Department of Computer Science

Gujarat University

MCA

(2 Years)

New Scheme & Syllabus

Effective From: July 2020

NEW SCHEME FOR MASTER OF COMPUTER APPLICATOINS (MCA) COURSE

SEMESTER - I of MCA

Applicable from July 2020 onwards

Sr. No.	Subject Code	Name of the Subject	TEACHING SCHEME					EXAMINATION SCHEME					
			THEORY	TUTORIAL	PRACTICAL	SESSIONAL		THEORY		PRACTICAL		TERM WORK	TOTAL MARKS
			HR.	HR.	HR.	Marks.	HR.	Marks.	HR.	Marks.	HR.		
1.	MCA111	Bridge Course#	4	-	-	-	-	-	-	-	-	-	**100**
2.	MCA112	Mathematical Foundations	3	1	-	25	2	50	3	-	-	25	**100**
3.	MCA113	Object Oriented Concepts & Programming	4	-	3	25	2	50	3	50	3	25	**150**
4.	MCA114	Data Structures	4	-	3	25	2	50	3	50	3	25	**150**
5.	MCA115	Relational Database Management Systems	3	1	3	25	2	50	3	50	3	25	**150**
6.	MCA116	Elective-1 (Any one subject from Track1/2/3)	4	-	3	25	2	50	3	50	3	25	**150**
Total			**22**	**2**	**12**	**125**	**-**	**250**	**-**	**200**	**-**	**125**	**700**

Marks not counted for Class or Grand Total

Subjects for Elective - I

1. Introduction to Python Programming (Track - 1)
2. Introduction to Linux Programming (Track - 2)
3. Web Application Development (Track - 3)

Track No	Track-1	Track-2	Track-3
Track Name	Artificial Intelligence & Machine Learning	Networking	Web Technologies
Subject	MCA 116(1)	MCA 116(2)	MCA 116(3)

DEPARTMENT OF COMPUTER SCIENCE, GUJARAT UNIVERSITY

TEACHING SCHEME (WITH EFFECT FROM JULY 2020)

M.C.A. (Master of Computer Applications)

Semester– I

Subject Code	Subject	Hours per week			Credits*
		Theory	Tutorial	Lab	
MCA111	Bridge Course#	4	-	-	4
MCA112	Mathematical Foundations	3	1	-	4
MCA113	Object Oriented Concepts & Programming	4	-	3	6
MCA114	Data Structures	4	-	3	6
MCA115	Relational Database Management Systems	3	1	3	6
MCA116	Elective-1 (Any one subject from track1/2/3)	4	-	3	6
Total		**22**	**2**	**12**	**32**

Marks not counted for Class or Grand Total

* Theory & Tutorial: - 1 hour = 1 credit
Practical: - 1 hour = 2/3 credit

Subjects for Elective - I

1. Introduction to Python Programming (Track - 1)
2. Introduction to Linux Programming (Track - 2)
3. Web Application Development (Track - 3)

Note: *Students are required to select a single track for the electives throughout the MCA program*

MCA - I
Bridge Course Syllabus

Subject Name: Bridge Course

Course Code: MCA111

Module: Fundaments of Computer Organization

Objectives:
Students will learn

- The elements of Computer Organization and Architecture
- The basic knowledge necessary to understand the number system, gates operations of digital computers

Prerequisites:
Any programming language like C, C++

Contents:

1. **Number Systems**
   Decimal System, Bistable Devices, Binary, Octal and Hexadecimal numbers, Number Base conversions, Binary Addition, Subtraction, Multiplication, Division, *Complements*: Use of complements to represent Negative Numbers, Binary Number Complements, Complements in other Number Systems

2. **Logic Gates**
   Logical Multiplication, AND & OR Gates, Complementation & Inverters, Evaluation of Logical Expressions, Design Using NAND Gates, Design Using NOR Gates, NAND to AND & NOR to OR gate Networks

3. **Logic Design**
   Flip-Flops, Transfer Circuit, Clocks, Flip-Flop Designs

4. **Digital Components**
   Integrated Circuits: Decoders, NAND gate Decoder, Decoder Expansion, Encoders, Multiplexers

References:

1. Thomas C. Bartee, “Digital Computer Fundamentals”, Tata McGraw Hill
2. Leach D., Malvino A., "Digital Principles and Applications", McGraw Hill Education

Accomplishments of the student after completing the course:

After completion of this course, students will be able to

- Understand the computer organization and architecture, number system, gates operations of digital computers

***** ***** *****

Module: Database Management Systems

Objectives:

Students will learn

- To the fundamentals of Database Management Systems (DBMS)
- To understand the Structured Query Language
- To get Familiarized with various types of DBMS

Prerequisites:

Any programming language like C, C++

Contents:

1. **Basic Concepts of Database Management System**
   Fundamental concepts of File and Databases, Overview of Physical Storage Media, file organization, Purpose of Database system, *Introduction to Data models*: Conceptual Data model, ER model, *Record based Data models*: Hierarchical, Relational, Network, Features of Database Systems, Overall system structure

2. **Relational Database Design**
   Relational structure: Tables (relations), Rows (tuples), Domains, Columns (attributes), *Keys*: Super key, candidate keys, primary key, entity integrity constraints, referential integrity constraints, Database design process, Anomalies in a Database

3. **Query Languages**
   Introduction to SQL, Advantages of using SQL, *Type of Database Languages*: DDL, DML, TCL, Introduction to Data definition and Data Manipulation language commands, Basic queries in SQL

4. **Data Dictionary**
   Introduction to Data Dictionary, Usage of Data Dictionary, Components of Data Dictionary

References:

1. S. Sudarshan, H.F. Korth, A. Silbershatz, "*Database System Concepts*", Tata McGraw Hill
2. Ivan Bayross, "*SQL - MySQL – MySQL 5 for Professionals"*, Shroff Publishing Series
3. Ramakrishnan, Gehrke, "*Database Management Systems*", Tata McGraw Hill
4. Michael V. Mannino, "*Database Design Application Development and Administration*", Tata McGraw Hill
5. S.K.Singh, "*Database Systems: Concepts, Design and Applications*", Pearson Education

Accomplishments of the student after completing the course:

After completion of this course, students will be able to

- Understand basic concepts of Database Management System

******* ***** *******

Module: Computer Networking

Objectives:

Students will learn

- To get familiar with layered communication architectures (OSI and TCP/IP)
- To understand the client/server model and key application layer protocols
- To use of transmission media and various devices for networking
- To understand the concept of protocols and familiarize with widely used protocols
- To understand concepts of network security

Prerequisites:

None

Contents:

1. **Overview of Computer Networks**
   Introduction to Computer Networks, Applications of Computer Networks, Advantages of Computer Networks, Types of Network Connections, Network Topologies, Categories of Computer Networks (LAN, MAN, WAN, BAN, PAN), Transmission Media, Devices for Networking

2. **Introduction to Protocols**
   Overview of OSI Model, TCP/IP Model, Physical and Logical Addressing, widely used protocols like IP, TCP, UDP, HTTP, DHCP, DNS, FTP, Telnet, SSH, SMTP, IMAP, POP, MIME, ICMP

3. **Network Security Fundamentals**
   Fundamentals of Cryptography, Symmetric Key and Asymmetric Key Cryptography, Overview of technologies like NAT, VPN, Firewall, Intrusion Detection System

References:

1. Bhushan Trivedi, "*Computer Networks*", Oxford University Press, India
2. James Kurose and Keith Ross, "*Computer Networking: A Top-Down Approach*", Pearson Education
3. Andrew S. Tanenbaum, "*Computer Networks*", Pearson Education
4. Edelman J., Lowe S., Oswalt M., "*Network Programmability and Automation*", O'Reilly Press

Accomplishments of the student after completing the course:

After completion of this course, students will be able to

- Understand the role of protocols in computer networks and become capable of identifying network security techniques in day to day usage of networks

***** ***** *****

Module: Fundamentals of Programming

Objectives:

Students will learn

- To introduce the rudiments of programming to the students.
- Students will become familiar with problem solving techniques and algorithm development using computers.
- Students will understand programming using C, a high-level programming language.

Prerequisites:

None

Contents:

1. **Introduction to programming & Basics of C**
   Generation of languages, Basic features of C Language like Identifier, Keywords, Variable, data types, Operators and Expression, Basic screen and keyboard I/O

2. **Control Statements**
   Test Conditions, Conditional execution and selection, Iteration and Repetitive Executions, Nested loops

3. **Arrays**
   Introduction to contiguous data types. One dimensional arrays, Multidimensional arrays, Array as strings, multidimensional character arrays, Operations on strings

References:

1. Pradip Dey & Manas Ghosh, "*Programming in C*", Oxford University Press, India
2. Balagurusamy, "*Programming in ANSI C ",* Tata McGraw Hill
3. Behrouz A. Forouzan& Richard F. Gilberg, "*Computer Science: A Structured Programming Approach Using C s*", Thomson Education
4. Ashok N Kamthane, "*Programming with ANSI and Turbo C*", Pearson Education
5. Venugopal & Prasad, "*Mastering C*", Tata McGraw Hill
6. Herbert Schildt, "C: The Complete Reference", Tata McGraw Hill
7. Yashwant Kanitkar, "*Let us C*", BPB Publication
8. Byron Gottfried, "*Schaum's Outline of Programming with C*", Shaum Series

Accomplishments of the student after completing the course:

After completion of this course, students will be able to

- Solve problems using computers through C programming language

***** ***** *****

Module: Client-Side Scripting

Objectives:

Students will learn

- The basic knowledge about Internet
- To understand the fundamentals of web designing
- To use HTML and JavaScript to develop the web site

Prerequisites:

Basic knowledge of Computer operations and its functionality

Contents:

1. **Introduction to Internet, HTML5**
   Introduction to fundamentals of Internet, Introduction to HTML5, Editors, Introduction to the basic of web page, Overview of elements, attributes, Styles and Formatting

2. **HTML5 and CSS**
   Overview of anchors, hyperlinks, Lists, Tables and comments, Overview of Forms, Introduction to CSS, Types of CSS

3. **Introduction to JavaScript**
   Introduction to JavaScript, JavaScript code with HTML, Variables and Datatypes, Operators, Control Structure in JavaScript, Loops and Introduction to array

4. **JavaScript advance and DOM**
   Introduction to Functions, Introduction to Object, Introduction to DOM

References:

1. John Dean, "*Web Programming with HTML5, CSS, and JavaScript,*", Jones & Bartlett Learning
2. John Paul Mueller, "*HTML5 Programming with JavaScript for Dummies*", John Wiley & Sons, Inc.
3. Jonathan Lane, Tom Barker, Joe Lewis, Meitar Moscovitz, "*Foundation Website Creation with HTML5, CSS3 and JavaScript*", Apress

Accomplishments of the student after completing the course:

After completion of this course, students will be able to

- Understand the basic concepts of HTML5, CSS and JavaScript and their application for the development of Website
- Develop a dynamic website

***** ***** *****

Course Name: Mathematical Foundations

Course Code: MCA112

Objectives:

Students will be introduced to

- The concepts of calculus, vectors, data representation and linear algebra
- Apply these concepts to real life problems and machine learning problems

Prerequisites:

Basic knowledge of mathematical fundamentals

Contents:

1. **Introduction to Set Theory**
   Basic concepts, notations, Inclusion and equality, Power set, Operations of union, Intersection and complement, Venn diagrams, Set identities (associative, distributive, etc.), Ordered pairs and n-tuples, Cartesian product

2. **Fundamentals of Single Variable Calculus**
   Functions of single variable, Definition and their graphs, Special functions like polynomials, Trigonometric, Exponential, Hyperbolic, Limit, Continuity, Definition of derivative and its graphical meaning, Rules of differentiation, Chain rule, Higher order derivatives, Definition of integration and its geometric interpretation, Indefinite and definite integral and their evaluation, Optimization of functions: Local Maxima and minima of functions, Saddle point, Necessary and sufficient conditions, Global maxima, Convex functions, Taylor Series

3. **Graph and Tree**
   Basic concepts of Graph theory, Paths, Reachability and connectedness, Matrix representation of graphs, Trees

4. **Fundamentals of Vectors**
   Definition of vector, Scalars, Addition and subtraction of vectors, Scalar multiplication, Inner product (dot product) of vectors, Norms, Direction, Orthogonal vectors, Projection of vectors, Cosine similarity, Normal and orthonormal vectors, Gram-Schmidt procedure, Orthogonal decomposition

5. **Introduction to Matrices**
   Definition, addition of two matrices, transpose, scalar multiplication, matrix multiplication, properties of matrix addition and multiplication, square matrix, null and identity matrix, invertible matrix and inverse, hadamard product and its properties, determinant of a square matrix and its properties, rank, trace, popular type of matrices symmetric, diagonal, orthogonal, orthonormal

6. **Introduction to Numerical Methods**
   Introduction, Characteristic of numerical methods, Types and sources of errors in data, Quantification of errors, Nature of iterative methods to find a solution, Numerical methods of finding roots of an equation f(x) = 0: Bisection method, False position method, Secant method, Newton Raphson Method, Gradient Descent method

7. **Linear Equations**
   Systems of Linear Equations, Cramer's Rule, Elementary row operations, Row reduced and Echelon forms, Homogeneous Systems, Matrix inversion method

References:

1. Erwin Kreyszig, "*Advanced engineering Mathematics*", 10th Edition, ISV, John Wiley and sons, INC.
2. J.P. Trembly, R. Manohar, "*Discrete mathematical structures with applications to computer science*", McGraw Hill Education
3. David C. Lay, "*Linear Algebra and its applications*", 3rd Edition, Pearson
4. Dr. Chandrika Prasad, "*Advanced Mathematics for engineers*", Pothishala Private Ltd.

Accomplishments of the student after completing the course:

After completion of this course, students will be able to

- Develop problem solving skills using mathematics techniques such as geometry, calculus and vectors
- Interpret, analyze and represent real life data into mathematical models
- Use the knowledge of mathematical models in various computer science domains such as artificial intelligence, network security, trend analysis and many more

***** ***** *****

Course Name: Object Oriented Concepts & Programming

Course Code: MCA113

Objectives:

Students will be introduced to

- Differentiate between procedural and object oriented programming
- Learn C++ as a language and various features to implement Object oriented concepts
- Learn Object Oriented principles and their application using C++

Prerequisites:

Knowledge of C language, Programming concepts including algorithm building and logic

Contents:

1. **Introduction to C++**
   Identifiers and constants (Literals), Keywords, Data Types, The Operators, New Casting Operators, Typeid and throw, The Conditional structures and Looping Constructs, The Difference between struct and class in C++, The difference between Union and Class, Static Data members of a class, Pointer to objects and pointer to members of class, The local classes, Assigning Objects

2. **Functions**
   Introduction, Inline function, Default Arguments to the function, Functions with object as parameters, Call by reference and return by reference, Prototyping and Overloading, Friend functions, Const and Volatile functions, Static functions, Private and Public functions, Function Pointers, Adding C functions to the C++ program

3. **Constructors and Destructors**
   Introduction to constructors, The explicit constructors, Parameterized constructors, Having multiple constructors, Constructors with default arguments, Dynamic Initialization, Constructor with dynamic allocation, Copy constructors, The member initialization list, Destructors

4. **Operator Overloading and User Defined Conversions**
   Introduction, Unary Operators, Binary Operators, Using Friends as operator functions, Overloading other Operators, The need for user defined conversion, Four different cases where user defined conversions are needed, Comparison of both the methods of conversion

5. **Templates**
   Function Templates, Non-Generic (Non-Type) Parameters in Template functions, Template function and specialization, Overloading a template function, Using Default Arguments, Class Templates, Classes with multiple generic data types, Static data members, Primary and Partial Specialization, The Export Keyword, The other use of typename

6. Inheritance

The need, Defining derived class using single base class, Derivation using public, private and protected access modifiers, The implementation of inheritance in the C++ object model, The Access Control, The Access Declaration, The multiple-inheritance, Abstract classes, Composite objects (container objects)

7. Runtime polymorphism by virtual functions

Compile Time and Runtime Polymorphism, Pointers to Objects, *this* pointer, Compatibility of Derived and base class pointers, The subobject concept, Virtual functions, Static invocation of virtual function, Default arguments to virtual functions, Virtual destructors, Pure virtual functions

8. IO Streams

Need for streams, Advantages of using C++ I/O over C IO, The C++ Predefined streams, Formatting IO, Formatting using *ios* members, Manipulators, Creating our own manipulator

9. Using Files for IO

Why IO is special, Text and binary streams, Opening and closing files, Dealing with text files Dealing with binary files, Providing Random Access using seek, IO Modes, Handling Errors

10. Namespaces

Introduction and need, Use the using syntax, Defining namespaces, Extending the namespace, Unnamed namespaces, Nested Namespaces, Namespace aliases, The *std* namespace, The Koenig lookup, Overhead with namespaces

11. The Standard Template Library

The STL (Standard Template Library) Introduction, Generic Programming, Generic Software Components, Generic Algorithms, Iterators, Containers, Algorithms

References:

1. Bhushan Trivedi, "*Programming with ANSI C++*", Oxford University Press
2. E. Balagurusamy, "*Object-Oriented Programming with C++*", McGraw Hill Publications
3. Marshall Cline, "*C++ FAQs*", Pearson Education
4. Stanley Lippmann, "*C++ Primer*", Pearson Education
5. Bjarne Stroustrup, "*The C++ Programming Language*", Pearson Education
6. Scott Mayer, "*Effective C++*", Addison-Wesley
7. Herbert Schildt, "*Complete Reference C++*", McGraw Hill Publications
8. ePgPathshala, "*Subject: Information Technology – P-01. Object Oriented Concepts and Programming using C++*", http://epgp.inflibnet.ac.in/Home/ViewSubject?catid=305

Accomplishments of the student after completing the course:

After completion of this course, students will be able to

- Understand and appreciate the Object Oriented approach of programming
- Get aware of the working and architectural model of C++
- solve problems given to him/her using C++ with keeping balance between efficiency and flexibility

***** ***** *****

Course Name: Data Structures

Course Code: MCA114

Objectives:

Students will be introduced to

- Develop proficiency in the specification, representation, and implementation of Data Types and Data Structures
- Be able to carry out the Analysis of various Algorithms for mainly Time and Space Complexity
- Get a good understanding of applications of Data Structures
- Develop a base for advanced computer science study

Prerequisites:

Any programming language like C, C++

Contents:

1. **Introduction**
   Data types, ADT, data structure: Definition & classification, Analysis of algorithms (recursive and non-recursive) with emphasis on best case, average case and worst case

2. **Structures, pointers & Dynamic Memory Allocation**
   Structures, Nesting of structures, Arrays of structures, Structures and Pointers, Structures and functions, Union and its usage, *Pointer*: Pointer arithmetic, Array and Pointers, Pointers and strings, Pointer to Pointer, Pointers and functions, Introduction to dynamic memory allocation, Allocating memory dynamically, resizing and releasing dynamically allocated memory

3. **Linear Data structures with applications**
   Array data structure: Storage, Mapping, Applications (sparse matrix, polynomial representation, strings), *List*: Introduction, Implementation using array & linked list (singly, doubly, circular, multi-list), *Applications*: Polynomial, representation, Sparse matrix, *Stack*: Introduction, Implementation using array & linked list, *Applications*: Function call, Recursion, balancing of parenthesis, *Polish Notation*: infix to postfix conversion and evaluation of postfix expression, *Queue*: Introduction (queue, circular queue, deque, priority queue), implementation using array & linked list, *Applications*: Job Scheduling

4. **Non-Linear data structures**
   Tree: Introduction and representation, Forest, Tree traversal, Binary Tree (representation using array and links), Binary tree, Traversal (recursive& non-recursive implementation), Expression tree, *Graph*: Introduction, Representations, Traversal (BFS, DFS), *Applications*: Shortest path (Single source-all destinations), Minimal spanning tree (Prim's algorithm, Kruskal's algorithm)

5. Searching and Sorting

Linear Search, Binary Search, Transpose sequential search, Binarysearch tree, Heap tree (application in priority queue and sorting), AVL tree, Splay tree, M-way search tree, B tree (insertion), B+ tree (Definition and introduction), B* tree (Definition and introduction), Tries, Application of B tree and B+ tree in File Structures, Hash *Tables*: Introduction, hash functions and hash keys, Collisions, Resolving collisions, Rehashing, Sorting with algorithm analysis (best case, worst case, average): Bubble, Selection, Insertion, Shell, Merge, Quick, Heap, Radix

Notes:
Term work is to be carried out as per the above syllabus
Data Structures to be implemented in any programming language

References:

1. Mark Allen Weiss, "*Data Structures and Algorithm Analysis in C*", 2nd Edition, Pearson Education
2. G. A.V. PAI, "*Data structures and algorithms, concepts, Techniques and Applications*", 1st Edition, TMH
3. Horowitz, Sahni, Anderson-Freed, "*Fundamentals of Data Structures in C*", 2nd edition, University Press
4. Jean-Paul Tremblay, Paul G. Sorenson, "*An Introduction to Data Structures with Applications*", 2nd Edition, Tata McGraw-Hill
5. Cormen, Leiserson, Rivest, Stein, "*Introduction to Algorithm*", 2nd Edition, PHI
6. Gilberg&Forouzan, "*Data Structures: A Pseudo-code Approach with C*", Thomson Learning
7. Parag Dave & Himanshu Dave, "*Design and Analysis of Algorithms*", Pearson Education
8. Tanenbaum, "*Data Structures Using C & C++*", PHI
9. Michel Goodrich, Roberto Tamassia, "*Algorithm design-foundation, analysis & internet examples*", Wiley
10. A. V. Aho, J. E. Hopcroft, J. D. Ullman, "*Data Structures & Algorithms*", Addison-Wesley Publishing
11. Michael Berman, "*Data Structures Via C++: Objects by Evolution"*, Oxford University Press
12. D. E. Knuth, "*Sorting & Searching - The Art of Computer Programming*", Vol. 3, Addison- Wesley Publishing

Accomplishments of the student after completing the course:

After completion of this course, students will be able to

- Decide the appropriate data type and data structure for a given problem
- Select the best algorithm to solve a problem by considering various problem characteristics, such as the data size, the type of operations, etc.
- Ability to compare algorithms with respect to time and space complexity

***** ***** *****

Course Name: Relational Database Management Systems

Course Code: MCA115

Objectives:

Students will be introduced to

- Give students a solid background in Database management systems
- NoSQL systems
- Carry out the analysis of various algorithms for mainly *time* and *space* Complexity

Prerequisites:

Basics of Database Management Systems

Contents:

1. **Data Modeling/Conceptual Design & Relational Data Model**
   Data models: Introduction, Three level architecture, Overall architecture of DBMS, Various components of a DBMS, Enhanced ER diagrams, Database modeling using entity and relationships, Enhanced entity-relationship diagrams

2. **Relational Database Design**
   Relational structure: tables (relations), rows (tuples), domains, columns (attributes), *Keys*: super key, candidate keys, primary key, Entity integrity constraints, Referential integrity constraints, Database design process, Anomalies in a Database, *Functional Dependencies*: Lossless decomposition, Dependency preservance, Closure set of FD, Canonical cover, Lossless Joins, Finding Candidate keys using Armstrong rules, *Stages of Normalization*: 1NF ,2NF ,3NF, BCNF (with general definition also) and Multi-valued Dependency : 4NF & 5NF(Project Join NF), Translation of E-R schemes (logical design) to relational schemes (physical design): *A case study*

3. **Relational Algebra**
   Basic operators: Select, project, union, set, difference, cartesian product and rename, *Additional operators*: Set interaction, Natural Join, Division and Assignment operator, Insert, Update, Delete operators

4. **Transaction Management and Concurrency Control**
   Transaction management: ACID properties, serializability and concurrency control, Lock based concurrency control (2PL, Deadlocks), Time stamping methods, optimistic methods, Database recovery management

5. **Database Backup and Recovery**
   Need of Database backup, Database backup techniques, Types of Database failures, *Types of Database recovery*: Forward recovery, backward recovery and Media recovery

6. **Security**
   Introduction, Discretionary access control, Mandatory Access Control, Data Encryption

7. **Introduction to NoSQL**
   What is NoSQL, Types of NoSQL Databases, How NoSQL Databases work, Distributed Databases, Introduction to Mongodb, SQL vs NoSQL, Document Database, Use of JSON with document Database, JSON in relational Databases

8. **Introduction to Other Databases**
   Parallel Databases, *Database System architectures*: Centralized and Client-Server architectures, Server System architectures, Parallel systems, *Parallel Databases*: I/O Parallelism, Inter and intra query parallelism, Inter and intra operation parallelism, *Distributed Database Concepts*: Distributed Data Storage, Distributed transactions, Commit protocols, Concurrency control, Distributed query processing, Multidimensional Databases and their uses in data analytics, *Temporal Databases*: Introduction to Temporality, Temporal relationships, Temporal hierarchies, *Spatial Databases*: Spatial data types, Spatial relationships, Topological relationships, Spatial data structures and methods of storage

9. **Query languages &Procedural SQL Practical only**
   SQL Concepts: Basics of SQL, DDL, DML, DCL, *Structure*: Creation, Alteration, *Defining constraints*: Primary key, Foreign key, Unique, Not NULL, Check, IN operator, *Functions*: Aggregate functions, Built-in functions (numeric, date, string functions), Set operations, Sub-queries, Correlated sub-queries, Use of group by, Having, Order by, Join and its types, Exist, Any, All, View and its types, *Transaction control commands*: Commit, Rollback, Savepoint¸ *PL/SQL Concepts*: Cursors, Stored Procedures, Stored Function, Database Triggers

References:

1. S. Sudarshan, H.F. Korth, A. Silbershatz, "*Database System Concepts*", Tata McGraw Hill
2. Ivan Bayross, "*SQL - MySQL – MySQL 5 for Professionals*", Shroff Publishing Series
3. Ramakrishnan, Gehrke, "*Database Management Systems*", Tata McGraw Hill
4. Michael V. Mannino, "*Database Design Application Development and Administration*", Tata McGraw Hill
5. S.K.Singh, "*Database Systems: Concepts, Design and Applications*", Pearson Education
6. Peter Rob, Carlos Coronel, "*Database Systems: Design, Implementation and Management*", Thomson Course technology
7. Mongodb NoSQL Documentation, https://www.mongodb.com/nosql-explained
8. Mongodb Documentation, https://www.mongodb.com/document-Databases

Accomplishments of the student after completing the course:

After completion of this course, students will be able to

- Effectively squeeze the *real-world* data into the required data models of the Database system and retrieve the data afterwards

***** ***** *****

Course Name: Introduction to Python Programming

Course Code: MCA116(1)

Objectives:

Students will be introduced to

- Introduce the core concepts of Python Programming
- Introduce the basic insight of programming using Python libraries and how to use functionality of various Python libraries for various tasks
- Give hands on with major focus on practical implementation of these concepts

Prerequisites:

Fundamentals of Computers

Contents:

1. **Introduction**
   Introduction to Python as a programming language, Introduction to various python programming editors (IDE), Python Virtual Machine, Memory Management in Python, Garbage Collector, Writing and Running Python programs, Built-in Data types, values and variables in Python, Evaluating Expressions and performing operations, Operators in Python, Conditional Execution in Python, Using iterations within Python programs, Input and Output in Python

2. **Functions in Python**
   Introduction to Functions, Built-in Functions, Standard Functions: Mathematical Functions, time Functions, Random Numbers, System-specific Functions, The *eval* and *exec* Functions, Writing Functions, Function Basics Parameter Passing Documenting Functions, User defined Functions Vs. Standard Functions, Global Variables, Default Parameters, Introduction to Recursion, Functions as Data, Anonymous Functions, Function Decorators, Generators

3. **Arrays using array module and numpy() in Python**
   Introduction to Array, Array using array module, Introduction to *numpy* module, Creating and importing Array, Types of Array, Indexing and slicing on Array, Mathematical operations on Array, Comparing Arrays, Aliasing Arrays, Attributes of an Array, Array methods, Basic operations on Multi-dimensional Array, Matrices in numpy, Random Numbers

4. **Lists, Tuples, Dictionaries & Sets in Python**
   Lists: Introduction to Lists, Use of List, Building Lists, List Membership, List Assignment and Equivalence, List Bounds, Slicing, List Element Removal, Lists and Functions, List Methods, Prime Generation with a List, Command-line Arguments, List Comprehensions, Multidimensional Lists, Lists Vs. Generators
   Tuples, Dictionaries, Sets: Tuples, Basic operations on Tuple, Nested Tuple, Dictionaries, Dictionary Methods, Counting with Dictionaries, Grouping with Dictionaries, Keyword Arguments, Sets, Set Quantification with all and any, Enumerating the Elements of a Data Structure

4. **Strings in Python**
   Creating Strings, Indexing and Slicing in Strings, Concatenation of Strings, membership in Strings, Comparing Strings, Replacing a String, Splitting and Joining Strings, Changing case in Strings, String Testing methods, Formatting Strings, Sorting Strings, Substring, Inserting Substring into String

5. **Matrix in Python**
   Importing Matrix, basic operations on Matrix: finding maximum and minimum elements, Sum and average of elements, Products of elements, Sorting the Matrix, Transpose of a Matrix, Matrix operations, Diagonal elements of a Matrix, Random numbers

6. **Introduction to Object Oriented programming with Python**
   Introduction of OOPS, Classes and Objects: Creating Class, Constructor, Types of variables, types of Methods, Inner Classes, Encapsulation, Abstraction, Inheritance, Types of Inheritance, Overriding Super Class Constructors and Methods, super(), Method Resolution Order, Polymorphism, Duck typing philosophy of Python, Operator Overloading, Method Overloading, Method Overriding, Abstract Method, Abstract Class, Interface in Python, Abstract Classes vs. Interface

7. **Exceptions**
   Errors in Python, Exception and Exception handling, Types of Exceptions, *Except* block, *assert* statement, User-Defined Exceptions, Logging the Exceptions

8. **Files in Python**
   Files, Types of Files, Opening and Closing Files, Working with Text Files, Working with Binary *Files*, with statement, Pickle in Python, seek() and tell() methods, Random access of Binary Files, Zipping and unzipping Files

9. **Overview of Objects and Modules used with Python**
   Objects: Introduction to Objects, Introduction to Turtle Graphics Objects, Graphics with tkinter Objects, Date and time Objects, *import* statement, Introduction to *Pandas* and *Matplotlib* modules

References:

1. Rao N.R., "*Core Python Programming*", Dreamtech Publication India
2. Sarker M.O.F., "*Python Network Programming Cookbook*", Packt Publication
3. Halterman R., "*Fundamentals of Python Programming*", Southern Adventist University
4. Guttag J.V., "*Introduction to Computation and Programming Using Python*", Prentice Hall India
5. Willi Richert, "*Building Machine Learning Systems with Python*", Packt publication
6. Chun W., "*Core Python Programming*", Prentice Hall India

Accomplishments of the student after completing the course:

After completion of this course, students will be able to

- Gain awareness about various Python programming concepts, objects and libraries
- Solve challenging problems using Python programming language

***** ***** *****

Course Name: Introduction to Linux Programming

Course Code: MCA116(2)

Objectives:

Students will be able to

- Get a good understanding of Linux internals
- Develop proficiency in creating applications on Linux platform.
- Create a shell script for task automation

Prerequisites:

Fundamentals of Computers

Contents:

1. **Linux Environment**
   A brief history of Linux, architecture of Linux, features of Linux, introduction to vi editor
   Linux commands: PATH, man, echo, script, passwd, uname, who, date, stty, pwd, cd, mkdir, rmdir, ls, cp, mv, rm, cat, more, wc, lp, od, tar, gzip, file handling utilities, security by file permissions, process utilities, disk utilities, networking commands, unlink, du, df, mount, umount, find, unmask, ulimit, ps, w, finger, arp, ftp, telnet, rlogin
   Text Processing utilities and backup utilities: tail, head, sort, nl, uniq, grep, egrep, fgrep, cut, paste, join, tee, pg, comm, cmp, diff, tr, awk, cpio

2. **Introduction to Shell**
   Linux Session, Standard Streams, Redirection, Pipes, Tee Command, Command Execution, Command-Line Editing, Quotes, Command Substitution, Job Control, Aliases, Variables, Predefined Variables, Options, Shell/Environment, *Customization Filters*: Filters and Pipes, Concatenating files, Display Beginning and End of files, Cut and Paste, Sorting, Translating Characters, Files with Duplicate Lines, Count Characters, Words or Lines, Comparing Files

3. **Shell Scripting**
   Uses of shell scripting, input and output to shell scripts, execution of shell scripts, positional parameters, loops and control structures, arrays and functions, creating automated tasks

4. **Process and Memory Management**
   Linux processes and signals, POSIX threads: creation, synchronization, attributes, canceling, Semaphores, shared memory and Message queues, Inter-process communication, Creating and using shared memory, Linux tools to debugging and make

5. **Socket Programming**
   Socket, socket connections - socket attributes, socket addresses, socket, connect, bind, listen, accept, socket communications

References:

1. Neil Matthew, Richard Stones, "*Beginning Linux Programming*", 4th Edition, Wrox Publication (Wiley India)
2. Venkateshmurthy M.G., "*Introduction to Unix and Shell Programming*", Pearson India
3. Arnold Robbins, "*Linux Programming by example – The Fundamentals*", Pearson Education
4. Richard Stevens, "Advanced Unix *Programming*", Pearson Education
5. N.B. Venkateshwarlu, "*Linux Programming Tools*", B. S. Publication – Hyderabad
6. Eric S. Raymond, "*The Art of Unix Programing*", Pearson Education

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Develop console applications on Linux
- Tweak Linux for optimum performance
- Write scripts for recurring tasks

***** ***** *****

Course Name: Web Application Development

Course Code: MCA116(3)

Objectives:

Students will be introduced to

- The fundamentals of secure, dynamic web application development
- Implement a web application using one specific set of open sources server-side tools: PHP and MySQL

Prerequisites:

None

Contents:

1. **PHP Crash Course**
   Accessing PHP, Embedding PHP in HTML, Adding dynamic Content, Accessing Form Variables, Understanding Identifiers, Examining Variable Types, Declaring and Using Constants, Understanding Variable Scope, Using Operators, Using Variable Functions, Making Decision with Conditionals, Repeating Actions through Iteration, Using Declare

2. **Storing and Retrieving Data**
   Saving Data for Later, Processing Files, Opening a File, Writing to a File, Closing a File, Reading from File, Using other Useful File Functions, Locking Files

3. **Using Arrays**
   What is an array?, Numerically Indexed Arrays, Arrays with Different Indices, Array Operators, Multidimensional Arrays, Sorting Arrays, Sorting Multidimensional Arrays, Reordering Arrays, Loading Arrays from Files, Performing other array Manipulations

4. **String Manipulation and Regular Expressions**
   Format Strings, Joining and Splitting Strings with String Functions, Compare Strings, Matching and Replacing Substrings with String Functions, Regular Expressions

5. **Reusing Code and Writing Functions**
   The advantages of Reusing Code, Using require( ) and include( ), Using Functions in PHP, Defining your own functions, Passing by Reference and Passing By Value, Implementing Recursions

6. **Object-Oriented PHP**
   Understanding Object-Oriented Concepts, Creating Classes, Attributes and Operations in PHP, Implementing Inheritance in PHP, Understanding Advanced Object-Oriented Functionality in PHP

7. **Error and Exception Handling**
   Exception Handling Concepts, The Exception Class, User-Defined Exceptions, Exceptions and PHP's Other Error Handling Mechanism

8. Designing Your Web Database
Relation Database Concepts, Designing Your Web Database, Web Database Architecture

9. Creating Your Web Database
Using the MySQL Monitor, Logging in to MySQL, Creating Databases and Users, Setting Up Users and Privileges, Introduction to MySQL's Privilege System, Setting Up a user for the Web, Creating Database Tables, Understanding MySQL Identifiers, Choosing Column Datatypes

10. Working with Your MySQL Database
What is SQL, Inserting data into the Database, Retrieving Data from the Database, Updating Records in the Database, Altering Tables After Creation, Deleting Records from the Database, Dropping Tables, Dropping a Whole Database

11. Accessing Your MySQL Database from the Web with PHP
How Web Database architecture work?, Querying a Database from the Web, Putting new information in the Database, Using Prepared Statements, Using Other PHP-Database Interfaces

12. Interacting with the File System and the Server
Uploading Files, Using Directory Functions, Interacting with the File System, Using Program Execution Functions

13. Managing the Date and Time
Getting the Date and Time from PHP, Converting Between PHP and MySQL Date Formats, Calculating Dates in PHP, Calculating Dates in MySQL, Using Microseconds, Using the Calendar Functions

14. Using Session Control in PHP
Overview of Session Control, Understanding Basic Session Functionality, Implementing Simple Sessions, Creating a simple session, Configuring Session control, Implementing Authentication with Session Control

References:

1. Luke Welling, Laura Thomson, "*PHP and MySQL Web Development*", 4th Edition, Pearson
2. W. Jason Gilmore, "*Beginning PHP and MySQL 5 From Novice to Professional*", Apress,
3. Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass, "*Beginning PHP5, Apache, and MySQL Web Development*", Wrox
4. Robin Nixon, "*Learning PHP, MySQL, and JavaScript*", O'Reilly Media

Accomplishments of the student after completing the course:

After completion of this course, students will be able to

- Efficiently continue to expand their web development knowledge on their own with the solid foundation gained in the course

***** ***** *****

NEW SCHEME FOR MASTER OF COMPUTER APPLICATOINS (MCA) COURSE

SEMESTER - II of MCA

Applicable from December 2020 onwards

Sr. No.	Subject Code	Name of the Subject	TEACHING SCHEME					EXAMINATION SCHEME					
			THEORY	TUTORIAL	PRACTICAL	SESSIONAL		THEORY		PRACTICAL		TERM WORK	TOTAL MARKS
			HR.	HR.	HR.	Marks.	HR.	Marks.	HR.	Marks.	HR.		
1.	MCA121	Advanced Networking	3	1	-	25	2	50	3	-	-	25	100
2.	MCA122	Artificial Intelligence	4	-	-	25	2	50	3	-	-	25	100
3.	MCA123	Data Analytics	4	-	3	25	2	50	3	50	3	25	150
4.	MCA124	Java Programming	4	-	3	25	2	50	3	50	3	25	150
5.	MCA125	Operating Systems	3	1	3	25	2	50	3	50	3	25	150
6.	MCA126	Elective - 2 (Any one subject from track 1/2/3)	4	-	3	25	2	50	3	50	3	25	150
Total			**22**	**2**	**12**	**150**	**-**	**300**	**-**	**200**	**-**	**150**	**800**

Subjects for Elective - II

1. Machine Learning (Track-1)
2. Network Administration (Track-2)
3. Web Designing (Track-3)

Track No	Track-1	Track-2	Track-3
Track Name	Artificial Intelligence & Machine Learning	Networking	Web Technologies
Subject	MCA 126(1)	MCA 126(2)	MCA 126(3)

DEPARTMENT OF COMPUTER SCIENCE, GUJARAT UNIVERSITY

TEACHING SCHEME (WITH EFFECT FROM DECEMBER 2020)

M.C.A. (Master of Computer Applications)

Semester– II

Subject Code	Subject	Hours per week			Credits*
		Theory	Tutorial	Lab	
MCA121	Advanced Networking	3	1	-	4
MCA122	Artificial Intelligence	4	-	-	4
MCA123	Data Analytics	4	-	3	6
MCA124	Java Programming	4	-	3	6
MCA125	Operating Systems	3	1	3	6
MCA126	Elective - 2 (Any one subject from track 1/2/3)	4	-	3	6
Total		**22**	**2**	**12**	**32**

* Theory & Tutorial: - 1 hour = 1 credit
Practical: - 1 hour = 2/3 credit

Subjects for Elective - II

1. Machine Learning
2. Network Administration
3. Web Designing

Note: *Students are required to select a single track for the electives throughout the MCA program*

Course Name: Advanced Networking

Course Code: MCA121

Objectives:

The objective of the course is to enable students to

- Understand the need for dividing network functionalities into layers
- Understand the functionality of each layer of OSI and TCP/IP models
- Understand the concepts of data transfer and how different protocols implement these concepts
- Design network-based services using programming languages

Prerequisites:

Linux Operating Systems

Contents:

1. **Overview of Computer Networks**
   Introduction to Computer Networks and the Internet, The Network Edge, The Network Core, Delay, Loss and Throughput in Packet-Switched Networks, Protocol Layers and their Service Models, Networks under attack

2. **The Application Layer**
   Principles of Network Applications, The Web and HTTP, E-mail Services and its Protocols, DNS, FTP, SSH and Telnet, Peer-to-Peer Applications, Video Streaming & Content Distribution Networks, Overview of Socket Programming for designing Network based applications

3. **The Transport Layer**
   Transport Layer Services, Multiplexing and Demultiplexing, Connectionless Transport, Principles of Reliable Data Transfer, Connection Oriented Transfer, Principles of Congestion Control, TCP Congestion Control, Variants of TCP

4. **The Network Layer**
   Data Plane
   Overview of Network Layer, Functioning of Router, Routing mechanism, Internet Protocol (IPv4 and IPv6), Network Address Translation, Virtual Private Network (VPN) Overview of SDN
   Control Plane
   Routing Algorithms (LS & DV), Intra-AS routing in the Internet, SDN Control Plane, ICMP, Overview of Network Management and SNMP

5. **Link Layer and LANs**
   Introduction, Error-Detection and Error-Correction Techniques, Multiple Access Links & Protocols, Switched LANs, Link Virtualization

6. **Wireless Networks**
   Introduction, Wireless Links & Network Characteristics, WiFi 802.11 and Wireless LANs, Overview of Sensor Networks & IoT

7. **Socket Programming**
   The java.net package, classes related to IP addressing, classes related to URL, TCP Socket programming, UDP Socket Programming

References:

1. Kurose J., Ross K., "*Computer Networks: A Top-Down Approach*", 7th Edition, Pearson Education
2. Comer D., "*Internetworking with TCP/IP - Vol I*", Prentice Hall India
3. Hasan M. Jain R., "*High Performance TCP/IP Networking*", Prentice Hall India
4. Trivedi B., "*Computer Networks*", Oxford University Press - India
5. Tanenbaum A., Wetherall D., "*Computer Networks",* Pearson Education
6. Forouzan B., "*Data Communications and Networking*", Tata McGraw-Hill
7. Schiller J., "*Mobile Communications",* Pearson Education
8. Harold E., "*Java Network Programming*", 4th Edition, O'Reilly Media

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Implement appropriate client/server programming solutions for network-based services
- Appreciate the role of protocols and layered services in Computer Networking

***** ***** *****

Course Name: Artificial Intelligence

Course Code: MCA122

Objectives:

The objective of the course is to enable students to

- Introduce the necessary understanding of human intelligence and to explore the mechanisms that enables the intelligent thought and action
- Understand and learn effective ways for representing knowledge, applying intelligent problem-solving techniques & searching techniques

Prerequisites:

Basic knowledge of Mathematical Logic

Contents:

1. **AI Fundamentals**
   Defining Artificial Intelligence, AI: Early history, Overview of AI Application Areas, Defining AI techniques, Turing Test, Intelligent Agents

2. **State Space Search and Heuristic Search Techniques**
   Defining problems as State Space Search, Production Systems and Characteristics, Breadth First and Depth First Search, Hill Climbing Techniques, Problems and Solutions, Best First Search, Constraint Satisfaction Problems

3. **Knowledge Representation**
   Knowledge Representation Techniques: Semantic Nets, Partitioned Semantic Nets, Conceptual Dependency, Rules for Knowledge Representation, Advantages, Limitations

4. **Using Predicate Logic and Representing Knowledge as Rules**
   Representing simple facts in logic, Prepositional Logic, FOPL, Computable functions and predicates, Procedural vs. Declarative knowledge, Forward Vs. Backward Chaining

5. **Important Applications**
   Natural Language Processing: Steps of NLP, Applications, Importance and Limitations, *Neural Networks Overview:* Biological Neural Networks Vs. Artificial Neural Networks, Applications, Perceptron Learning Algorithm, SLP, MLP, Back Propagation Algorithm, *Game Playing:* Overview, Applications, Minimax Method, Alpha Beta Pruning, *Expert Systems:* Components, Characteristics, Advantages and Disadvantages, Real time examples of Rule-based expert system

References:

1. Elaine Rich, Kevin Knight, Shivashankar B. Nair, "*Artificial Intelligence*", 3rd Edition, McGraw Hill
2. Stuart Russell and Peter Norvig "*Artificial Intelligence-A modern Approach*", 2nd Edition, Pearson Education
3. George Lugar, "*Artificial Intelligence - Structures and Strategies for Complex Problem Solving"*, Pearson Education
4. Michael Negnevitsky, "*Artificial Intelligence - A guide to Intelligent Systems*", Pearson Education
5. N.P. Padhy, "*Artificial Intelligence and Intelligent Systems*", Oxford University Press

Accomplishments of the student after completing the course:

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

- Store and represent the knowledge in various applications and use different AI searching techniques.
- Deal with poorly defined or inexact problems that do not respond to the algorithmic solutions.

***** ***** *****

Course Name: Data Analytics

Course Code: MCA123

Objectives:

The objective of the course is to enable students to

- Understand key concepts of data, data mining, data analysis and data analytics
- Understand how data is created, stored, accessed
- Understand tools to effectively organize and visualize data
- Put the principles and methods of statistical analysis into practice using a range of real-world data sets

Prerequisites:

Knowledge of linear algebra, calculus, Python language and generic libraries

Contents:

1. **Introduction**
   Introduction to data, Data Analytics/Mining/Science, data sets, features, data scales, categorical and numerical data, cross-sectional and time series data. Univariate, bivariate and Multivariate data. Set and matrix representations, relations. Data Pre-processing: Handling missing values, normalization of data, dimensionality reduction, and outlier reduction. Introduction to Python libraries for data handling and statistics: Statistics library, SciPy, Pandas, NumPy

2. **Descriptive statistics : Graphical & Tabular methods**
   Summarizing data. Frequency Distribution Relative Frequency and Percent Frequency distributions, cross-tabulations. Bar Charts, Scatter plots, line chart, area chart and Pie Charts, stem and leaf display. Dot Plot, Histogram Cumulative Distributions, Ogive. Data summarization using pandas and data visualization using MatplotLib libraries of Python

3. **Descriptive statistics : Numerical methods**
   Measures of location: Mean median, mode, percentiles, quartiles,
   Measures of variability: range, interquartile range, variance, standard deviation, coefficient of variation,
   Measures of association between two variables: covariance interpretation of the covariance correlation coefficient interpretation of the correlation coefficient,
   Measures of distribution: shape, relative location, and detecting outliers z-scores empirical rule detecting outliers box plot, Handling descriptive statistics through python programs and/or libraries

4. **Regression models**
   Simple and Multiple linear regression model, least squares method, coefficient of determination model assumptions testing for significance, estimate of $\sigma 2$, t-test, and confidence interval. Implementing regression models using Sci-kit library

5. **Probability and Probability distribution**
   Counting rules, combinations, and permutations assigning probabilities, events and their probabilities, some basic relationships of probability complement of an event addition law, conditional probability independent events multiplication law. Random Variables, Discrete vs. Continuous random variable. Discrete probability distribution: Binomial, Poisson, Hyper geometric. Continuous probability distribution: Normal and standard normal distributions. Implementation of distributions using python libraries

6. **Hypothesis Testing**
   Focusing on Key Performance Indicators , Using KPI's for Web 2.0 , KPI's Setting Objectives and Key Results, Demonstrate how web analytics fits into the business structure

7. **Time Series and forecasting**
   Real World Tasks, Identifying and Optimizing Poorly Performing Pages , Search Engine Optimizing, Interpret the business benefits of web analytics

References:

1. Anderson, David R., et al, "*Statistics for business & economics*", Cengage learning
2. Thomas a. Runkler. – Wiesbaden, "*Data analytics: models and algorithms for intelligent data analysis*", Springer (cop. 2012), Verlag
3. Parag Kulkarni, Sarang Joshi,, Meta S. Brown, "*Big data analytics*"
4. Soraya Sedkaoui, "*Data analytics and big data*", Wiley
5. Emc Education Services, "*Data science and big data analytics*", Wiley

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Apply key concepts of Data Analytics
- Use Data Analytics tools to create, store and access the data and information
- Implement tools to organize and visualize the data effectively
- Perform methods of statistical analysis with a range of real-world data

***** ***** *****

Course Name: Java Programming

Course Code: MCA124

Objectives:

The objective of the course is to enable students to

- Understand the concepts of Object-Oriented Programming Language and easily use Java
- Get good understanding of developing multi-threaded applications using the Java Programming Language
- Harness the features of Java using APIs of Collection Framework, Lambda expressions and streams for effective programming

Prerequisites:

Basic knowledge of C/C++ programming language

Contents:

1. **Introduction**
   Recap of OOP features and relating them to Java, Features of the Java Language, Java Environment, Object Oriented Programming in Java, Java Program Structure, Java and Unicode, Writing first Java Application, command-line arguments, Use of System. Out, System.console().printf and System.console().readLine() methods

2. **Data Type, Variables and Constants, Loops and Logic**
   Data and Variables, Primitive Types, Reference Types, Difference from C++ in usage of Reference types, Arithmetic Calculations, Mixed Arithmetic expressions, Mathematical functions and constants, Bitwise operators, Enumerated data type, Boolean Variables Operator precedence, Program comments, Loops and Logic Making decisions, Logical operators, The Conditional Operator, the new operators and statements, The switch Statement, Variable scope, Loops, Assertions

3. **Defining Classes**
   Significance of class, Defining classes, Various members which go in a class definition – instance variable, class variable, methods, static methods, constructors, initializer block and class initializer block, method and constructor overloading, Using Objects, Use of this keyword, Nested classes, The finalize() method

4. **Extending classes and Inheritance**
   Recap of Inheritance, Use of keywords extends and super, “is-a” relationship for inheritance, Packages and access specifiers, package keyword, import and import static keywords, private, public and protected keywords, Abstract classes, Abstract methods, Multiple Inheritance using Interfaces, Static and Default methods in Interfaces, Functional Interfaces and Syntax of Lambda Expression

5. **Generics in Java and Commonly used Classes**
   Generic classes, Interfaces and Methods, Concept of type parameter, Bounded type parameter, using unbounded and bounded wild-cards, Commonly used classes like String, StringBuilder, Math, Wrapper classes, BigInteger, BigDecimal

6. **Comparators and Lambda Expressions**
   Comparison using Comparable and Comparator interfaces, Using methods and constructor reference for lambda expressions with single method invocation, Arrays class from java.util package, sorting arrays, Using Comparators, Functional interfaces from java.util.function package, Revisiting Comparator and using new static and default methods for creating Comparator

7. **Exceptions**
   Understanding Exceptions, Types of Exceptions, throw and throws keywords, checked and unchecked exceptions, handling exceptions using try-catch and finally blocks, multi-catch blocks, Creating custom exceptions

8. **Collection Framework, java.time package, java.util package**
   The Collection Framework from java.util package, Collection, Iterable, List, Set, Map, Queue, Dequeue, utility methods from Collections class, java.time package, classes, LocalDate, LocalTime, LocalDateTime, OffsetDate, ZonedDateTime, Duration, Period, Clock and Instant classes, The IntSummaryStatistics, LongSummaryStatistics, DoubleSummaryStatistics from java.util package, understanding the Optional class

9. **Stream API from java.util.stream package**
   Stream, IntStream, LongStream, DoubleStream, Understanding the intermediate and terminal operations on streams, reduce and collect methods, understanding the Collector, creating useful Collector using methods from the Collectors class

10. **The java.io package**
    File class, IO Stream classes for reading and writing to files, RandomAccessFile class

11. **Threads**
    Understanding Threads, the Thread class, creating thread of executing using Runnable and using sub-class of Thread, various properties of Thread, using synchronized keyword, using wait and notify

References:

1. Jain Pravin, "*The class of Java*", Pearson India
2. Horton I., "*Beginning Java*", 7th Edition, Wrox India
3. Cay S Horstmann, Gary Cornell, "*Core Java, Volume 1 – Fundamentals*", Pearson Education
4. Ken Arnold, James Gosling, David Holmes, "*The Java Programming Language*", Addison-Wesley Professional, Pearson Education
5. Subramian, "*Functional Programming in Java*", VPragmatic Bookshelf
6. Urma R., Fusco M., Mycroft A., "*Java 8 in action*", Manning Publications
7. Harold E., "*Java Network Programming*", 4th Edition, O'Reilly Media

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Create appropriate classes using the Java Programming Language to solve a problem using Object Oriented Approach and use concepts of Functional programming
- Develop multi-threaded applications using the Java Programming Language
- Use data structures effectively to design applications in Java Programming Language

***** ***** *****

Course Name: Operating Systems

Course Code: MCA125

Objectives:

The objective of the course is to enable students to

- Understand the functionalities and internals of operating systems
- Optimize their programs to execute on any operating system
- Able to implement tasks using Linux shell scripting

Prerequisites:

Fundamentals of Computers

Contents:

1. **Overview of Operating Systems**
   Role of Operating Systems (OS), Operations of OS, Resource Management, Security and Protection, Virtualization, Distributed Systems, Kernel Data Structures, Computing Environments, Free and Open-Source OS, OS Services, User and OS Interface, System Calls, System Services, OS Specific applications, OS Design and Implementation, OS Structure, Building and Booting an OS, Overview of Containers and Application Containerization

2. **Process Management**
   Process Concept, Process Scheduling, Operations on Process, Interprocess Communication, IPC in Shared-Memory Systems and Message-Passing Systems, Examples of IPC Systems, Communication in Client-Server Systems, Threads & Concurrency, Multicore Programming, Multithreading Models, Thread Libraries, Implicit Threading, Threading Issues, Concepts of CPU Scheduling, Scheduling Criteria, Scheduling Algorithms, Thread Scheduling, Multiprocessor Scheduling, Overview of Real-Time Scheduling, Algorithm Evaluation

3. **Process Synchronization**
   Overview of Synchronization Tools, The Critical-Section Problem, Peterson's Solution, Hardware Support for Synchronization, Mutex Locks, Semaphores, Monitors, Liveness, Evaluation, Classical Problems of Synchronization, POSIX and Java Synchronization, Overview of Deadlocks, Deadlock in Multithreaded Applications, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock

4. **Memory Management**
   Background, Contiguous Memory Allocation, Paging, Structure of the Page Table, Swapping, Virtual Memory Overview, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Memory Compression, Allocating Kernel Memory, Considerations for Virtual Memory Management

5. **Storage Management**
   Overview of Mass Storage Structure, HDD Scheduling, NVM Scheduling, Error Detection and Correction, Storage Device Management, Swap-Space Management, Storage Attachment, RAID Structure, Overview of I/O Systems, Application I/O Interface, Streams, Performance of I/O

6. **File System**
   Overview of File-System Interface, File-System Structure, File-System Operations, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, File Systems, File-System Mounting, Partitions and Mounting, File Sharing, Virtual File Systems, Remote File Systems, NFS

7. **Linux Shell Scripting**
   Basic commands of Linux, Process related commands of Linux, Linux APIs for process management and IPC, File related commands of Linux, Linux Shell Scripting to automate tasks, Processing files using Shell Scripts

References:

1. Silberschatz, A., Peter B. Galvin and Greg Gagne, "*Operating System Concepts*", 10th Edition, Wiley International Edition
2. Venkateshmurthy, "*Introduction to Unix and Shell Programming*", Pearson Education (India)
3. Stallings W, "*Operating Systems*",7th Edition, Pearson Education
4. Bach M J, "*The Design of Unix Operating System*", Prentice Hall India
5. Flynn I. M, "*Understanding Operating Systems*", Cengage India Publication
6. Tanenbaum A.S., "*Modern Operating Systems*",4th Edition, Pearson Education India
7. Eric Foster, John Welch, Micah Anderson, "*Beginning Shell Scripting*", Wrox Publication
8. Sibsankar Haldar, Alex Aravind, "*Operating Systems*", Pearson Education

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Apply practical knowledge of Linux System and shell scripting
- Harness the facilities provided by Operating System in application development
- Optimize the programs for specific Operating System

***** ***** *****

Course Name: Machine Learning

Course Code: MCA126(1)

Objectives:

The objective of the course is to enable students to

- Learn the concept of learning patterns from data and develop a strong theoretical foundation for understanding of state of the art Machine Learning algorithms
- Identify, formulate and solve machine learning problems that arise in practical applications

Prerequisites:

Knowledge of linear algebra, calculus, Python language and generic libraries

Contents:

1. **Overview of Machine Learning**
   Introduction to Machine learning from data, *Types of Machine Learning*: Supervised, Unsupervised, Reinforcement, concepts of regression, classification, clustering

2. **Linear Regression**
   Scatter diagram, Model representation for single variable, Single variable Cost Function, Least Square line fit, Normal Equations, Gradient Descent method for Linear Regression, Assumptions in linear regression, properties of regression line, Model Performance through R2, Multivariable model representation, Multivariable cost function, multiple linear regression, Normal Equations and non-invertibility, Gradient Descent method for multiple linear regression , Overfitting, Underfitting, Bias and variance, Regularization

3. **Logistic Regression**
   Issues of using Linear Regression in Classification, Sigmoid function, odds of an event, Logit function, Decision Boundary, Maximum Likelihood function, Linear regression verses Logistic Regression, Cost function, Multi-classification , confusion matrix, statistical measures to measure binary classification : Recall, sensitivity, specificity, precision, accuracy, pros and cons of logistic regression

4. **Supervised Learning**
   Classification problems, decision boundaries, K nearest neighbour methods, Linear classifiers, Bayes 'Rule and Naive Bayes Model, SVM - Introduction, Support Vectors & Margin, Optimization Objective, Linear & Non-Linear SVM, Hard Margin & Soft Margin in, Large Margin Classifiers, Kernels, SVM practical considerations, *Ensemble methods for classification and regression*: Bagging, Random Forests, Boosting, Decision Tree

5. **Unsupervised learning**
   Cluster Analysis, Classification and Clustering, Definition of Clusters, Clustering Applications, Distance measures, Proximity Measures for Discrete Variables, Proximity Measures for Mixed Variables, Partitional Clustering, Clustering Criteria, K-Means Algorithm, Fuzzy Clustering , Hierarchical Clustering, Agglomerative Hierarchical Clustering, Divisive Hierarchical Clustering, Cluster Validity, External Criteria, Internal Criteria

References:

1. Richert & Coelho, "*Building Machine Learning Systems with Python*", Packt Publishing Ltd.
2. Joel Grus, "*Data Science from Scratch*", O'Reilly Publications
3. Stephen Marsland , "*MACHINE LEARNING: An Algorithmic Perspective*", CRC Press
4. Rui Xu & Donald C. Wunsch II , "*Clustering*", IEEE Press
5. Tom M. Mitchell, "*Machine Learning*", McGraw-Hill publications
6. K.p. Soman, R. Loganathan, v. Ajay, "*Machine Learning with SVM and other Kernel methods*", PHI Learning
7. Ethem Alpaydın, "*Introduction to Machine Learning*", The MIT Press

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Develop an appreciation for what is involved in learning models from data
- Understand a wide variety of learning algorithms
- Understand how to evaluate models generated from data
- Understand and develop application involving computer vision and Natural Language Processing

***** ***** *****

Course Name: Network Administration

Course Code: MCA126(2)

Objectives:

The objective of the course is to enable students to

- Study various standards and technologies pertaining to Networking Devices
- Understand small, switched network implementation and diagnostics
- Implement an IP addressing scheme and IP services to meet network requirements
- Implement a small, routed network
- Implement and verify WAN Links
- Perform system administration tasks on Linux Server

Prerequisites:

Fundamentals of Networking

Contents:

1. **Linux System Administration**
   Introduction: Overview of Linux Distros, Installation & Package Management, File System Management (Partitioning, LVM, RAID), *Network Related Configurations*: Network Interfaces, Network Management & Configurations, Configuring DHCP, Linux Permissions, Controlling Access to Services, Runlevels, Authentication Configuration, Open SSH, DNS Services, HTTP Services, FTP Services, Mail Services (Postfix), Proxy Server, *System Configurations*: Boot Sequence, Configuring Environment Settings, Process Management , Automated Tasks, System logging & Log Files, Archiving

2. **Building Local Area Networks (LANs)**
   Functions of a Network, IP Network Addressing, Packet Delivery Process, Overview of Ethernet, Connecting to an Ethernet LAN, Challenges of a Shared LAN, Packet Delivery Process, Cisco IOS Software, starting a Switch, Maximizing the Benefits of Switching, Overview of Wireless LANs, Implementing a WLAN, WLAN Security

3. **LAN Connection**
   Functions of Routing, Constructing a Network Addressing Scheme, starting a Router, Configuring routers, Packet Delivery Process, Router Security, Accessing Remote Devices

4. **WAN Connection**
   Overview of WAN Technologies, Enabling the Internet Connection, Enabling Static Routing, Configuring Serial Encapsulation, Enabling RIP

5. **Medium-Sized Switched Network Construction**
   Overview of VLANs and Trunks, Implementing VLANs, Routing between VLANs, Troubleshooting Switched Networks

6. **Medium-Sized Routed Network Construction**
   Overview of Dynamic Routing, Implementing Variable-Length Subnet Masks

7. **Scaling the Network**
   Managing Address Spaces with NAT, Transitioning into IPv6, Overview of VPN Solutions, Establishing a Point-to-Point WAN Connection with PPP

References:

1. Colligs T., Wall K., "*Red Hat Linux Networking & System Administration*", Wiley India
2. Cox K., "*Red Hat Linux Administrators Guide*", Prentice Hall India
3. Ambawade D., Shah D., "*Linux Lab: Hands on Linux*", Wiley India
4. McQuerry Steve, "*Interconnecting Cisco Network Devices – Part I*", Cisco Press - Pearson Education
5. McQuerry Steve, "*Interconnecting Cisco Network Devices – Part 2*", Cisco Press - Pearson Education
6. Odom W., "*CCENT/CCNA ICND1 Official Exam Certification Guide*", Vol-I, Cisco Press – Pearson Education
7. Odom W., "*CCENT/CCNA ICND1 Official Exam Certification Guide*", Vol-II, Cisco Press – Pearson Education
8. Lammle T., "*CCENT : Cisco Certified Entry Networking Technician Guide*", John Wiley & Sons
9. Mueller S., "*Upgrading and Repairing PCs*", Pearson Education
10. Lammle T., "*Comptia Network+ Deluxe Study Guide*", Wiley India

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Ability to perform the tasks of Network Technician.
- Ability in Planning and Designing a Network Infrastructure.
- Ability to troubleshoot the networks

***** ***** *****

Course Name: Web Designing

Course Code: MCA126(3)

Objectives:

The objective of the course is to enable students to

- Design of web sites in terms of content organization, navigation, page and site design Measure website traffic and enhance business presence
- Understand the general principles of web usability

Prerequisites:

Knowledge of working with Computer System

Contents:

1. **HTML5**
   Creating Web Pages with HTML, Adding Style with CSS, Understanding the History of HTML, Working with HTML5, Using Good Coding Practices

2. **Web Site Design Principles**
   Understanding the Web Design Environment, Designing for Multiple Screen Resolutions, Crafting the Look and Feel of the Site, Creating a Unified Site Design, Designing for the User, Designing for Accessibility

3. **Web Site Planning**
   Understanding the Web Site Development Process, Creating a Site Specification, Identifying the Content Goal, Analyzing Your Audience, Building a Web Site Development Team, Creating Conventions for Filenames and URLs, Setting a Directory Structure, Using a Single Folder Structure, Creating a Site Storyboard, Publishing Your Web Site, Testing Your Web Site

4. **Cascading Style Sheets**
   Recognizing the Benefits of Using CSS, Activity: Building a Basic Style Sheet, Using Inheritance to Write Simpler Style Rules, Examining Basic Selection Techniques, Activity: Applying Basic Selection Techniques, Using class and id Selectors, Using the and Elements, Using Other Selectors

5. **Web Typography**
   Understanding Type Design Principles, Understanding CSS Measurement Units, Using the CSS Font Properties, Using the CSS Text Properties, Currently Unsupported CSS3 Properties, Customizing Bulleted and Numbered Lists

6. **Box Properties**
   Understanding the CSS Visual Formatting Model, Using the CSS Box Model, Applying the Margin Properties, Applying the Padding Properties, Applying the Border Properties, Using the Page Layout Box Properties

7. **Page Layouts**
   Understanding the Normal Flow of Elements, Creating Content Containers, Creating Floating Layouts, Building a Flexible Page Layout, Building a Fixed Page Layout

8. **Graphics And Color**
   Understanding Graphics File Formats, Choosing a Graphics Tool, Using the Image Element, Controlling Image Properties with CSS, Creating Web Site Color Schemes, Controlling Color Properties with CSS, Controlling Background Images with CSS

9. **Site Navigation**
   Creating Usable Navigation, Designing Navigation for Mobile Devices, Using Graphics for Navigation and Linking, Using Lists for Navigation, Building Horizontal Navigation Bars, Building Vertical Navigation Bars, Using Background Color and Graphics to enhance Navigation, Creating Hover Rollovers

10. **Data Tables**
    Using Table Elements, Using Table Headers and Footers, Styling Table Borders, Applying Padding, Margins, and Floats to Tables, Styling Table Background Colors

11. **Web Forms**
    Understanding How Forms Work, Using the <form> Element to Create Forms, Creating Input Objects, Styling Forms with CSS

12. **Responsive Web Design**
    Recognizing the Need for Responsive Web Design, Using Media Queries to Apply Conditional Styles, Creating Flexible Responsive Layouts, Creating Responsive Navigation Schemes, Using Responsive Images, Building a Responsive Design

References:

1. Joel Sklar, "*Principles of Web Design*", Cengage Learning
2. Daniel D. McCracken , Rosalee J. Wolfe , "*User-Centered Web Site Development: A Human-Computer Interaction Approach*", Prentice Hall
3. Faulkner X, "*Usability Engineering*", Houndsmills Palgrave
4. Pearrow M, "*Web Site Usability Handbook*", Rockland: Charles River Media
5. Thomas A. Powell, "*Web Design – The Complete Reference*", McGraw Hill
6. Wendy Willard, "*Web Design – A Beginner's Guide*", McGrawHill,

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Define a web site mission, target user population and collect the user requirements for a web site
- Conceptually design an appropriate page and site design
- Write the code to implement a web site and perform usability testing on a web site
- Implement and manage a web site successfully

***** ***** *****

NEW SCHEME FOR MASTER OF COMPUTER APPLICATOINS (MCA) COURSE

SEMESTER - III of MCA

Applicable from July 2021 onwards

Sr. No.	Subject Code	Name of the Subject	TEACHING SCHEME					EXAMINATION SCHEME					
			THEORY	TUTORIAL	PRACTICAL	SESSIONAL		THEORY		PRACTICAL		TERM WORK	TOTAL MARKS
			HR.	HR.	HR.	Marks.	HR.	Marks.	HR.	Marks.	HR.		
1.	MCA211	Cloud Computing	3	1	-	25	2	50	3	-	-	25	100
2.	MCA212	Object Oriented Software Engineering	4	-	-	25	2	50	3	-	-	25	100
3.	MCA213	Mobile Application Development	4	-	3	25	2	50	3	50	3	25	150
4.	MCA214	Enterprise Java Technologies	4	-	3	25	2	50	3	50	3	25	150
5.	MCA215	Elective – 3 (Any one subject from track 1/2/3)	3	1	3	25	2	50	3	50	3	25	150
6.	MCA216	Elective – 4 (Any one subject from track 1/2/3)	4	-	3	25	2	50	3	50	3	25	150
Total			**22**	**2**	**12**	**150**	**-**	**300**	**-**	**200**	**-**	**150**	**800**

Subjects for Elective - 3

1. Deep Learning (Track-1)
2. Network Security (Track-2)
3. Full Stack Web Development (Track-3)

Subjects for Elective - 4

1. Computer Vision / Natural Language Processing (Track-1)
2. Blockchain Technology / Network Analysis and Design (Track-2)
3. Web Security / Web Analytics (Track-3)

Track No	Track-1	Track-2	Track-3
Track Name	Artificial Intelligence & Machine Learning	Networking	Web Technologies
Subject (Elective-3)	MCA 215(1)	MCA 215(2)	MCA 215(3)
Subject (Elective-4)	MCA 216(1A) or MCA 216(1B)	MCA 216(2A) or MCA 216(2B)	MCA 216(3A) or MCA 216(3B)

Note: Students are required to select a single track for the electives throughout the MCA programme

A student can opt for either MCA216(1A) or MCA216(1B) for track-1, similarly student can opt for MCA216(2A) or MCA216(2B) for track-2 and MCA216(3A) or MCA216(3B) for track-3

DEPARTMENT OF COMPUTER SCIENCE, GUJARAT UNIVERSITY

TEACHING SCHEME (WITH EFFECT FROM JULY 2020)

M.C.A. (Master of Computer Applications)

Semester– III

Subject Code	Subject	Hours per week			Credits*
		Theory	Tutorial	Lab	
MCA211	Cloud Computing	3	1	-	4
MCA212	Object Oriented Software Engineering	4	-	-	4
MCA213	Mobile Application Development	4	-	3	6
MCA214	Enterprise Java Technologies	4	-	3	6
MCA215	Elective – 3 (Any one subject from track 1/2/3)	3	1	3	6
MCA216	Elective – 4 (Any one subject from track 1/2/3)	4	-	3	6
Total		**22**	**2**	**12**	**32**

* Theory & Tutorial: - 1 hour = 1 credit
Practical: - 1 hour = 2/3 credit

Subjects for Elective - 3

1. Deep Learning (Track-1)
2. Network Security (Track-2)
3. Full Stack Web Development (Track-3)

Subjects for Elective - 4

1. Computer Vision / Natural Language Processing (Track-1)
2. Blockchain Technology / Network Analysis and Design (Track-2)
3. Web Security / Web Analytics (Track-3)

Note: Students are required to select a single track for the electives throughout the MCA programe

A student can opt for either MCA216(1A) or MCA216(1B) for track-1, similarly student can opt for MCA216(2A) or MCA216(2B) for track-2 and MCA216(3A) or MCA216(3B) for track-3

Course Name: Cloud Computing

Course Code: MCA211

Objectives:

The objective of the course is to enable students to

- Articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing
- Implement business solutions over cloud computing platform
- Harness the cloud infrastructure to provide efficient software-based solutions
- To understand the service model with reference to cloud computing

Prerequisites:

Basic Knowledge of Operating Systems and Computer Networking, Knowledge of Database Systems

Contents:

1. **Cloud Computing Fundamentals**
   Evolution and enabling technologies for cloud computing, Motivation for cloud computing, Benefits & challenges, Defining cloud computing, Principles and challenges, Cloud computing model, Cloud computing services

2. **Virtualization, Scaling & Capacity Planning**
   Resource Virtualization, Approaches to virtualization, Hypervisors, Resource pooling, Resource Sharing and Provisioning, Scaling in the cloud, Capacity Planning and Load Balancing

3. **Technologies over Cloud**
   The File system and Storage services, Database technologies, Networking technologies over cloud & CDN, Cloud Management and Programming Model, Security reference model, Security aspects, Platform related security, Audit and Compliance, Portability and Interoperability issues

4. **Open-Source support for cloud**
   Open-source tools for IaaS, Open source tools for PaaS, Open source tools for SaaS, Open source tools for research, Distributed Computing Tools for Management of Distributed Systems

5. **Software development in the cloud**
   Introduction, Different perspectives on SaaS development, Challenges in cloud environment for software development, Launching a virtual machine, Creation of an instance, Connecting to an instance, Security services & roles, Storage Services, Database services and launching the database instance, Creating a static website, Deployment of code, Harnessing Serverless Architecture, Container Services, Notification Services, Overview of Microservices

References:

1. Bhowmik S., "*Cloud Computing*", Cambridge University Press
2. Sosinsky B., "*Cloud Computing Bible*", Wiley India
3. Buyya R., Broberg J., Goscinski A., "*Cloud Computing : Principles and Paradigm*", John Wiley & Sons
4. Baron J., et al., "*AWS Certified Solutions Architect Official Study Guide: Associate Exam"*, Sybex
5. Hunter T., Porter S., "*Google Cloud Platform for Developers: Build highly scalable cloud solutions with the power of Google Cloud Platform*", Packt Publishing Limited
6. Chandrasekaran K., "*Essentials of Cloud Computing*", CRC Press
7. Marinescu D., "*Cloud Computing - Theory and Practice*", Morgan Kauffman
8. Kant K., Doshi R., "*Cloud Computing*", BPB Publications

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Ability to decide and implement appropriate cloud computing infrastructure for any organization
- Ability to design customized programming solutions using cloud computing model

***** ***** *****

Course Name: Object Oriented Software Engineering

Course Code: MCA212

Objectives:

The objective of the course is to enable students to

- Understand the concept and importance of Software Engineering
- Understand the concept of object oriented paradigm
- Understand the UML diagrams which are used at different stages of Software development life cycle
- Understand all the activities under project management

Prerequisites:

Basic knowledge of Programming, Systems Analysis and Design

Contents:

1. **Introduction to Software Engineering and Modeling through UML**
   Software Engineering Concepts, Object Oriented Concepts, OO methodologies, OO modeling, definitions and terminologies, Software development life cycle models (Prescriptive and Specialized Process Models), Agile Development methodologies, Object Oriented Analysis, classes, Objects, Relationships, state and Behavior, Introduction to UML

2. **Requirement Engineering and object-oriented software estimation**
   Functional and non-functional requirements, requirement elicitation techniques, Software requirement specification, Requirements change management, Need of Cost and Schedule estimation, Lorenz and Kidd estimation method, Use case Points method, class Point method, Object Oriented Function point, Risk Management

3. **Object Oriented Software Design and development**
   Interaction diagrams, Sequence diagrams, Collaboration diagrams, Class diagrams, State Chart Diagrams, Reuse concepts, Software Architecture, Architectural Genres, Architectural Styles, Architectural Design, Interface Design, Content Design, Navigation Design, Object Oriented Hypermedia design method

4. **Software Quality and Metrics**
   Software quality attributes, Software Quality Models, Measurements Basics, Analysis of metric data, metrics for measuring size and structure, measuring software quality, object oriented Metrics. Cost Impact of Software Defects, Defect Amplification and Removal, Elements of Software Quality Assurance, SQA Tasks, Goals and Metrics, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, ISO - 9000 Quality Standards, SQA Plan

5. Software Testing

Verification and Validation, Software verification techniques, testing strategies, functional testing, Structural Testing, Class testing, state based testing, Mutation testing, levels of testing, Testing OOA and OOD Models, Object oriented Testing Strategies, Object oriented Testing Methods, Testing Methods Applicable at Class Level, Interclass Test-Case Design, software testing tools

6. Software Configuration Management, Risk management and maintenance

Software Configuration Management, SCM Repository, SCM Process, types of software risks, risk identification, projection and refinement, the RMMM plan, project scheduling, TimeLine Chart, categories and challenges of software maintenance

References:

1. Roger Pressman, "*Software Engineering – A Practitioner's Approach*", 7th Edition, McGraw Hill Higher Education
2. Yogesh singh, Ruchika Malhotra, "*Object-Oriented Software Engineering*"
3. Bernd Bruegge, Allen Dutoit, "*Object-Oriented Software Engineering Using UML, Patterns, and Java*", Pearson New International Edition
4. Satzinger, Jackson, Burd, "*Object-Oriented Analysis and Design with the Unified Process*", Cengage learning
5. A Sommerville, "*Software Engineering*", Pearson Education
6. W S Jawadekar, "*Software Engineering – Principles and Practices*", TMH Publication

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Understand and appreciate the importance of Software Engineering and its object oriented approach in today's world
- Understand and perform the various activities required to develop good quality software within time and cost budget
- Estimate the expected cost and risks

***** ***** *****

Course Name: Mobile Application Development

Course Code: MCA213

Objectives:

The objective of the course is to enable students to

- Understand the architecture and components of Android SDK
- Understand use of Tools / Technologies like ANDROID Studio (Latest Version), ANDROID Version (Jelly Bean and later) for application development
- Create mobile applications on the Android Platform with advanced functionalities
- Understand process of monetizing and publishing applications

Prerequisites:

Knowledge of the Core Java Programming, Database concepts

Contents:

1. **An Overview of the Android platform**
   Introducing android, setting up your android development environment, writing your first android application, android application basics, understanding the anatomy of an android application, defining your application using the android manifest file, managing application resources, android's underlying architecture, security and permissions, setting up your android development environment, configuring your development environment, configuring your operating system for device debugging, configuring your android hardware for debugging, exploring the android SDK

2. **Understanding the anatomy of an Android application**
   Mastering important android terminology, performing application tasks with activities, organizing activity components with fragments, managing activity transitions with intents, working with services, receiving and broadcasting intents, Configuring android applications using the android manifest file, managing your application's identity, enforcing application system requirements, registering activities in the android manifest, working with permissions, exploring other manifest file settings

3. **Managing application resources**
   Creating resources, simple values, styles and themes, drawables, layouts, animations, menus, using resources, using resources in code, referencing resources within resources, using system resources, referring to styles in the current theme, creating resources for different languages and hardware, runtime configuration changes, The android application lifecycle, understanding an application's priority and its process' states, introducing the android application class

4. Android user interface fundamentals

Assigning user interfaces to activities, introducing layouts, introducing fragments, the android widget toolbox, creating new views, introducing adapters, working with dialogs and dialog fragments

5. Drawing and animations

Working with animations, tweened view animations, creating and using frame-by-frame animations, advanced canvas drawing, introducing the surface view, creating interactive controls, advanced drawable resources, copy, paste, and the clipboard

6. Working with files and directories

Exploring with the android application directories, working with other directories and files on the android file system

7. Databases and content providers

Introducing android databases, sqlite databases and content providers. Introducing sqlite content values and cursors, working with sqlite databases, exploring android's content providers, modifying content provider data, using third-party content providers

8. Working in background

Introducing services, using background threads, using alarms enhancing user experience introduction and addition of action bar, menus and dialogs

9. Hardware sensors

Using sensors and the sensor manager, monitoring a device's movement and orientation, introducing the environmental sensors, maps, geocoding, and location-based services, using location-based services, using the emulator with location-based services, using proximity alerts, creating map-based activities

10. Multimedia and wireless networking and telephony

Playing audio and video, manipulating raw audio, creating a sound pool, using audio effect, using the camera for taking pictures, recording video, using Bluetooth managing network and internet connectivity, managing wi-fi transferring data using wi-fi direct, near field communication, hardware support for telephony, introducing sms and mms

11. Monetizing, promoting, and distributing applications

Signing and Publishing Applications, Distributing Applications, An Introduction to Monetizing Your Applications, Application Marketing, Promotion, and Distribution Strategies, Analytics and Referral Tracking

References:

1. Reto Meier, "*Professional ANDROID Application Development*", Wrox Publication
2. Lauren Darcey and Shane Conder, "*Android Wireless Application Development*", Pearson Education,

3. Mark L Murphy, "*Beginning Android*", Wiley India Pvt Ltd
4. Josh Skeen, David Greenhalgh , "*Android Programming: The Big Nerd Ranch Guide*, Big Nerd Ranch Guides
5. John Horton, "*Android Programming for beginners"*, Packt Publication

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Apply key concepts of Android Programming
- Develop GUI and animated android applications
- Harness the mobile sensors through programming
- Deployment of Android Applications

***** ***** *****

Course Name: Enterprise Java Technologies

Course Code: MCA214

Objectives:

The objective of the course is to enable students to

- Learn the concepts of web technologies and apply it in real life applications
- Understand the concept of Web Services and APIs

Prerequisites:

HTML, DHTML and Object-Oriented programming

Contents:

1. **Introduction to JDBC and Web Services**
   The Design of JDBC, JDBC Configuration, Executing SQL Statements, Query Execution, Fundamentals of Web Services

2. **Web Application Components**
   Understanding web applications, Introducing the MVC design pattern, JAR file, WAR file, HTTP, GET request method , POST request method, GET and POST in HTML processing, other Request methods, The HTTP Response, Using Deployment Descriptors, Overview of REST API

3. **Servlets**
   Introducing Servlet and MVC Pattern, Introducing the javax.servlet Package, Introducing Servlet Interface, Introducing HTTP and Servlets, Understanding the Request/Response cycle, Understanding the Deployment Descriptor, Introducing the ServletContext Lifecycle classes, ReuestDispatcher interface

4. **JSP**
   Overview of JSP Technology, Invoking Java Code with JSP Scripting elements, controlling the structure of generated servlets: The JSP Page directive, Including files and Applets in JSP Pages, Using JavaBeans Components in JSP Documents, Integrating Servlets and JSP: The MVC Architecture, Simplifying access to Java Code: The JSP 2.0 Expression Language

5. **Advance JSP: Using JSTL**
   Tag Libraries: The Basics, JSP Standard Tag Library (JSTL)

References:

1. Marty Hall, Larry Brown, "*Core Servlets and JavaServer Pages Volume - 1*", 2nd edition, Pearson Education
2. Marty Hall, Larry Brown, "*Core Servlets and JavaServer Pages Volume - 2*", 2nd edition, Pearson Education
3. Cay S Horstman, Gray Corenell, "*Core Java 2, Volume 2 – Advanced Features*", Pearson Education
4. Kogent Learning Solutions Inc., "*Web Technologies Black Book*", Dreamtech press
5. Sue Spielman, Meeraj Kunnumpurath , "*Pro J2EE 1.4 From Professional to Expert*", APress Publication
6. Servlets Documentation, https://docs.oracle.com/javaee/7/tutorial/servlets.htm

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Understand the concepts of web technologies
- Create Java application using JDBC
- Create a Web Application using JSP ,JSTL and Servlet
- Apply web technologies concepts to develop web applications

***** ***** *****

Course Name: Deep Learning

Course Code: MCA215(1)

Objectives:

The objective of the course is to enable students to

- Understand the latest algorithms and architectures of deep learning with practical viewpoint
- Understand the necessary background of the ongoing research and gain required implementation knowledge

Prerequisites:

Knowledge of linear algebra, calculus, Probability, Machine Learning Fundamentals, Python programming

Contents:

1. **Feed-forward Deep Networks**
   Review of Machine Learning Basics, Overview of Deep Networks, Bias and Variance, the curse of dimensionality, Vanilla MLP, Flow Graphs and Back propagation, Universal Approximation Theorem, Feature representation

2. **Convolutional Neural Networks**
   Concept of Convolution, Convolution Operation, Pooling, Stride, Convolution Modules, Efficient Convolution Algorithms, Random or Unsupervised features, Applications in Computer Vision

3. **Recurrent and Recursive Nets**
   Unfolding Flow graphs and parameter sharing, Recurrent Neural Networks, Bidirectional RNNs, Deep Recurrent Architecture, Recursive Neural Networks, Auto-Regressive Networks, Recurrent Vs. Recursive Neural Nets

4. **Regularization of Deep Models**
   Regularization from Bayesian Perspective, Parameter Norm Penalty, Regularization as Constrained Optimization, Under-Constrained Problems, Classical Regularization as Noise Robustness, Dropout, Multi-Task training, Adversarial Training

5. **Optimization for Training Deep Models**
   Optimization for Model Training, Challenges in Optimization, Basic Algorithms, Adaptive learning rates, Second order methods, Natural gradient methods, Global Optimization

6. Practical Implementations

Image Classification, Types of Image Classifiers, Building a deep neural architecture, Building feature set, Preparing Data for training, Adding Dropout, Understanding existing architectures such as Alex Net and Google LeNet, Using existing architectures in applications, Using inbuilt Tensor Flow functionality to build a Convolutional Neural Network

References:

1. Ian Goodfellow, Yoshua Bengio and Aaron Courville, "*Deep Learning,*", MIT Press
2. Jeff Heaton, "*Artificial Intelligence for Humans: Deep Learning and Neural Networks, Book 3*", 1st edition, Heaton Research, Inc.
3. Giancarlo Zaccone, "*Deep Learning with Tensor Flow*", Packt Publishing Limited

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Know the ongoing trends in Deep Learning
- Understand the architecture of various deep networks from the viewpoint of implementation
- Learn to design and develop various types of deep networks
- Understand how to implement and control the behavior of deep networks

***** ***** *****

Course Name: Network Security

Course Code: MCA215(2)

Objectives:

The objective of the course is to enable students to

- Understand the state-of-the-art in network security and computer security
- Study the security mechanisms at various layers of TCP/IP Model
- Develop strong analysis, testing and troubleshooting skills regarding to computer network security
- Implement various open source tools related to network security

Prerequisites:

Knowledge of basics of Computer Networking

Contents:

1. Network Security Fundamentals, Issues and Challenges

Computer Network Security Fundamentals: Introduction, Securing the Computer Network, Forms of Protection, Security Standards, *Security Threats and Threat Motives to Computer Networks*: Sources of Security Threats, Security Threat Motives, Security Threat Management, Security Threat Correlation, Security Threat Awareness, *Computer Network Vulnerabilities*: Definition, Sources of Vulnerabilities, Vulnerability Assessment, *Cybercrimes and Hackers*: Introduction, Cybercrimes, Hackers, Dealing with Cybercrimes, *Security Assessment, Analysis, and Assurance*: System Security Policy, Building a Security Policy, Security Requirements Specification, Threat Identification, Threat Analysis, Vulnerability Identification and Assessment, Security Certification, Security Monitoring and Auditing, Products and Services

2. Cryptography

Overview of Cryptography, Classical Algorithms, Symmetric Key Cryptography, Asymmetric Key Cryptography, Public Key Cryptography, Integrity Check Algorithms, Authentication Algorithms, Overview of Key Management

3. Access Control, Authorization and Authentication

Access Control and Authorization: Definitions, Access Rights, Access Control Systems, Authorization, Types of Authorization System, Authorization Principles, Authorization Granularity, Web Access and Authorization, *Authentication*: Definition, Multiple Factors and Effectiveness of Authentication, Types of Authentication, Authentication Methods, Developing an Authentication Policy

4. Firewalls, IDS and IPS

Firewalls: Definition, Types of Firewalls, Configuration and Implementation of a Firewall, The Demilitarized Zone (DMZ), Improving Security Through the Firewall, Firewall Forensics, Firewall Services and Limitations, *Intrusion Detection & Prevention* : Definition, Intrusion Detection, Intrusion Detection Systems (IDSs), Types of Intrusion Detection Systems, The Changing Nature of IDS Tools, Other Types of Intrusion Detection Systems, Response to System Intrusion, Challenges to Intrusion Detection Systems, Implementing an Intrusion Detection System, Intrusion Prevention Systems (IPSs), Intrusion Detection Tools

5. Dealing with Computer Network Security Challenges

Computer and Network Forensics: Introduction, Computer Forensics, Network Forensics, Forensic Tools, *Virus and Content Filtering*: Definitions, Scanning, Filtering, and Blocking, Virus Filtering, Content Filtering, Spam, *Standardization and Security Criteria*: Introduction, Product Standardization, Security Evaluations, Major Security, Evaluation Criteria, *Computer Network Security Protocols*: Introduction, Application-Level Security, Security in the Transport Layer, Security in the Network Layer, Security in the Physical Layer

6. Overview of Security Beyond Wired Networks

Overview of Security in Wireless Networks and Sensor Networks, Security in Cloud Environment, Security in Virtualization, Need of Security in IoT and Mobile Systems

Case Study of Open Source Tools

Nmap utility, Wireshark, iptables, OpenSSL, OSSEC, OpenVAS

References:

1. Kizza J.M., "*Guide to Computer Network Security*", 4th edition., Springer International Publishing
2. Kaufman C., Perlman R., Speciner M., "*Network Security: Private Communication in Public World*", Prentice Hall India
3. Perez A., "*Network Security*", John Wiley & Sons
4. Stallings W., "*Network Security Essentials – Applications and Standards*", Pearson India
5. Forouzan B., "*Cryptography and Network Security*", Tata McGraw Hill
6. Howlett T., "*Open Source Security Tools*", Prentice Hall

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Analyze and develop solutions to solve network security related problems
- Have thorough understanding of various standards and protocols related to computer network security
- Effectively use various tools and utilities for computer networking

***** ***** *****

Course Name: Full Stack Web Development

Course Code: MCA215(3)

Objectives:

The objective of the course is to enable students to

- Learn the concepts of front end and back end development
- Implement full-stack concepts in real-life applications

Prerequisites:

Knowledge of HTML, DHTML and Object-Oriented programming

Contents:

1. **Introduction to Java Script and JSON**
   Overview of Java Script, Primitives, Operations and Expressions, Screen Output and Keyboard Input, Control Statement, Object Creation and modification, Arrays, Functions, Constructors, Pattern Matching using regular expressions, Overview of JSON, Comparison of JSON with XML and CSV, Callback functions, Promise Objects, async…await, Overview of ES6

2. **AngularJS Core Concepts**
   Introduction to AngularJS, Advantages of Angular, AngularJS MVC, Introduction to SPA, Setting up the environment, First App using MVC architecture, Understanding attributes, Expression and Data Binding, Working with directives, Angular Modules, Controller, Scope and View, Create Controller and Module, $scope hierarchy, Introduction to Lifecycle hooks

3. **Filters, Forms and WebSocket**
   Filters - Built-in filters - upper case and lower case filters, date ,currency and number formatting ,orderBy, filter, custom filter, Angular JS Forms – Working with AngularJS forms, model binding, form controller ,Using CSS classes, form events, custom model, update triggers ,custom validation, $http service, Ajax implementation using $http, Web APIs, Overview of AJAX, Fundamentals of WebSocket

4. **Dependency Injection, Services ,Routing and Navigation**
   Dependency injection, Using dependency injection, Angular JS service – Understanding services, Using built-in service, Creating custom service, Injecting dependency in service, Overview of Routing, Routing using ngRoute and UI- Router, ngView Directive, Configuring $routeProvider, $stateProvider, Animating Angular App, Overview of TypeScript

5. **Node JS**
   Introduction to Node.js, Features of Node.js, Setup Development Environment- Installing Node.js, Working with REPL, Node.js Console, Node.js Module, Node Package Manager, Node.js Basics, File System, HTTP and HTTPs, Creating Web Server- Handling http request, Node.js Callbacks, Node.js Events

References:

1. Brad Dayley, "*Node.js, MongoDB, and AngularJS Web Development* "; 2nd edition, Addison-Wesley
2. Ivan Bayross, "*Web Enabled Commercial Application Development Using HTML,DHTML, PERL, Java Script*" ,Revised Edition, BPB publications
3. Adam Freeman, "*Pro Angular JS*", Apress edition, Wiley india
4. Agus Kurniawan, "*AngularJS Programming by Example*", PE Press
5. Amos Q. Haviv, "*Mean Web Development*", 2nd edition, Ingram short title
6. Mozilla Developers Network(MDN) Web Docs, https://developer.mozilla.org/en-US/
7. Angular Documentation, https://angular.io/docs
8. Nodejs Documentation, https://nodejs.org/en/docs/

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Design the frontend and backend application after learning full stack web development
- Apply the newly available framework for web designing

***** ***** *****

Course Name: Computer Vision

Course Code: MCA216(1A)

Objectives:

The objective of the course is to enable students to

- Learn basic principles of image formation
- Understand Image processing algorithms and different algorithms for 3D reconstruction and recognition from single or multiple images (video)
- Emphasis on the core vision tasks of scene understanding and recognition

Prerequisites:

Knowledge of Programming and Mathematical foundations

Contents:

1. **Computer Vision and Applications**
   What is Computer Vision, Overview of Computer Vision Applications, *Image Formation Models*: Monocular imaging system , *Radiosity*: The 'Physics' of Image Formation, Radiance, Irradiance, BRDF, color etc., Orthographic & Perspective Projection, Camera model and Camera, calibration, Binocular imaging systems, Multiple views geometry, Structure determination, shape from shading , Photometric Stereo, Depth, from Defocus , Construction of 3D model from images

2. **Image representation and pre-processing**
   Image representation, discrete and continuous, image enhancement, restoration, histogram equalization and matching , Shape Representation and Segmentation : Contour based representation, Region based representation, Deformable curves and surfaces , Snakes and active contours, Level set representations , Fourier and wavelet descriptors , Medial representations , Multiresolution analysis

3. **Feature Extraction**
   Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT

4. **Image Segmentation**
   Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection and recognition

5. **Motion Estimation**
   Motion analysis and Activity Recognition ,Motion detection and tracking, Inference of human activity from image sequences

6. **Applications of computer vision**
   Document Image, Analysis, Biometrics, Object Recognition, Tracking, Medical Image .Analysis, Content-Based Image Retrieval, Video Data Processing, Multimedia, Virtual Reality and Augmented Reality

References:

1. D. Forsyth, J. Ponce, "*Computer Vision - A modern approach*", Prentice Hall Robot Vision - B. K. P. Horn, McGraw-Hill
2. R. C. Gonzalez, R. E. Woods, "*Digital Image Processing*", Addison Wesley Longman Inc.
3. Richard Szeliski, "*Computer Vision: Algorithms and Applications (CVAA)*, Springer, 2010
4. Mark Nixon, Alberto S. Aquado, "*Feature Extraction & Image Processing for Computer Vision*", 3rd edition, Academic Press, 2012

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Apply Computer vision to 3D modeling, video analysis, video surveillance, object recognition and vision based control

***** ***** *****

Course Name: Natural Language Processing

Course Code: MCA216(1B)

Objectives:

The objective of the course is to enable students to

- Learn the key concepts pertaining to Linguistics and Natural Language Processing that are used to describe and analyze natural language
- Gain insights into statistical and semantic approaches to Natural Language Processing
- Understand basic principles of machine learning to natural language data
- Learn the use standard software packages for machine learning in the domain of Natural Language Processing
- To understand how data structures and algorithms are used in Natural Language Processing

Prerequisites:

Knowledge of linear algebra, Probability & Statistics, Linear Algebra, Python programming, AI & Machine Learning

Contents:

1. **Introduction to natural language processing**
   Structural features of texts in natural language, ambiguity on all levels of language, the main challenges of natural language processing, *basic approaches to problem solving*: manually written rules and machine learning

2. **Basic text processing and edit distance**
   Pre-processing: tokenization and segmentation, *normalization of words*: stemming, lemmatization, Bag of Words, TF-IDF, morphological analyzers, regular expressions, edit distance

3. **Language models**
   N-grams, perplexity, the use of language models: input prediction, error correction, speech recognition, text generation

4. **Tagging problems and hidden Markov models**
   PoS tagging; named entity recognition as a tagging problem; hidden Markov models, their advantages and disadvantages; the Viterbi algorithm

5. **Text classification and sentiment analysis**
   Classification problems; naive Bayes classifier; text classification; sentiment analysis

6. Parsing

Constituency and dependency trees; context-free grammar; probabilistic approach to parsing; lexicalized PCFGs; CKY algorithm

7. Computational semantics

Word senses and meanings; WordNet; semantic similarity measures: thesaurus-based and distributional methods

8. Vector space models of semantics

Word2vec and doc2vec, Word embedding, Character to Sentence Embedding

9. Text summarization

Extractive and abstractive summarization; multiple-document summarization; query-based summarization; supervised and unsupervised learning; evaluation of summarization systems; ROUGE

10. NLP with Sequence Models

Recurrent Neural Networks: LSTM; GRU, Neural Networks for sentiment analysis, Deep N-Grams

11. Practical

Working with Text: Tokenization, Token Normalization,
Stemming: Porter Stemmer,
Lemmatization: Wordnet lemmatizer, Bag of words (BoW), TF -IDF, Text Classification Model: Sentiment analysis
Parts-of-Speech Tagging:
Creating POS-tagged corpora, Selecting a machine learning algorithm, Statistical modelling involving the n-gram approach, Developing a chunker using pos-tagged corpora
Parsing:
Treebank construction, Extracting Context Free Grammar (CFG) rules from Treebank, Creating a probabilistic CFG, CYK chart parsing algorithm, Early chart parsing algorithm
Semantic Analysis:
Named Entity Recognition (NER) using Hidden Markov Model, NER using POS tagging, Disambiguating senses using Wordnet

References:

1. J. Perkins, "*Python 3 Text Processing with NLTK 3 Cookbook*", Packt Publishing Ltd.
2. C. D. Manning, P. Raghavan, H. Schütze, and others, "*Introduction to information retrieval, vol. 1*", Cambridge univers
3. Steven Bird, Ewan Klein, and Edward Loper, "*Natural language processing with Python: analyzing text with the natural language toolkit*", O'Reilly Media, Inc.
4. J. Pustejovsky and A. Stubbs, "*Natural language annotation for machine learning*", O'Reilly Media, Inc., 2012
5. U. S. Tiwary and T. Siddiqui, "*Natural language processing and information retrieval",* Oxford University Press, Inc., 2008

6. D. Chopra, N. Joshi, and I. Mathur, "*Mastering Natural Language Processing with Python*", Packt Publishing Ltd, 2016

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Develop formal models to express natural language phenomenon
- Utilize mathematical expressions and notations to describe algorithms for language processing, to implement NLP systems in a clean and structured manner,
- Design and Implement tools for NLP and to appreciate the use of Machine Learning techniques in the domain of NLP

***** ***** *****

Course Name: Blockchain Technology

Course Code: MCA216(2A)

Objectives:

The aim of this course is to enable students to

- Get the introduction of Blockchain technology and Cryptocurrency
- Integrate ideas from Blockchain technology into their own projects

Prerequisites:

Database Management System

Contents:

1. **Blockchain Fundamental**
   Overview of Cryptography, Introduction to Blockchain, Components, Concept of Block, Block Chain Types and Consensus Mechanism

2. **Cryptocurrency**
   Cryptocurrency Basics, Types of Cryptocurrency, Cryptocurrency Usage : Ecosystem Players, Cryptomining, Airdrop, Token and Coin Burning, Investing and Trading, Cryptocurrency Safety

3. **Public and Private Blockchain System**
   Introduction, Popular Public Blockchain, Bitcoin, Ethereum, Smart Contract, Characteristics of Private Blockchain, Private Blockchain and Open source

4. **Consortium Blockchain and ICO**
   Introduction, Characteristics, Hyperledger Platform, Fundraising Methods, Lunching ICO, Investing in ICO, ICO Pros and Cons, Evolution of ICO, ICO Platforms : Launching and Listing

5. **Security**
   Security Aspect in Bitcoin, Challenges, Performance and Scalability, Identity Management and Authentication, Regulatory Compliance and Assurance, Safeguarding Blockchain Smart Contract

6. **Application of Blockchain**
   Uses of Blockchain in Computing, Limitations and Challenges in Blockchain, Blockchain Platform using Python

References:

1. Chandramouli Subramanian, Asha A George, Abhilash K A and Meena Karthikeyan, "*Blockchain Technology*", Universities Press
2. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, "*Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*", Princeton University Press
3. Melanie Swan, "*Blockchain Blue print for Economy*", O'Reilly Media, Inc.
4. Daniel Drescher, "*Blockchain Basics: A Non-Technical Introduction in 25 Steps*", Apress
5. Imran Bashir , "*Mastering Blockchain*", 2nd Edition, Packt publishing, Mumbai

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Understand Blockchain mechanism
- Develop various applications of Blockchain

***** ***** *****

Course Name: Network Analysis and Design

Course Code: MCA216(2B)

Objectives:

The objective of the course is to enable students to

- Learn the principles of Network Design
- Understand the process of Network Deployment
- Gain basic insight of Network Operations and Maintenance

Prerequisites:

Knowledge of Fundamentals of Networking

Contents:

1. **Introduction**
   Overview of Analysis, Architecture and Design Process, Systems Methodology, System Description, Service Characteristics, Performance Characteristics, Network Supportability

2. **Requirement Analysis – Concepts & Process**
   Need for Requirement Analysis, User Requirements, Application Requirements, Device Requirements, Network Requirements, Other Requirements, Requirement Specification and Map, Gathering & Listing Requirements, Developing Service Metrics, Characterizing Behavior, Developing RMA Requirements, Developing Delay Requirements, Developing Capacity Requirements, Developing Supplemental Performance Requirements, *Environment-Specific*: Performance Requirements, Thresholds and Limits, Requirements for Predictable and Guaranteed Performance

3. **Flow Analysis**
   Overview and types of Flows, Identifying and Developing Flows, Data Sources and Sinks, Flow Models, Flow Prioritization and Specification, Applications of Flow Analysis

4. **Network Architecture**
   Introduction, Component Architecture, Reference Architecture, Architectural Models, Systems and Network Architecture

5. **Addressing and Routing Architecture**
   Addressing Mechanisms, Routing Mechanisms, Addressing Strategies, Routing Strategies, Architectural Considerations

6. Network Management Architecture & Security Architecture

Defining Network Management, Network Management Mechanisms, Architectural Considerations, Security and Privacy Administration, Security Architectural Considerations

7. Network Design

Design Concepts, Design Process, Vendor, Equipment and Service-Provider Evaluations, Network Layout, Design Traceability, Design Metrics, Testing Network Design, Optimizing Network Design, Documenting Network Design

References:

1. McCabe James, "*Network Analysis, Architecture, and Design*", 3rd edition, Morgan Kaufmann Publishers
2. Priscilla Oppenheimer, "*Top-Down Network Design*", Pearson Education India
3. Hussein Al-Bahadili, "*Simulation in Computer Network Design and Modeling*", Information Science Reference
4. Piliouras Teresa C., "*Network Design: Management and Technical Perspective*", CRC Press
5. Darren Spohn, "*Data Network Design*", McGraw Hill

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Gain capability to make efficient Network Engineering decisions
- Gain ability to Design Networks
- Test, Optimize and Document Networks

***** ***** *****

Course Name: Web Security

Course Code: MCA216(3A)

Objectives:

The objective of the course is to enable students to

- Understand necessity for securing web applications
- Know different risks to web applications
- Take the steps required to mitigate those risks

Prerequisites:

Knowledge of basics of computer system and fundamentals of web applications, database and development

Contents:

1. **Web Application Basics**
   Introduction, HTTP Protocol, Web Functionality, Encoding Schemes, Enumerating Content and Functionality, Analyzing the Application

2. **Authentication Security**
   Authentication Techniques, Design Flaws in Authentication, Implementation Flaws in Authentication, Securing Authentication, Path Traversal Attacks

3. **Injection Attacks**
   Injecting into Interpreted Contexts, SQL Injection, NoSQL Injection, XPath Injection, LDAP Injection, XML Injection, Http Injection, Mail Service Injection

4. **Cross Site Scripting (XSS)**
   Types of XSS, XSS in Real World, Finding and Exploiting XSS Vulnerabilities, Preventing XSS Attacks

5. **User Attacks**
   Inducing User Actions, Capturing Cross- Domain Data, Client-Side Injection Attacks, Local Privacy Attacks, ActiveX Control attacks, Browser Attacks

References:

1. Dafydd Stuttard, "*The Web Application Hacker's Handbook*", Wiley India
2. Andrew Hoffman, "*Web Application Security*", O'Reilly
3. Malcolm Mcdonald, "*Web Security for Developers*", No Starch Press
4. Anand Shinde, "*Introduction to Cyber Security*", Notion Press
5. Jennifer Bayuk, "*Cyber Security Policy Guide Book*", Wiley Publishers

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Detect and solve common web application security vulnerabilities
- Secure Web Applications

***** ***** *****

Course Name: Web Analytics

Course Code: MCA216(3B)

Objectives:

The objective of the course is to enable students to

- Web analytics concepts, principles and techniques
- Measure website traffic and enhance business presence

Prerequisites:

Knowledge of web development and basic java scripting

Contents:

1. **Introduction to Web Analytics**
   Why Understanding Your Web Traffic is Important to Your Business - Available Methodologies and Their Accuracy , Page Tags and Log Files, Apply key concepts of Web Analytics, Introduction to Web Analytics Apply key tools and diagnostics associated with Web analytics

2. **Google Analytics Features and its Interface**
   Google Analytics Features, Benefits and Limitations , Open Source Tool , Tracking the Mobile Visitor, Use free open source Web analytics tools to collect, identify information and data Prepare Embedded JavaScript Page Tracking Code

3. **Web Analytics Reports**
   Reports Explained , Web Analytics, Up and Running , Traffic Sources: AdWords, Implement website traffic reports

4. **Best Practices Configuration**
   Advance Implementation , Best Practices Configuration Guide , The Importance of Defining Goals and Funnels, Capture E-Commerce Transactions Compare and Contrast Web Server Log Files vs. Java Scripting

5. **Google Analytics Customization**
   Labeling Visitors, Sessions, and Pages , Roll-Up Reporting , Define Web Analytics reporting, Implement effective Web analytics strategies for e-Commerce, business, and marketing solutions

6. **Key Performance Indicators**
   Focusing on Key Performance Indicators , Using KPI’s for Web 2.0 , KPI's Setting Objectives and Key Results, Demonstrate how web analytics fits into the business structure

7. **Real World Tasks and Optimization**
   Real World Tasks, Identifying and Optimizing Poorly Performing Pages , Search Engine Optimizing, Interpret the business benefits of web analytics

8. **Integrating Google Analytics with Third-Party Applications**
   Reviews of Analytics Tools, Apply key concepts of Web Analytics , Apply key tools and diagnostics associated with Web analytics, Demonstrate how web analytics fits into the business structure, Interpret the business benefits of web analytics

References:

1. Brian Clifton, "*Advance Web Metrics with Google Analytics*", Sybex
2. Avinash Kaushik., "*Web Analytics 2.0, The Art of Online Accountability and Science of Customer*" , Wiley
3. Michael Beasley, "*Practical Web Analytics for User Experience*", Morgan Kaufmann
4. Stephan A. Miller, Piwik, "*Web Analytics Essentials*", Packt
5. Jim Sterne , "*Web Metrics: Proven Methods for Measuring Web Site Success*", Wiley
6. Brian Clifton, "*Successful Analytics*"

Accomplishments of the student after completing the course:

At the end of the course, students will be able to

- Apply key concepts of Web Analytics
- Use Web Analytics tools to collect data and information
- Implement Web site traffic reports
- Prepare Embedded Java Script Page Tracking Code
- Explain the use of web analytics in the business structure
- Explain the business benefits of Web Analytics

***** ***** *****

NEW SCHEME FOR MASTER OF COMPUTER APPLICATOINS (MCA) COURSE

SEMESTER - IV of MCA

Applicable from December 2021 onwards

Sr. No.	Subject Code	Name of the Subject	TEACHING SCHEME					EXAMINATION SCHEME					
			THEORY	TUTORIAL	PRACTICAL	SESSIONAL		THEORY		PRACTICAL		TERM WORK	TOTAL MARKS
			HR.	HR.	HR.	Marks.	HR.	Marks.	HR.	Marks.	HR.		
1.	MCA221	Software Development Project	-	8	36	200	-	-	-	300	-	200	700
Total			**-**	**8**	**36**	**200**	**-**	**-**	**-**	**300**	**-**	**200**	**700**

DEPARTMENT OF COMPUTER SCIENCE, GUJARAT UNIVERSITY

TEACHING SCHEME (WITH EFFECT FROM DECEMBER 2021)

M.C.A. (Master of Computer Applications)

Semester– IV

Subject Code	SUBJECT	Hours per week			Credits*
		Theory	Tutorial	Lab	
MCA221	Software Development Project	-	8	36	32
Total		**-**	**8**	**36**	**32**

* Theory & Tutorial: - 1 hour = 1 credit
Practical: - 1 hour = 2/3 credit