, AnanyLevitin, PEA,3/e
2) Design and Analysis of Algorithms, Pearson Education, Parag Himanshu Dave, HimansuBalachandraDave,2/e
3) Introduction to Design and Analysis of Algorithms A strategic approach, R.C.T. Lee, S.S.Tseng, R.C.Chang and T.Tsai, McGrawHill.2005
4) DesignandAnalysisofalgorithms,Pearsoneducation,Aho,UllmanandHopcrof t,1/e

L	T	P
3	0	0

MCA II Semester

NOSQL DATABASES
Code: MCA2105

Course Objectives:
The objective of the course is to:

- Define, compare and use the four types of NoSQL Databases (Document- oriented, Key Value Pairs, Column oriented andGraph)
- Demonstrate an understanding of the detailed architecture, define objects, load data, query data and performance tune Column-oriented NoSQL databases
- Explain the detailed architecture, define objects, load data, query data and performance tune Document oriented NoSQL databases
- Ability to design entity relationship model and convert entity relationship diagrams into RDBMS and formulate SQL queries on the data

UNIT I:
Introduction to NoSQL: Definition And Introduction, Sorted Ordered Column- Oriented Stores, Key/Value Stores, Document Databases, Graph Databases, Examining Two Simple Examples, Location Preferences Store, Car Make And Model Database, Working With Language Bindings.

UNIT II:
Interacting with NoSQL:If NoSql Then What, Language Bindings For NoSQL Data Stores, Performing Crud Operations, Creating Records, Accessing Data, Updating And Deleting Data

UNIT III:
NoSQL Storage Architecture: Working With Column-Oriented Databases, Hbase Distributed Storage Architecture, Document Store Internals, Understanding Key/Value Stores In Memcached And Redis, Eventually Consistent Non-Relational Databases.

UNIT IV:
NoSQL Stores:Similarities Between Sql And Mongodb Query Features, Accessing Data From Column-Oriented Databases Like Hbase, Querying Redis Data Stores, Changing Document Databases, Schema Evolution In Column- Oriented Databases, Hbase Data Import And Export, Data Evolution In Key/Value Stores.

UNIT V

Indexing and Ordering Data Sets : Essential Concepts Behind A Database Index, Indexing And Ordering In Mongodb, Creating and Using Indexes In Mongodb, Indexing And Ordering In Couchdb, Indexing In Apache Cassandra.

Course Outcomes:

After the completion of the course, student will be able to do the following

- Identify what type of NoSQL database to implement based on business requirements (key-value, document, full text, graph, etc.)
- Apply NoSQL data modeling from application specific queries
- Use Atomic Aggregates and denormalization as data modelling techniques to optimize query processing

Text Books:

1) Pramod Sadalage and Martin Fowler, NoSQL Distilled, Addison-Wesley Professional, 2012.
2) Dan McCreary and Ann Kelly, Making Sense of NoSQL, Manning Publications, 2013.

Reference Books:

1) Shashank Tiwari, Professional NoSQL, Wrox Press, Wiley, 2011, ISBN: 978-0-470-94224-6
2) Gaurav Vaish, Getting Started with NoSQL, Packt Publishing, 2013.

MCA II Semester

L	T	P
3	0	0

MOBILE APPLICATION DEVELOPMENT

Code: MCA2105

Course Objectives:

- To demonstrate the introduction and characteristics of mobile applications
- Application models of mobile application frameworks. Managing application data and User-interface design for mobile applications
- Integrating networking, the OS and hardware into mobile-applications
- Addressing enterprise requirements in mobile applications – performance, scalability, modifiability, availability and security
- Testing methodologies for mobile applications- Publishing, deployment, maintenance and management. To demonstrate their skills of using Android software development tools
- To demonstrate their ability to deploy software to mobile devices

UNIT I:

Introduction to mobile devices: Introduction to Mobile Computing, Introduction to Android Development Environment, Mobile devices vs. desktop devices, ARM and Intel architectures, Screen resolution, Touch interfaces, Application deployment, App Store, Google Play, Windows Store.

Development environments: XCode, Eclipse, VS2012, PhoneGAP, etc.; Native vs. web applications.

Factors in Developing Mobile Applications: Mobile Software Engineering, Frameworks and Tools, Generic UI Development, Android User.

UNIT II:

Android User Interface: Measurements – Device and pixel density independent measuring units User Interface (UI) Components – Editable and non editable Text Views, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers Fragments – Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities.

UNIT III:

Back Ground Running Process, Networking and Telephony Services: Services: Introduction to services – local service, remote service and binding the service, the communication between service and activity, Intent Service.

MultiThreading: Handlers, AsyncTask.

Broad cast receivers: Local Broadcast Manager, Dynamic broadcast receiver, System Broadcast. Pending Intent, Notifications.

UNIT IV:

Android: Introduction – Establishing the development environment – Android architecture – Activities and views – Interacting with UI – Persisting data using SQLite – Packaging and deployment – Interaction with server side applications – Using Google Maps, GPS and Wifi – Integration with social media applications.

Android network programming: Http Url Connection, Connecting to REST- based and SOAP based Web services.

UNIT V:

Advanced Topics: Power Management: Wake locks and assertions, Low-level OS support, Writing power-smart applications.

Augmented Reality via GPS and other sensors: GPS, Accelerometer, Camera. Mobile device security in depth: Mobile malware, Device protections, iOS "Jailbreaking", Android "rooting" and Windows" "defenestration"; Security and Hacking: Active Transactions, More on Security, Hacking Android.

Course Outcomes:

Upon completion of the course students should be able to:

- Install and configure Android application development tools
- Design and develop user Interfaces for the Android platform
- Save state information across important operating system events
- Apply Java programming concepts to Android application development

Text Books:

1) Bill Phillips, Chris Stewart, Brian Hardy, and Kristin Marsicano, Android Programming: The Big NerdRanch Guide, Big Nerd Ranch LLC, 2nd edition, 2015.
2) Valentino Lee, Heather Schneider, and Robbie Schell, Mobile Applications: Architecture, Design and Development, Prentice Hall, 2004.
3) Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox) , 2012
4) Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013
5) Dawn Griffiths, David Griffiths, *"Head First: Android Development" ,OReilly2015,ISBN*: 9781449362188
6) Jeff McWherter and Scott Gowell, "Professional Mobile Application Development", Wrox, 2012

Reference Books:

1) Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013
2) Tomasz Nurkiewicz and Ben Christensen, Reactive Programming with RxJava, O"Reilly Media, 2016.
3) Brian Fling, Mobile Design and Development, O"Reilly Media, Inc., 2009.
4) Maximiliano Firtman, Programming the Mobile Web, O"Reilly Media, Inc., 2nd ed., 2013.
5) Cristian Crumlish and Erin Malone, Designing Social Interfaces, 2nd ed., O"Reilly Media, Inc., 2014.
6) Suzanne Ginsburg, Designing the iPhone User Experience: A User- Centered Approach to Sketching and Prototyping iPhone Apps, Addison- Wesley Professional, 2010.

MCA II Semester

L	T	P
3	0	0

ARTIFICIAL INTELLIGENCE

Code: MCA2105

Course Objectives:

- To have a basic proficiency in a traditional AI language including an ability to write simple to intermediate programs and an ability to understand code written in that language
- To have an understanding of the basic issues of knowledge representation and blind and heuristic search, as well as an understanding of other topics such as minimax, resolution that play an important role in AI programs
- To have a basic understanding of some of the more advanced topics of AI

UNIT I:

Introduction: History, intelligent systems, foundations of AI, applications, tic- tac-toe game playing, development of AI languages, current trends.

UNIT II:

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 Problem reduction and game playing: Introduction, problem reduction, game playing, alpha beta pruning, two-player perfect information games.

UNIT III:

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 IV:

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.

UNIT V:

Expert system and applications: Introduction phases in building expert systems, expert system versus traditional systems

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.

Course Outcomes:

- Outline problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem
- Apply the language/framework of different AI methods for a given problem
- Implement basic AI algorithms
- Design and carry out an empirical evaluation of different algorithms on problem formalization and state the conclusions that the evaluation supports

Text Books:

1) Artificial Intelligence- Saroj Kaushik, CENGAGELearning.
2) Artificial intelligence, A modern Approach, 2nded, Stuart Russel, Peter Norvig, PEA. Rich&knight

Reference Books:

1) Artificial Intelligence- Deepak Khemani, TMH,2013.
2) Introduction to Artificial Intelligence, Patterson,PHI.
3) Artificial intelligence, structures and Strategies for Complex problem solving, George F Lugar, 5th ed,PEA.

e-Resources:

1)https://nptel.ac.in/courses/106/105/106105077/
2)http://aima.cs.berkeley.edu/

MCA II Semester

L	T	P
3	0	0

IMAGE PROCESSING

Code: MCA2105

Course Objectives:

To make the students to understand

1. The fundamentals of Computer Graphics and Image Processing
2. The concepts related edge detection, segmentation, morphology and image compression methods.

SYLLABUS:

UNIT I:

Introduction: Applications of Computer Graphics and Image Processing, Fundamentals on Pixel concepts, effect of
Aliasing and Jaggles, Advantages of high resolution systems
DDA line algorithms: Bresenhams line and circle derivations and algorithms

UNIT II:

Transformations: Translations, Scaling, rotation, reflection and shear transformations, Homogeneous
coordinates, Composite Transformations- Reflection about an arbitrary line;
Windowing and clipping, viewing
transformations, Cohen- Sutherland clipping algorithm

UNIT III:

Digital Image Properties: Metric and topological properties of Digital Images, Histogram, entropy, Visual
Perception, Image Quality, Color perceived by humans, Color Spaces, Palette Images, color Constancy
Color Images: Pixel brightness transformations, Local Preprocessing, image smoothing, Edge detectors, Robert
Operators, Laplace, Prewitt, Sobel, Fri-chen, Canny Edge detection

UNIT IV:

Mathematical Morphology: Basic Mathematical Concepts, Binary dilation and Erosion, Opening and closing, Gray
Scale dilation and erosion, Skeleton, Thinning , Thickening Ultimate erosion,
Geodesic transformations, Morphology
and reconstruction, Morphological Segmentation

UNIT V:

SEGMENTATION: Threshold detection methods, Optimal Thresholding, Edge based Segmentation- Edge image
thresholding, Edge relaxation, Border tracing, Hough Transforms,
Image Data Compression: Image data Properties, Discrete Image Transformations in data compression, Discrete
Cosine and Wavelet Transforms, Types of DWT andmerits

Course Outcomes:

- Understanding of digital image processing fundamentals: hardware and software, digitization, enhancement
  and restoration, encoding, segmentation, feature detection
- Ability to apply image processing techniques in both the spatial and frequency (Fourier) domains
- Ability To understand (i.e., be able to describe, analyse and reason about) how digital images are represented, manipulated, encoded and processed, with emphasis on algorithm design, implementation and
- performance evaluation

Text Books:

1. Computer Graphics C Version, Donald Hearn, M Paulli Baker , Pearson (Uniit I and Unit II)
2. Image Processing, Analysis and Machine Vision, Millan Sonka, Vaclov Halvoc, Roger Boyle, Cengage
3. Learning, 3ed, (Unit III, Unit IV, Unit V and Unit VI)

References:

1. Computer & Machine Vision, Theory , Algorithms , Practicles, E R Davies, Elsevier, 4ed
2. Digital Image Processing with MATLAB and LABVIEW, Vipul Singh, Elsevier
3. Digital Image Processing, R C Gonzalez &R E woods, Addison Pearson, 3ed.

MCA II Semester

L	T	P
3	0	0

ACCOUNTING FOR MANAGERS

Code: MCA2105

Course Objectives:

- To learn different Accounting Systems, preparation of Financial Statement and uses of different tools for performance evaluation. GAAP Principles
- To understand the concept of financial management and financial interpretations cost and management accounting principles and applications of standard costing and marginal costing analysis

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, labor 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.
Case study Tally

Course Outcomes:

- The Learner is able to prepare Financial Statements and the usage of various Accounting tools for Analysis and to evaluate various techniques for decision making.

Reference Books:

1) Accounting for Management, N.P.Srinivasan and M.SakthivelMurugan
2) Financial Accounting, S.N Maheswari and S.K. Maheswari, Vikas.
3) Financial Accounting, A. Mukherjee and M. Heneef, TMH.
4) Basic Financial Accounting for Management, Ambaresh Gupta, Pearson.
5) Accounts And Finance for Non accounts, Chatterjee, D.K. Himalaya.
6) Financial Analysis and Accounting, P. Premchand Babu and M. Madan Mohan, Himalaya.
7) Essential of Financial Accounting, Ashish, K and Ballacharya, PHI.
8) Guide to Financial Management, John Tannent, Viva.

MCA II Semester

L	T	P
0	0	3

DATABASE MANAGEMENT SYSTEMS LAB

Course Objectives:

1) This Course will enable students to
2) Populate and query a database using SQL DDL/DML Commands
3) Declare and enforce integrity constraints on adatabase
4) Writing Queries using advanced concepts of SQL
5) Programming PL/SQL including procedures, functions, cursors and triggers

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) Implementation of different types of joins with suitable examples
4) Create views, partitions, Sequence, Indexes and locks for a particular DB
5) Implement different types of constraints onrelations.
6) Implementation of sub queries and nested queries.
7) Implement Queries on Group By & Having Clauses, ALIAS, Sequence By, Order By
8) Control Structure
    a) Write a PL/SQL block for Addition of TwoNumbers
    b) Write a PL/SQL block for IF, IF and else condition
    c) Write a PL/SQL block for implementation of loops
    d) Write a PL/SQL block for greatest of three numbers using IF ANDELSEIF
9) ExceptionHandling-Implementthefollowingwithrespectto exception handling.
    Raising Exceptions, User Defined Exceptions, Pre-Defined Exceptions
10) Write PL/SQL block for an application using exception handling*Procedures*
    a) Write a PL/SQL Procedure using Positional Parameters
    b) Write a PL/SQL Procedure using notationalparameters
    c) Write a PL/SQL Procedure for GCDNumbers
    d) Write a PL/SQL Procedures for cursor implementation (explicit and implicit cursors)
11) Functions:
    a) Write a PL/SQL block to implement factorial using functions
    b) Write a PL/SQL function to search an address from the given database
12) Write a DBMS program to prepare Pl/SQL reports for an application using functions.

Triggers:

    a) Write a Trigger to pop-up the DMLoperations

b) Write a Trigger to check the age valid or not Using Message Alert.
c) Create a Trigger to Raise appropriate error code and error message.
d) Create a Trigger on a table so that it will update another table while inserting values

13) Write PL/SQL block for an application using cursors and all types of triggers.
14) Write a PL/SQL block for transaction operations of a typical application using package

Course Outcomes:

1) At the end of the course the student will be able to:
2) Utilize SQL to execute queries for creating database and performing data manipulation operations
3) Examine integrity constraints to build efficient databases
4) Apply Queries using Advanced Concepts of SQL
5) Build PL/SQL programs including stored procedures, functions, cursors and triggers

Text Books:

1) Oracle: The Complete Reference by Oracle Press
2) Nilesh Shah, "Database Systems Using Oracle", PHI, 2007
3) Rick F Vander Lans, "Introduction to SQL", Fourth Edition, Pearson Education, 2007

MCA II Semester

L	T	P
0	0	3

Computer Networks Lab

Course Objectives:

At the end of the course, the students will be ableto:

- Understands the fundamental concepts of computer networking and OSI Reference model.
- Familiarize the student with the basic taxonomy and terminology of thecomputer networking area.
- Learn and understand the advanced networking concepts, preparing thestudent for entry advanced courses in computer networking.
- Develop and gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

PART - A

1) Implement the data link layer farming methods such as character stuffing and bit stuffing.
2) Implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.
3) Implement Dijkstra„s algorithm to compute the Shortest path through a graph.
4) Take an example subnet graph with weights indicating delay between nodes. Now obtain Routing table art each node using distance vector routing algorithm
5) Take an example subnet of hosts. Obtain broadcast tree for it.

PART - B

1) Implement the following forms of IPC.
   a. a)Pipes b)FIFO
2) Implement file transfer using Message Queue form of IPC
3) Write a programme to create an integer variable using shared memory concept and increment the variable
4) Simultaneously by two processes. Use semaphores to avoid race conditions
5) Design TCP iterative Client and server application to reverse the given input sentence
6) Design TCP client and server application to transfer file
7) Design a TCP concurrent server to convert a given text into upper case using multiplexing system call "select"
8) Design a TCP concurrent server to echo given set of sentences using poll functions
9) Design UDP Client and server application to reverse the given input sentence
10) Design UDP Client server to transfer a file
11) Design using poll client server application to multiplex TCP and UDP requests for converting a given text into upper case.
12) Design a RPC application to add and subtract a given pair of integers

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

- Explain the network architecture, TCP/IP and OSI reference models
- Identify and understand various techniques and modes of transmission
- Demonstrate the 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
- Develop network security and define various protocols such as FTP, HTTP, Telnet, DNS

Text Books:

1. Computer Networks: Andrew S Tanenbaum David J. Wetherall, 5/e, Pearson
2. Data communications and networking: Behrouz Forouzan, 5/e, McGraw Hill
3. Data and computer Communications, William Stallings, PEARSON, 10/e

Reference Books:

1. Computer Networks – A System Approach, Peterson, Bruce Davie, 2/e , Harcourt Asia
2. Compute communications and networking technologies, Gallo, Hancock, Cengage
3. An Engineering approach to compute networking, Kesha, Pearson

MCA II Semester

L	T	P
0	0	3

JAVA Programming Lab

Code: MCA1108

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

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).

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.

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,1/e
4. JAVA& Object Orientation an Introduction, 2/e, John Hunt, Springer
5. Introduction to JAVA Programming, 7/e, Y. Daniel Liang, Pearson. , TMH

L	T	P
0	0	1

MCA II Semester

Employability Skills

Course Objectives:

The main of this course is

- To learn how to make effective presentations and impressive interviews
- To learn skills for discussing and resolving problems on the work site
- To assess and improve personal grooming
- To promote safety awareness including rules and procedures on the work site
- To develop and practice self management skills for the work site

A list of vital employability skills from the standpoint of engineering students with discussion how to potentially develop such skills through campus life.

1) Soft Skills: An Introduction – Definition and Significance of Soft Skills; Process, Importance and Measurement of Soft Skill Development.
2) Self-Discovery: Discovering the Self; Setting Goals; Beliefs, Values, Attitude, Virtue.
3) Positivity and Motivation: Developing Positive Thinking and Attitude; Driving out Negativity; Meaning and Theories of Motivation; Enhancing Motivation Levels.
4) Time Management – Concept, Essentials, Tips.
5) Personality Development – Meaning, Nature, Features, Stages, Models; Learning Skills; Adaptability Skills.
6) Decision-Making and Problem-Solving Skills: Meaning, Types and Models, Group and Ethical Decision-Making, Problems and Dilemmas in application of these skills.
7) Conflict Management: Conflict - Definition, Nature, Types and Causes; Methods of Conflict Resoultion.
8) Stress Management: Stress - Definition, Nature, Types, Symptoms and Causes; Stress Analysis Models and Impact of Stress; Measurement and Managemet of Stress
9) Leadership and Assertiveness Skills: A Good Leader; Leaders and Managers; Leadership Theories; Types of Leaders; Leadership Behaviour; Assertivness Skills.

Note: The student shall be instructed to Record a 2 min video and add to profile before and after taking the course. Students are to be involved in Role Play, Team dynamics, Group Discussion and outcomes are to be recorded.

Course Outcomes:

By the end of this course, the student

- Recite the soft skills
- Make presentations effectively with appropriate body language
- Be composed with positive attitude
- Apply their core competencies to succeed in professional and personal life

Text Books:

- Barun K. Mitra, Personality Development and Soft Skills, Oxford University Press, 2011.
- S.P. Dhanavel, English and Soft Skills, Orient Blackswan, 2010.
- R.S.Aggarwal, A Modern Approach to Verbal & Non-Verbal Reasoning, S.Chand& Company Ltd., 2018.

Reference Books:

- Raman, Meenakshi & Sharma, Sangeeta, Technical Communication Principles and Practice, Oxford University Press, 2011.
- Managing Soft Skills for Personality Development – edited by B.N.Ghosh, McGraw Hill India, 2012.
- English and Soft Skills – S.P.Dhanavel, Orient Blackswan India, 2010.